



# Identification of nonlinear conservation laws for multiphase flow based on Bayesian inversion

Steinar Evje · Hans Joakim Skadsem · Geir Nævdal

Received: 6 March 2023 / Accepted: 27 July 2023 / Published online: 23 August 2023  
© The Author(s) 2023

**Abstract** Conservation laws of the generic form  $c_t + f(c)_x = 0$  play a central role in the mathematical description of various engineering related processes. Identification of an unknown flux function  $f(c)$  from observation data in space and time is challenging due to the fact that the solution  $c(x, t)$  develops discontinuities in finite time. We explore a Bayesian type of method based on representing the unknown flux  $f(c)$  as a Gaussian random process (parameter vector) combined with an iterative ensemble Kalman filter (EnKF) approach to learn the unknown, nonlinear flux function. As a testing ground, we consider displacement of two fluids in a vertical domain where the nonlinear dynamics is a result of a competition between gravity and viscous forces. This process is described by a multidimensional Navier–Stokes model. Subject to appropriate scaling and simplification constraints, a 1D nonlinear scalar conservation law  $c_t + f(c)_x = 0$  can be derived with an explicit expression for  $f(c)$  for the volume fraction  $c(x, t)$ . We consider small (noisy) observation data sets in terms of time series extracted at a few fixed positions in space. The proposed identification method is explored for a range of different displacement condi-

tions ranging from pure concave to highly non-convex  $f(c)$ . No a priori information about the sought flux function is given except a sound choice of smoothness for the a priori flux ensemble. It is demonstrated that the method possesses a strong ability to identify the unknown flux function. The role played by the choice of initial data  $c_0(x)$  as well various types of observation data is highlighted.

**Keywords** Nonlinear conservation law · Multiphase model · Data-driven mathematical modelling · Gaussian random process · Iterative ensemble (Kalman filter) smoother · Bayesian inversion

## 1 Introduction

### 1.1 Learning of PDE models from data in fluid mechanics

Scalar nonlinear conservation laws of the form

$$c_t + f(c)_x = 0, \quad x \in (0, L) \quad (1)$$

represent a rich class of partial differential equations (PDEs) that arise in many different applications including multiphase flow in porous media and wellbore operations relevant for subsurface flow. Here,  $c = c(x, t)$  is the unknown variable which varies in a one-dimensional (1D) domain  $x \in [0, L]$  as time  $t$  elapses. The conservation law (1) must be augmented with an initial state  $c(x, t = 0) = c_0(x)$  and appropriate

S. Evje (✉) · H. J. Skadsem  
University of Stavanger, Stavanger, Norway  
e-mail: steinar.evje@uis.no

H. J. Skadsem  
e-mail: hans.j.skadsem@uis.no

G. Nævdal  
NORCE Norwegian Research Centre AS, Bergen, Norway  
e-mail: geir.naevdal@norce-research.no

assumptions about boundary behavior at  $x = 0$  and  $x = L$ . Starting with general formulations based on first physical principles in terms of conservation of mass, momentum, and energy, one can derive scalar nonlinear conservation laws in the form (1) with an explicit expression for the flux function  $f(c)$ . However, there are many situations where such derivations are intractable or even impossible as they become too complicated or the governing physical laws are unknown. On the other hand, one often nowadays has access to spatio-temporal observation data  $c(x_i, t_j)$  from full numerical solution of the underlying conservation equations, or from controlled experiments. A natural question then is to what extent are we able to identify the unknown flux function  $f(c)$  from such data.

Currently, there is much focus on discovering hidden PDE models from spatio-temporal observation data [1]. A prototype example of PDE model to learn from data is Burgers' viscous equation

$$c_t + \lambda_1 c c_x = \lambda_2 c_{xx} \quad (2)$$

where the main purpose is to identify the unknown parameters  $\lambda_1$  and  $\lambda_2$  [2,3]. However, to the best of our knowledge, such methods have not been used to identify the unknown nonlinear function  $f(c)$  in (1) which appears to be a more complex problem due to the loss of regularity and the need for considering weak solutions supplemented with suitable entropy conditions [4–6].

More precisely, the unique weak entropy solution  $c$  can be characterized by the following integral inequality when (1) is considered on  $\mathbb{R}$  (Cauchy problem)

$$\int_0^T \int_{\mathbb{R}} (|c - k|\varphi_t + \operatorname{sgn}(c - k)(f(c) - f(k))\varphi_x) dx dt + \int_{\mathbb{R}} |c_0(x) - k|\varphi(x, 0) dx \geq 0 \quad (3)$$

for all nonnegative  $\varphi \in C_c^\infty(\mathbb{R} \times [0, T])$  and any constant  $k \in \mathbb{R}$ . The fact that solutions of (1) cannot be understood in a classical sense but in the sense of (3) hampers use of many modern neural network-based methods.

In this paper, we will focus on combining Gaussian random processes to represent the unknown  $f(c)$  through a continuous piecewise linear (affine) function combined with an iterative ensemble Kalman filter (EnKF) approach to update the parameter vector and thereby learn the nonlinear flux function. The EnKF method we use is similar to the one employed

in reservoir engineering setting [7,8]. The EnKF was first introduced as a state estimation approach within oceanography [9], and it was superior to older Kalman filter methods for large-scale nonlinear problems. Subsequently, it was introduced as an approach for parameter estimation in reservoir engineering [10–13]. A number of iterative approaches has since been suggested, e.g., [7,14,15]. An independent line of research was started in [16] which suggested to use the ensemble Kalman filter update equation for solving general inverse problems similar to the ensemble smoother approach. This approach was denoted as Ensemble Kalman Inversion (EKI) and represents a collection of algorithms [16–19] that is motivated by ideas from the EnKF [9].

## 1.2 Previous studies of learning nonlinear flux function from data

The problem of flux function identification related to the nonlinear conservation law (1) has been studied by different researchers the last two decades. Mathematical oriented works typically focus on detecting a precise understanding of the relation between initial data, observation data and underlying flux function. Kang and Tanuma investigated strictly convex flux functions and problems where the solution evolves to a single shock in finite time [20]. Their work proved that the flux function can be uniquely determined on an interval  $[0, \delta]$  with  $\delta > 0$  from late-time observation of the shock corresponding to a given initial datum [20]. James and Sepúlveda formulated the inverse problem of flux identification as that of minimizing a suitable cost function [21]. The flux identification problem by cost function minimization was studied also by Castro and Zuazua [22] who proved the existence of minimizers for solutions with discontinuities and developed appropriate descent algorithms for such problems. Holden et al. used the front-tracking algorithm to reconstruct the flux function from observed solutions to problems with suitable initial data [23].

Recent studies have addressed the reconstruction of the flux function for sedimentation problems that involve the separation of a flocculated suspension into a clear fluid and a concentrated sediment [24]. As discussed by Diehl, estimation of the batch settling flux function is sensitive to the choice of initial condition, and performing experiments with two different ini-

tial conditions can enable identification of both convex and concave parts of the function [25]. Subsequently, Bürger and Diehl showed that the inverse problem of identifying the batch flux density function has a unique solution and derived an explicit formula for the flux function [26]. Betancourt et al. applied the flux identification approach of [26] to experimental data by solving a constrained least-squares minimization problem [27]. The flux identification approach of [26] with different initial conditions (the Kynch test and the Diehl test, respectively) was used in the context of cylindrical vessel combined with data from experiments [28]. An approach based on symbolic multi-layer neural networks were proposed in [29] as a method to recover hidden nonlinear conservation laws. Work on Bayesian inversion in the context of hyperbolic conservation laws is limited. We refer readers to the work [30,31] for more theoretical aspects of wellposedness related to such Bayesian inversion relevant for identification of initial data and flux function.

### 1.3 The contribution of this work

The inverse problem associated with (1) we are interested in takes the form

$$d = h(\theta) + \eta, \tag{4}$$

where  $h$  is the solution operator associated with (1),  $\theta$  is the parameter vector (i.e., a representation of the unknown flux function  $f$ ), and  $d$  is the selected observation data with noise  $\eta$ . The main contribution of this paper is to explore the EnKF-based approach for general scalar conservation laws occurring in a multiphase flow setting which involves a family of non-convex flux functions. The EnKF method allows us to extract statistical information about the unknown flux  $f$  through the parameter vector  $\theta$  from noisy observation data  $d$ . To the best of our knowledge, this is a first work which combines EnKF, sparse observation data, and entropy solutions. The method we use is flexible regarding observation data  $d$  to include in the cost functional (loss function) and is gradient-free. Our aim is to obtain a practical and efficient algorithm which has a potential to identify the physical relevant flux function behind the observation data. A useful and unique feature of the assimilation method is that lack of uniqueness in the determination of the flux function is signaled by showing a larger spread in the updated ensemble of the

candidate flux functions, suggesting that more observation data may be added. The fluid mechanical setting we focus on involves displacement of one fluid by another in an annular geometry. In the following, we briefly describe the more general context of Newtonian fluid displacements in a multidimensional setting as described by the Navier–Stokes model. More details are provided in Appendix A.

### 1.4 Mathematical multiphase model for Newtonian fluid displacements

As the model problem for the current study, we consider laminar fluid displacement within the annular space formed by two concentric cylinders. More specifically, we consider the fluids to be incompressible and Newtonian, and that the displacing fluid is injected from the top. We focus on iso-viscous displacements, i.e., assume the fluids have the same constant viscosity and consider the cases where the displacing fluid may be either denser or lighter than the displaced fluid [32,33].

The laminar displacement flows are governed by the equations of mass and momentum conservation, and we will assume that the concentration of the respective fluids follow an advection–diffusion equation. Specifically, the assumption of incompressibility ensures that the velocity  $\mathbf{u}$  is solenoidal,

$$\nabla \cdot \mathbf{u} = 0. \tag{5}$$

The momentum conservation equation for fluid phase  $i$  is as follows:

$$\begin{aligned} \rho_i (\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u}) \\ = -\nabla p + \mu \nabla^2 \mathbf{u} + \rho_i \mathbf{g}, \quad i = 1, 2. \end{aligned} \tag{6}$$

Here,  $\mu$  is the common viscosity of the two fluids,  $\rho_i$  is the mass density of fluid  $i$ , and  $\mathbf{g}$  corresponds to the gravitational acceleration. We let  $i = 1$  denote the displacing fluid, and  $i = 2$  the displaced fluid. We refer to Fig. 1 for an illustration of the physical setting of the displacement process. Letting  $C \in [0, 1]$  denote the volume fraction of *displacing* fluid, the evolution of this scalar field is given by the advection equation

$$C_t + \nabla \cdot (\mathbf{u}C) = 0. \tag{7}$$

In Appendix A, we briefly show that, subject to certain simplifying assumptions, Eqs. (5), (6) and (7) can be combined to obtain a one-dimensional conservation

equation for the annular cross-sectional averaged concentration  $c(x, t) = \int_A C(\mathbf{r}, t) dA$ :

$$c_t + f(c; \chi)_x = 0. \quad (8)$$

The derivation shown in that section produces an explicit expression for the flux function  $f(c; \chi)$ , which depends on  $c$  and the dimensionless number  $\chi$  which represents the balance between buoyant and viscous stresses.

### 1.5 Problem statement

Motivated by the above derivation of a conservation law (8) from the general model (5)–(7), subject to certain constraints, the problem we explore in this work is: *How can we reconstruct the unknown flux function  $f(c)$  where we only have information about the initial state  $c_0(x)$  and a relatively sparse set of noisy observation data  $d$ ?*

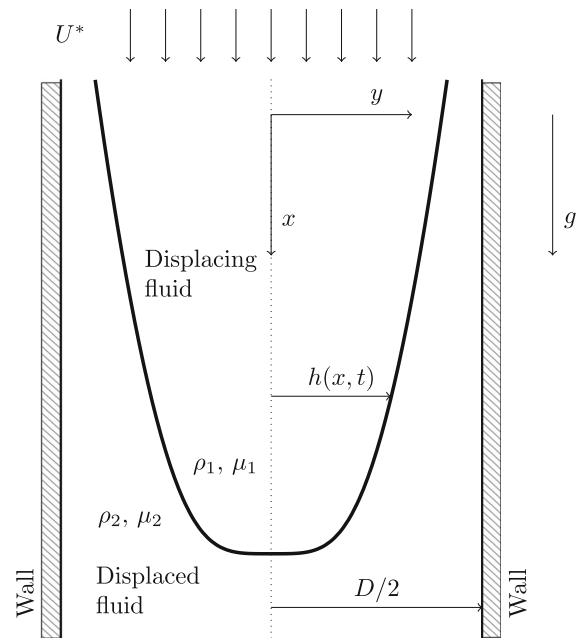
A special challenge with identification of the nonlinear flux function of a conservation law is that solutions typically contain discontinuities, i.e., a jump described by a left and right state  $(c_l(t), c_r(t))$  with  $c_l \neq c_r$  moving along a certain path  $x(t)$  in the  $x$ - $t$ -space, where there is no information used about the specific form of  $f(c)$  in the interval  $[\min(c_l, c_r), \max(c_l, c_r)]$  [4]. Under what circumstances, e.g., knowledge about initial state and corresponding observation data are we able to recover a reliable approximation of the flux function  $f(c)$ ?

Observation data can be given in terms of states at specific times, e.g.,  $c^{obs}(x, t_i^*)$  at times  $\{t_i^*\}_{i=1}^{T_{obs}}$  or time-dependent data  $c^{obs}(x_i^*, t)$  at fixed positions  $\{x_i^*\}_{i=1}^{X_{obs}}$ . The natural initial data  $c_0(x)$  to consider are one which contains an initial jump from 1 to 0 to mimic the situation depicted in Fig. 1 where fluid 1 displaces fluid 2. Consequently, there will be a “mixing zone” that evolves over time, i.e., a region where the volume fraction  $c(x, t)$  stays between  $\epsilon$  and  $1 - \epsilon$  as defined by the set  $E_\epsilon$  given by

$$E_\epsilon(t) = \{x : c(x, t) > \epsilon\} \cap \{x : c(x, t) < 1 - \epsilon\} \quad (9)$$

for some chosen  $\epsilon > 0$ . One may envision to use data like position and velocity related to this mixing zone as well.

Our framework is based on an ensemble Kalman filter (EnKF) approach combined with use of Gaussian random processes to approximate the unknown



**Fig. 1** Lubrication model geometry for vertical and concentric annulus displacements

function  $f(c)$ . Our starting point is that we seek to uncover the unknown flux function  $f(c)$  by relying only on time-dependent observations at fixed positions. We assume that “sensors” at four positions  $x_i^*$ , for  $i = 1, \dots, 4$  have been placed in the region where the interface between fluid 1 and fluid 2 is located. We typically start exploring how well we can learn the hidden flux function from data based on the case with an initial jump from 1 to 0. Then, we proceed by modifying this initial state such that it includes some intermediate states between 0 and 1. This allows for collecting data that potentially contains more information about the unknown flux function. We seek to keep the amount of data used for the learning process at a minimum. We will address questions like:

- (i) How well can we learn  $f(c)$  from noisy data of the volume fraction  $\{c(x_i^*, t)\}_{i=1}^4$  over a training time period  $[0, T_1]$ ? We will consider observation data corresponding to different flow regimes as represented by the parameter  $\chi$  in (8). This amounts to different shapes of the involved flux function, ranging from essentially concave to highly non-convex forms. A representative set of different examples of  $f(c; \chi)$  is shown in Fig. 19.

- (ii) How does the choice of initial condition impact the identified flux function? Can different initial conditions result in improved flux function identification?
- (iii) How can information about the position and velocity of the mixing zone be used?
- (iv) Are we able to also capture information about the role played by the parameter  $\chi$ ? I.e., are we able to extract a good approximation of  $f(u; \chi)$  over a range of different choices of  $\chi$ ?

### 1.6 Structure of this work

The ensemble-based method we explore in this work is described in Sect. 2. In Sect. 3, we give further background information related to the implementation of the identification method and which observation data to use. Finally, in Sect. 4 we carry out a range of different tests of the ability to learn the unknown flux function based on synthetically generated noisy data. We test the method for nonlinear flux functions that contain one or several inflection points.

## 2 Method based on Gaussian random processes and EnKF

Based on the given observation data, our aim is to identify a conservation law (8) now written in the form for  $(x, t) \in [0, L] \times [0, T]$

$$\begin{aligned}
 c_t + \gamma f(c)_x &= 0 \\
 c_x|_{x=0} &= c_x|_{x=L} = 0 \\
 c|_{t=0} &= c_0(x)
 \end{aligned} \tag{10}$$

where  $f(c)$  is the unknown, possible nonlinear flux function and  $\gamma$  is an unknown constant. The introduction of the parameter  $\gamma$  is motivated by the identification of PDEs as discussed in [2, 3] and represented by (2). It also seems convenient from the point of view of the ensemble-based method where  $\gamma$  serves the role as global unknown quantity, whereas the parametrization of  $f(c)$  accounts for the local variations as a function of  $c$ . To estimate  $f(c)$ , we have chosen an approach motivated by the success of large-scale parameter estimation from reservoir engineering [8] where ensemble-based approaches are used to identify spatial varying fields of unknown parameters. The complexity in representing these fields might vary, but quite often they

can be represented as Gaussian random fields. Here, we will adopt this approach by assuming that  $f(c)$  can be identified by representing it as a Gaussian process (a one-dimensional Gaussian random field). For numerical implementation,  $f(c)$  is represented as a piecewise linear function on a designated equidistant grid  $\{c_p\}_{p=1}^P$ .

### 2.1 Gaussian random processes

We will represent  $f(c)$  as a Gaussian process with the property that the correlation between  $f_1 = f(c_1)$  and  $f_2 = f(c_2)$  is given as

$$C_{c_1, c_2} = \sigma^2 e^{-\left(\frac{(c_1 - c_2)^2}{l^2}\right)} \tag{11}$$

where  $\sigma$  is standard deviation and  $l$  is correlation length, i.e., a scaling parameter that determines how fast the correlation decays as function of the distance between  $c_1$  and  $c_2$ . To calculate such a Gaussian process on  $P$  grid points  $\{c_p\}_{p=1}^P$ , we first sample a  $P$ -vector  $\zeta$  of independently zero-mean normally distributed random numbers with variance 1. A Gaussian process  $\psi$  with the required properties is then obtained by letting

$$\psi = L\zeta \tag{12}$$

where  $C = LL^T$ . A common choice is to obtain  $L$  by taking the Cholesky decomposition of  $C$ . Note that the Cholesky decomposition is only required once for all samples of  $f(c)$  that should be generated. For modest size  $P$  to describe the unknown  $f(c)$  as used in this work, this leads to an accurate algorithm with sufficient efficiency.

### 2.2 Ensemble smoother—multiple data assimilation

The initial ensemble  $E_f$  might be obtained using the algorithm presented in Sect. 2.1. The updated ensemble  $E_a$  is obtained using Algorithm 1. The ensemble smoother—multiple data assimilation (ES-MDA) was first presented in [7]. Algorithm 1 deviates slightly from their work by replacing the stochastic ensemble Kalman filter (see, e.g., [34]) for the update part with a square-root-filter. The requirement  $\sum_{i=1}^{N_{it}} \frac{1}{\alpha_i} = 1$  on the scaling factors ensures that for a linear problem with a Gaussian prior, the posterior solution will be the correct one. For nonlinear problems, the solution will



**Algorithm 1:** Ensemble smoother—multiple data assimilation (ES-MDA)

**Input:**  $N_{\text{ens}}$ : ensemble size;  $E_f$ : initial ensemble,  $m$ : measurement vector;  $W$ : covariance matrix for measurement errors;  $H$ : relationship between states and measurement;  $N_{\text{it}}$ : Number of iterations  
**Output:**  $E_a$ : final ensemble  
 Select  $N_{\text{it}}$  scaling factors,  $\alpha_i$ , such that  $\sum_{i=1}^{N_{\text{it}}} \frac{1}{\alpha_i} = 1$ .  
**for**  $i \leftarrow 1$  **to**  $N_{\text{it}}$  **do**  
     Solve (10) with  $N_{\text{ens}}$  parameter vectors as given by  $E_f$   
      $E_f \leftarrow$  Ensemble square-root filter( $N_{\text{ens}}, E_f, m, \frac{W}{\alpha_i}, H$ ) (Algorithm 2)  
 **$E_a \leftarrow E_f$**

be an approximation, and a number of alternative algorithms have been developed. For a recent discussion on algorithms available to solve large-scale parameter estimation problems with ensemble-based methods, see [8, Sect. 5.3.2].

**Algorithm 2:** Ensemble square-root filter

**Input:**  $N_{\text{ens}}$ : ensemble size;  $E_f$ : ensemble from forecast;  $m$ : measurement vector;  $W$ : covariance matrix for measurement errors;  $H$ : relationship between states and measurement  
**Data:**  $\mathbf{1}_{N_{\text{ens}}}$ : column vector of length  $N_{\text{ens}}$ ;  $I_{N_{\text{ens}}}$ :  $N_{\text{ens}} \times N_{\text{ens}}$ -identity matrix  
**Output:**  $E_a$ : updated ensemble  
 Step 1:  $\psi_f = \frac{1}{N_{\text{ens}}} E_f \mathbf{1}_{N_{\text{ens}}}$ ;  
 $L_f = \frac{1}{\sqrt{N_{\text{ens}}-1}} (E_f - (I_{N_{\text{ens}}}^T \otimes \psi_f))$ ;  
 $M_f = H L_f$ ;  
 $K = L_f (M_f)^T (M_f (M_f)^T + W)^{-1}$ ;  
 $\psi_a = \psi_f + K (m - H \psi_f)$ ;  
 Step 2:  $A_f = \sqrt{N_{\text{ens}} - 1} L_f$ ;  
 $[C, G] = \text{svd} \left( I_{N_{\text{ens}}} + \frac{(H A_f)^T W^{-1} (H A_f)}{(N_{\text{ens}} - 1)} \right)$ ;  
 $T = C G^{-1/2} C^T$ ;  
 $A_a = A_f T$ ;  
 $E_a = A_a + \mathbf{1}_{N_{\text{ens}}}^T \otimes \psi_a$ ;

We use the square-root-filter presented in Algorithm 2. It involves two main steps: (i) compute updated mean  $\psi_a$  based on  $K$ ; (ii) compute corresponding updated ensemble  $E_a$  based on  $T$ . The basic idea behind this square-root-filter was introduced in [35] and further developed by [36]. The square-root-filters are based on the basic idea of getting a posterior ensemble with the correct posterior covariance matrix, and a transformation matrix  $T$  achieving this has to be found. It was not immediately realized that this transformation

matrix should be mean preserving (which is equivalent to preserving the zero-mean property of  $A_f$ ). This was pointed out in [37,38]. Our choice of transformation matrix  $T$  fulfills both the two requirements. A discussion, and historical account, of square-root-filters are also included in [34]. In Algorithm 2, the singular value decomposition of a symmetric matrix is computed. The singular value decomposition of a symmetric matrix,  $A$  has the form  $A = U \Sigma U^T$  where  $U$  is an unitary matrix and  $\Sigma$  is a diagonal matrix. Calculating this singular value decomposition is feasible as the size of the matrix is  $N_{\text{ens}} \times N_{\text{ens}}$ .

2.3 Discrete numerical scheme

We base our discrete version of (10) on the Rusanov scheme [4] which takes the form

$$c_j^{n+1} = c_j^n - \lambda (F_{j+1/2}^n - F_{j-1/2}^n),$$

$$\lambda = \frac{\Delta t}{\Delta x} \gamma \tag{13}$$

with  $j = 2, \dots, N_x - 1$  and where the Rusanov flux takes the form

$$F_{j+1/2}^n = \frac{f_j^n + f_{j+1}^n}{2} - \frac{M}{2} (c_{j+1}^n - c_j^n),$$

$$M \sim \max_c |f'(c)|. \tag{14}$$

For the first and last grid cell, the boundary condition gives  $c_1^{n+1} = c_2^{n+1}$  and  $c_{N_x}^{n+1} = c_{N_x-1}^{n+1}$ . We use a slightly modified version of the Rusanov flux by relying on a global estimate of  $|f'(c)|$  instead of a local estimate of  $M$  in terms of  $M_{j+1/2} = \max\{|f'(c_j^n)|, |f'(c_{j+1}^n)|\}$  [4]. The CFL condition [4] determines the magnitude of  $\Delta t$  for a given  $\Delta x$  through the relation

$$CFL := \frac{\Delta t}{\Delta x} |\gamma| M \leq 1. \tag{15}$$

We have used  $\frac{\Delta t}{\Delta x} |\gamma| M = CFL \leq \frac{3}{4} < 1$  when we compute solutions based on the ensemble of flux functions involved in the learning process.

3 Preliminaries for numerical investigations

We focus on the flux function  $f(c; \chi)$  obtained in (A.3) which takes the form

$$f(c; \chi) = \frac{c}{2} (3 - c^2)$$

$$+ \frac{\chi c^2}{12} \left( \frac{3}{4} - 2c + \frac{3}{2}c^2 - \frac{1}{4}c^4 \right) \tag{16}$$

when we generate synthetic observation data. Initially, we are interested in an initial state of the form

$$c_0(x) = \begin{cases} 1, & x \in [0, 3.5) \\ 0, & x \in (3.5, 10] \end{cases} \tag{17}$$

where we have a pure interface separating the two fluids in the domain  $[0, 10]$ . This is motivated by the physical setting for carrying out laboratory experiments. However, we may expect to see limitations in identification of the true flux function when it is based on data which has been collected as a result of using (17). That is, the possibility for discontinuities to evolve may prevent collecting observation data for all  $c$  in  $[0, 1]$ . This motivates us to also consider the following initial state:

$$c_0(x; c_I, c_{II}) = \begin{cases} 1.0, & x \in [0, 2] \\ c_I, & x \in (2, 3.5] \\ c_{II}, & x \in (3.5, 5] \\ 0, & x \in (5, 10] \end{cases}, \quad 1 > c_I > c_{II} > 0 \tag{18}$$

where  $c_I$  and  $c_{II}$  are appropriate chosen intermediate states. This form of the initial data may imply that more information about the flux function can be obtained when we collect time-dependent data at the positions

$$x_1^* = 1.5, \quad x_2^* = 2.0, \quad x_3^* = 3.5, \quad x_4^* = 5.5. \tag{19}$$

The motivation for using (18) instead of (17) is to seek recovering more details of the unknown flux function  $f(c)$  recalling that for a shock wave  $(c_l, c_r)$  with  $c_l \neq c_r$ , it is only the linear slope that connects  $(c_l, f(c_l))$  and  $(c_r, f(c_r))$  that is involved. I.e., the part of  $f(c)$  within the interval  $[\min(c_l, c_r), \max(c_l, c_r)]$  is not used in the construction of the exact entropy solution [4]. In our investigations, we will also consider modified versions of (18) which involve more/fewer intermediate states connecting the initial jump from 1 to 0.

Observation data set

We consider the data set

$$d_A(t) = \left\{ c(x_1^*, t), c(x_2^*, t), c(x_3^*, t), c(x_4^*, t); \Delta_t c(x_1^*, t), \Delta_t c(x_2^*, t), \Delta_t c(x_3^*, t), \Delta_t c(x_4^*, t) \right\} \tag{20}$$

corresponding to collecting time dependent data at fixed positions  $\{x_I^*\}_{I=1}^4$ . Herein,  $\Delta_t c(x_I^*, t)$  refers to

a discrete difference along time axis that involve specified times where observation data are collected. Note that time derivative  $\frac{dc}{dt}$  does not make sense as the true solution typically involve a discontinuous function with respect to time variable. We also explore the effect of augmenting the observation data with information about the mixing zone as expressed by

$$d_A^{mix}(t) = \left\{ x_b^*(t), x_f^*(t); \Delta_t x_b^*(t), \Delta_t x_f^*(t) \right\} \tag{21}$$

where we have defined

$$x_b^*(t) = \inf_x E_\epsilon(t), \quad x_f^*(t) = \sup_x E_\epsilon(t) \tag{22}$$

and  $E_\epsilon(t)$  is defined by (9). We have used  $\epsilon = 0.025$  throughout the rest of this paper.

We consider a domain of length  $L = 10$  and consider simulations over the time period  $[0, T]$  with  $T = 2$ . We apply a numerical grid composed of  $N_x = 1000$  grid cells when we compute numerical solutions of (8) based on the numerical scheme (13) and (14). This is used both for obtaining the true solution and corresponding synthetic observation data (which we denote by  $\mathbf{m}$ ) by as well as when we compute predictions based on the ensemble of estimated flux functions (which we denote by  $\mathbf{d}$ ). We specify times for collecting the time dependent data

$$t_i^{obs} = i \Delta t^{obs}, \quad i = 1, \dots, N_t^{obs}. \tag{23}$$

We have set  $N_t^{obs} = 40$ , i.e.,  $\Delta t^{obs} = 0.05$ . Hence, the number of collected data points in time is sparse compared to the number of local time steps  $\Delta t$  we need to compute numerical solutions through the discrete scheme (13) and (14). As training data, we will use points in the time interval  $[0, T_1]$  with  $T_1 = 1$ .

Main algorithm

The main algorithm, based on Algorithm 1 and Algorithm 2, is given through the following steps:

- 1. Generating initial parameter vector  $\theta_k$ .** Generate an initial ensemble of samples  $\{\theta_k\}_{k=1}^{N_{ens}}$  with  $N_{ens} = 100$  where  $\theta_k = \{\gamma_k, f_k\}$ . Herein,  $\gamma_k$  is a scalar generated from the uniform random distribution of numbers in  $[0, 2]$  and  $f_k$  is a vector that represents the piecewise linear function generated as a Gaussian random process on the the uniform grid  $\{c_p\}_{p=1}^P$  which represents the interval  $[-0.1, +1.1]$  with  $c_p = -0.1 + \frac{1}{2} \Delta c$  and  $\Delta c = 1.2/P$ . That is,  $f_k = \mu_k + \psi_k$  (24)

where  $\mu_k$  is a mean value randomly sampled from a uniform distribution in the interval  $[0, 1]$  and the Gaussian process  $\psi_k = L\zeta_k$  is constructed from the correlation matrix  $C$ , see (11) and (12) in Sect. 2.1. We have used  $\sigma = 0.25$  and  $l = 20\Delta c$  as standard deviation and correlation length, respectively, in (11). We have used  $P = 52$  such that the length of the parameter vector  $\theta_k$  is 53 (length of  $f_k$  and the scalar  $\gamma_k$ ).

2. **Computing predicted observation**  $d_k = h(\theta_k)$ . Using the discrete scheme (13) and (14) and the initial data (e.g., (17) or (18)), we obtain the discrete approximation denoted by  $c^k = c(x_j, t^n; \theta_k)$  (to indicate its dependence on the flux function described by  $\theta_k$  through  $\gamma_k f_k$ ) until time  $T_1 = 1$  and collect the predicted data set  $d_k = h(\theta_k)$  where  $h(\theta_k)$  refers to the data set (20). More precisely, it refers to the discrete version

$$d_A^k(t_i^{\text{obs}}) = \left\{ c^k(x_1^*, t_i^{\text{obs}}), c^k(x_2^*, t_i^{\text{obs}}), c^k(x_3^*, t_i^{\text{obs}}), c^k(x_4^*, t_i^{\text{obs}}); \Delta_t c^k(x_1^*, t_i^{\text{obs}}), \Delta_t c^k(x_2^*, t_i^{\text{obs}}), \Delta_t c^k(x_3^*, t_i^{\text{obs}}), \Delta_t c^k(x_4^*, t_i^{\text{obs}}) \right\} \quad (25)$$

for  $i = 1, \dots, N_t^{\text{obs}}/2$ . Herein,  $\Delta_t c^k(x_I^*, t_i^{\text{obs}}) = \frac{1}{2} (c^k(x_I^*, t_{i+1}^{\text{obs}}) - c^k(x_I^*, t_{i-1}^{\text{obs}})) / \Delta t^{\text{obs}}$  for  $i = 2, \dots, N_t^{\text{obs}}/2 - 1$  and  $I = 1, 2, 3, 4$ . Similarly, we consider the discrete version of (21) given by

$$d_A^{\text{mix},k}(t_i^{\text{obs}}) = \left\{ x_b^{k,*}(t_i^{\text{obs}}), x_f^{k,*}(t_i^{\text{obs}}); \Delta_t x_b^{k,*}(t_i^{\text{obs}}), \Delta_t x_f^{k,*}(t_i^{\text{obs}}) \right\} \quad (26)$$

for  $i = 1, \dots, N_t^{\text{obs}}/2$ . Herein,  $\Delta_t x_b^{k,*}(t_i^{\text{obs}}) = \frac{1}{2} (x_b^{k,*}(t_{i+1}^{\text{obs}}) - x_b^{k,*}(t_{i-1}^{\text{obs}})) / \Delta t^{\text{obs}}$  for  $i = 2, \dots, N_t^{\text{obs}}/2 - 1$ . As default, we include observation data of the form (25) as well as (26). I.e.,

$$d_k = h(\theta_k) = \{ \lambda_1 d_A^k(t_i^{\text{obs}}) \} \cup \{ \lambda_2 d_A^{\text{mix},k}(t_i^{\text{obs}}) \}, \quad t_i^{\text{obs}} = 1, \dots, N_t^{\text{obs}}/2. \quad (27)$$

Herein,  $\lambda_1$  and  $\lambda_2$  are scaling parameters to ensure a proper balance between observation data of different types (i.e., order of magnitude should be comparable). We have used consistently  $\lambda_1 = 1$  and  $\lambda_2 = \frac{1}{2}$  throughout the simulation results if nothing else is specified.

3. **Initial data for Algorithm 1 (ES-MDA)** Based on Step 1 and 2, we can form the initial ensemble  $E_f$  given by

$$E_f = \begin{bmatrix} d_1 & d_2 & \dots & d_{N_{\text{ens}}} \\ \theta_1 & \theta_2 & \dots & \theta_{N_{\text{ens}}} \end{bmatrix}$$

where  $d_k$  and  $\theta_k$  are written as column vectors. Together with the measurement data vector  $m$ , which refers to the measured (observed) data of the form (27), and information about the measurement noise through  $W$  obtained by assuming Gaussian white noise, we are in the position that we can initiate ES-MDA( $N_{\text{ens}}, E_f, m, W, H, N_{\text{it}}$ ) (Algorithm 1). Herein,  $H$  represents a projection matrix which extracts from  $E_f$  the part that is related to the measurement vector  $m$ .

4. **Repeated use of Algorithm 2** Armed with the initial ensemble  $E_f$  from Step 3, we use Algorithm 1 in combination with Algorithm 2 (see Sect. 2.2) to update ensemble  $E_f$ . This involves repeated use of Step 2 given above to find updated  $d_k = h(\theta_k)$ . Finally, after  $N_{\text{it}}$  iterations, we have arrived at  $E_a$ , in accordance with Algorithm 1. From this, we extract the updated final ensemble  $\{ \theta_k^{N_{\text{it}}}, N_{\text{ens}} \}_{k=1}^{N_{\text{ens}}}$  which in turn gives us the ensemble  $\{ \gamma_k^{N_{\text{it}}} f_k^{N_{\text{it}}}, N_{\text{ens}} \}_{k=1}^{N_{\text{ens}}}$  of piecewise linear functions defined on the uniform grid  $\{ c_p \}_{p=1}^P$ .

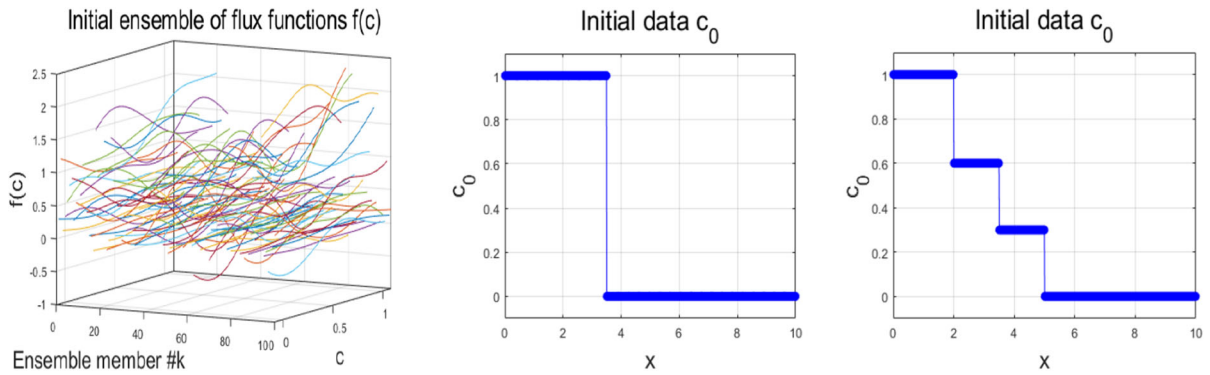
### 4 Results

An illustration of the initial ensemble of samples  $\{ \theta_k \}_{k=1}^{100}$  is found in Fig. 2 (left) which plots  $\{ \gamma_k f_k(c_p) \}_{p=1}^{52}$  for  $k = 1, \dots, 100$ . In the middle, we see the initial data (17) we have used to obtain the observation data and also employed in the learning process. Similarly, in the right figure we have plotted initial data of the form (18) corresponding to  $c_0(x; 0.6, 0.3)$ .

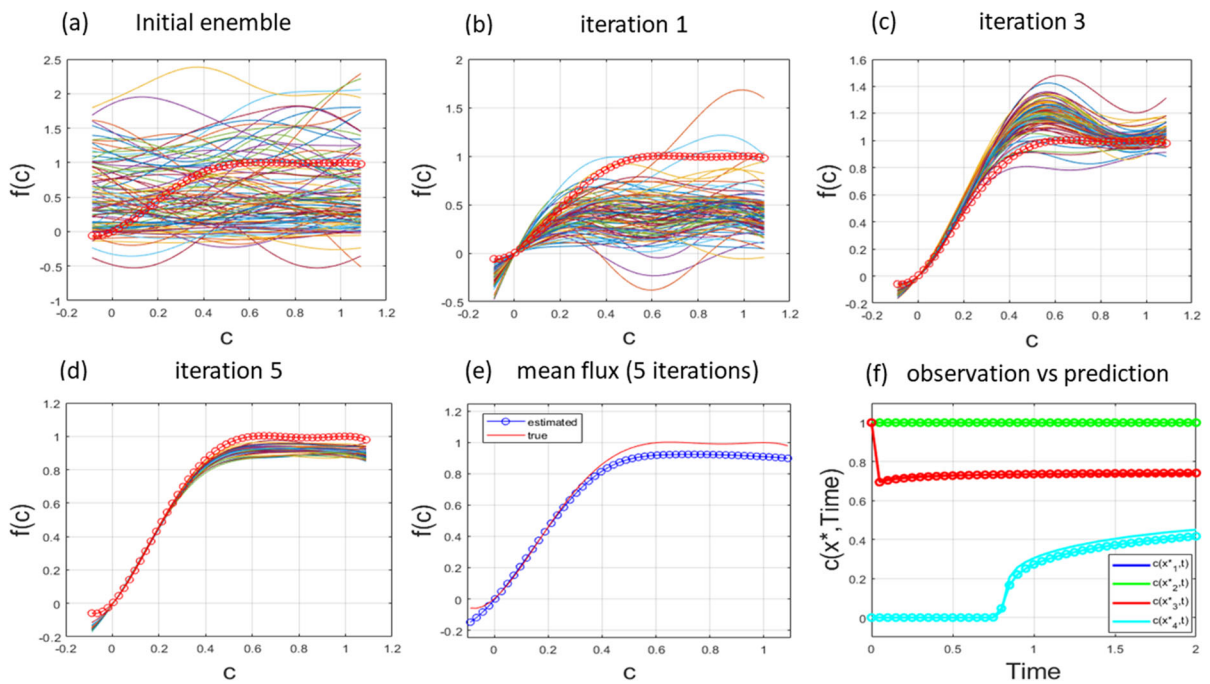
- 4.1 We can learn the unknown  $f(c)$  by using sparse time-dependent noisy data

We focus on the case where synthetic data are obtained using  $f(c; \chi = 120)$ , see (A.3), in combination with initial data (17). In Fig. 3, the results of the identification of the flux function are illustrated. The main observations are: The initial ensemble of possible flux functions span over a large space, see Fig. 3a. After 3





**Fig. 2** Left: Initial ensemble  $\{\theta_k\}$  where  $\theta_k = \{\gamma_k, f_k\}$ . Plot is shown of  $\gamma_k f_k$  for  $k = 1, \dots, 100$ . Middle: Initial data  $c_0(x)$  given by (17). Right: Initial data  $c_0(x; 0.6, 0.3)$  of the form (18)



**Fig. 3** Illustration of the learning process for the case where synthetic data are generated from the flux function (16) with  $\chi = 120$  combined with initial data (17). **a** The initial ensemble of flux functions  $\gamma_k f_k$  span over a wide class of possible flux functions. **b** After 1 iteration, i.e., we compute predictions based on initial ensemble and compute an updated ensemble  $\{\theta_k^{(1)}\}$ .

**c** Updated ensemble  $\{\theta_k^{(3)}\}$ . **d** Updated ensemble  $\{\theta_k^{(5)}\}$ . **e** The mean  $\theta^* = \text{mean}\{\theta_k^{(5)}\}$ . **f** Observed data versus predicted behavior. Solid line represents observed data, circles show predicted result. Note that in panels (b)–(d) we plot  $\gamma_k(f_k(c_i) - f_k(c_5))$  where  $f_k(c_5) \approx f_k(0)$  to make the comparison of the different ensemble flux functions clearer

iterations (Fig. 3c), there is still quite much uncertainty related to the region  $c > 0.2$ , whereas after 5 iterations, the more uncertain region has been reduced to  $c > 0.4$ . However, as seen from the mean flux function in Fig. 3e there is a systematic error in the learned  $f(c)$

for  $c > 0.4$ . What is the discrepancy between the predicted data based on the learned flux function in Fig. 3e and observed data? This is visualized in Fig. 3f where solid line represents observed data (shown here without noise) and circles are used to show predicted behavior

based on the mean flux in Fig. 3e. We observe that the time-dependent curves at  $x_1^*$  (blue) and  $x_2^*$  (green) coincide and contain no variations. This clearly makes the learning process challenging. Taking a closer look at data associated with  $x_4^*$  (light blue) we also see a small discrepancy between observed and predicted which is associated with the erroneous part of the learned  $f(u)$ .

In order to collect more characteristic information, we now replace the initial data (17) by (18) specified as  $c_0(x; 0.6, 0.3)$ . The result is shown in Fig. 4. Most strikingly, after 3 iterations there seems to be quite good agreement on the shape of the flux function  $f(c)$  (see Fig. 4c), and the resulting mean of the ensemble vector  $\{\theta_k^{(5)}\}$  after 5 iterations indeed fits well with the true  $f(c; \chi = 120)$ , see panel 4e. The observed data associated with  $x_1^*, \dots, x_4^*$  are shown in Fig. 4f from which we see a larger part of the interval  $c \in [0, 1]$  is covered through the time-dependent data. This explains why the true flux  $f(c; \chi = 120)$  can be identified much better than for the case with initial data (17) shown in Fig. 3f. This is also visualized in the histograms shown in Fig. 5 where we compare the observation data set for the two different initial states. Clearly, the use of  $c_0(x; 0.6, 0.3)$  (right figure) shows that observed  $c$ -values are more uniformly distributed in  $[0, 1]$  although there are relatively large gaps between them.

An interesting aspect of the method is that it provides insight into which regions the learning process, based on the used data, find it more challenging to decide on the shape of  $f(c)$ . This information may be used to suggest other type of observation data that can help improving the learning of the flux function.

What is the observation data we have used in this learning process? In Fig. 6, we have illustrated the time-dependent data collected at positions  $\{x_i^*\}_{i=1}^4$  given by (19) (left plot), whereas collected data of the positions of the mixture zone as given by (22) with  $\epsilon = 0.025$  are shown in the right figure. 3% noise has been added to the observation data before used for the learning process, i.e., we account for measurement error  $\eta \sim N(0, \gamma^2 \mathcal{I})$  associated with observations  $d$  in (4) where  $\gamma = 0.03$  and  $\mathcal{I}$  is the identity matrix. Similarly for  $d_A^{\text{mix},k}$  in (26).

We may wonder what is the role of the information used about the position of the mixing zone through  $x_b^*(t)$  and  $x_f^*(t)$ . To test this, we run an identification simulation by only using the data set (25), i.e., we set the scaling parameter  $\lambda_2 = 0$  in (27). The result is shown in Fig. 7. Main observation is that there is a larger degree

of uncertainty in the learning after 3 iterations (panel a) as compared to the case when we used the data set (27) illustrated in Fig. 4c. However, after 5 iterations shown in Fig. 7b results are quite comparable. And the mean flux function illustrated in Fig. 7c to a large extent coincides with the true flux  $f(c; \chi = 120)$  apart for  $c$  close to 1. For the sake of robustness, for example, to quicker detect the possibility for backflow for larger  $c$  values, we use the data set (27) in the continuation.

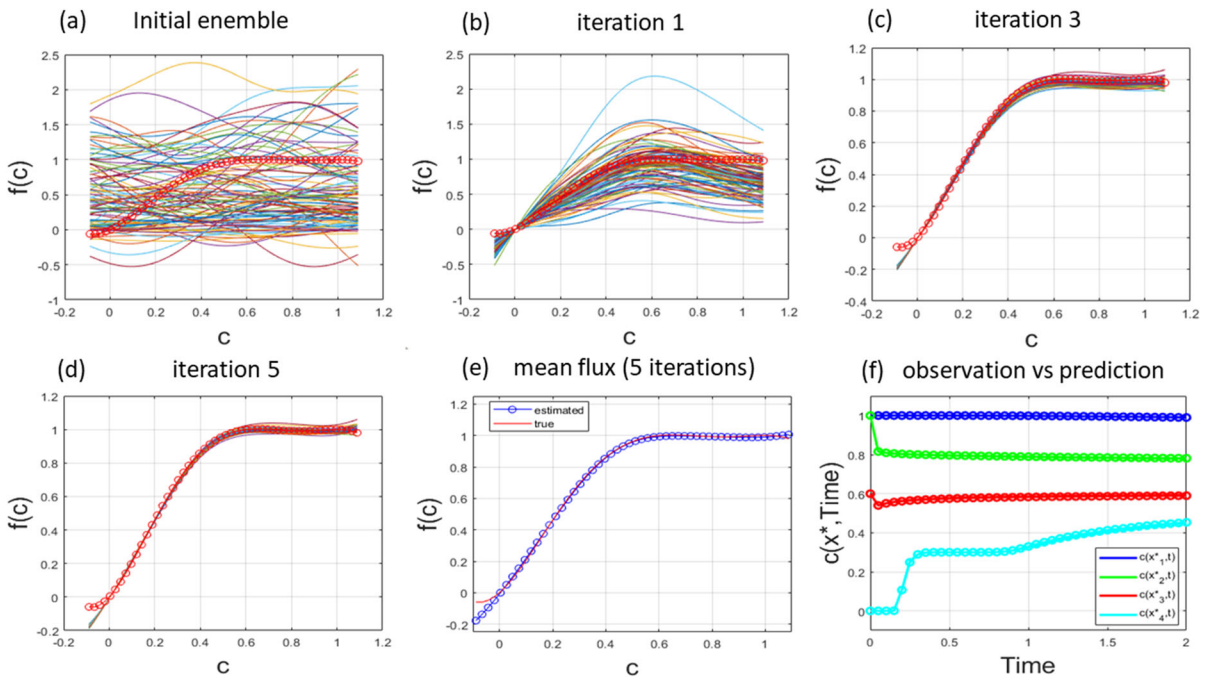
A main objective of this work is to demonstrate that the proposed methodology is general enough to learn the relevant nonlinear flux function for a large range of  $\chi$  values in (A.3). Hence, we now use the same type of observation data as above, i.e., based on (27), and consider synthetic data for flux functions in the range  $\chi \in [50, 300]$  combined with initial data  $c_0(x; 0.6, 0.3)$ . In Fig. 8, results are shown for synthetic data based on the conservation law with flux function  $f(u; \chi = 50)$  (column a),  $f(u; \chi = 150)$  (column b),  $f(u; \chi = 200)$  (column c), and  $f(u; \chi = 300)$  (column d). All cases reflect that the true flux function to a large extent is captured well, also when  $\chi$  increases which gives rise to a stronger convex dip for  $u \geq 0.5$ . This suggests that the identification process covers well the different flow regimes corresponding to  $\chi \in [0, 300]$ .

#### 4.2 The learning of $f(c)$ is mildly sensitive to the use of intermediate states in the initial interface

Clearly, the effectiveness of learning the nonlinear flux function depends on the data used in the iterative method to uncover the unknown flux function. Above, we have used an initial state that jumps from 1 to 0 via two intermediate states 0.6 and 0.3. What is the role played by these intermediate states? As a test case, we consider synthetic data based on (A.3) with  $\chi = 200$  combined with the initial data

$$c_0 = c_0(x; 0.7, 0.4) = \begin{cases} 1.0, & x \in [0, 2] \\ 0.7, & x \in (2, 3.5] \\ 0.4, & x \in (3.5, 5] \\ 0, & x \in (5, 10]. \end{cases} \quad (28)$$

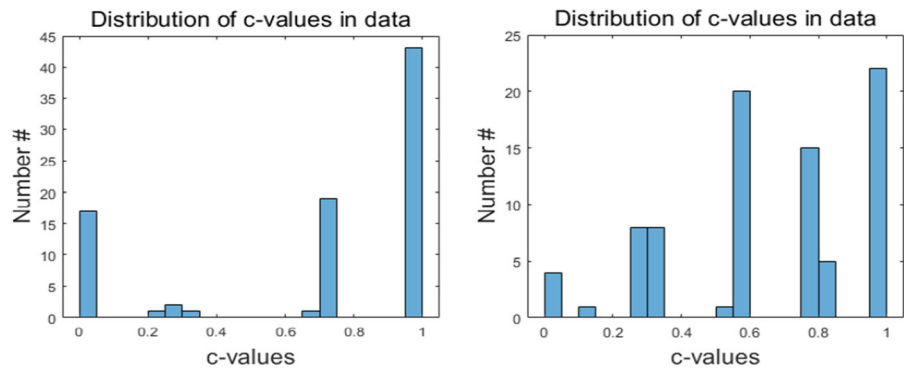
We find that there is virtually no difference from the case with initial data  $c_0(x; 0.6, 0.3)$  (results not shown) used above. Looking at the corresponding histograms in Fig. 9 for the case when we try to identify  $f(c; \chi = 200)$ , we see that they are quite similar, thereby suggesting that the learning processes are of similar quality.



**Fig. 4** Illustration of the identification process for the case where synthetic data are generated from the flux function (16) with  $\chi = 120$  and with initial data (18) given by  $c_0(x; 0.6, 0.3)$ . **a** The initial ensemble of flux functions  $\gamma_k f_k$  span over a wide possible class of flux functions. **b** Updated ensemble  $\{\theta_k^{(1)}\}$ . **c**

Updated ensemble  $\{\theta_k^{(3)}\}$ . **d** Updated ensemble  $\{\theta_k^{(5)}\}$ . **e** The mean  $\theta^* = \text{mean}\{\theta_k^{(5)}\}$ . **f** Observed data versus predicted behavior. Solid line represents observed data, and circles show predicted result

**Fig. 5** Distribution of the different  $c$ -values involved in the observed data used for the learning of  $f(c; \chi = 120)$ . Left: Observed data based on initial data  $c_0(x)$  given by (17). Right: Observed data based on initial data  $c_0(x; 0.6, 0.3)$  corresponding to (18)



Moreover, we also check the initial data

$$c_0 = c_0(x; 0.8, 0.5) = \begin{cases} 1.0, & x \in [0, 2] \\ 0.8, & x \in (2, 3.5] \\ 0.5, & x \in (3.5, 5] \\ 0, & x \in (5, 10] \end{cases} \quad (29)$$

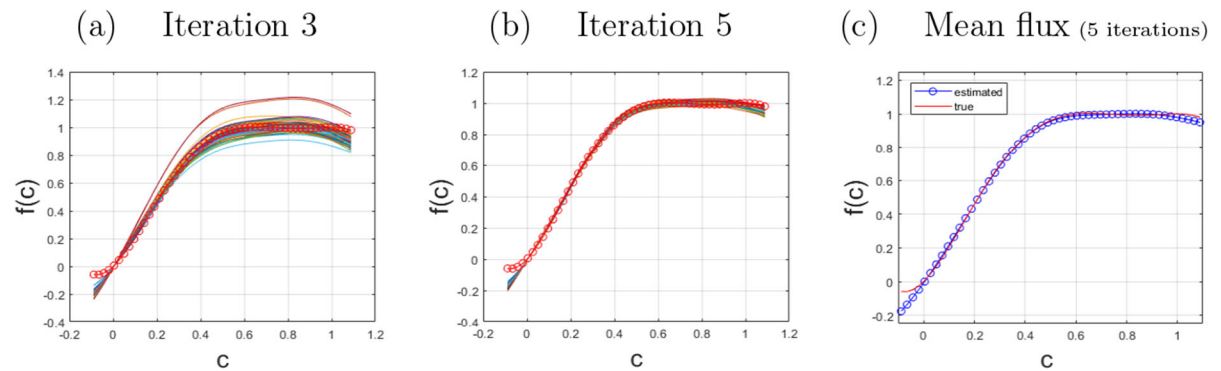
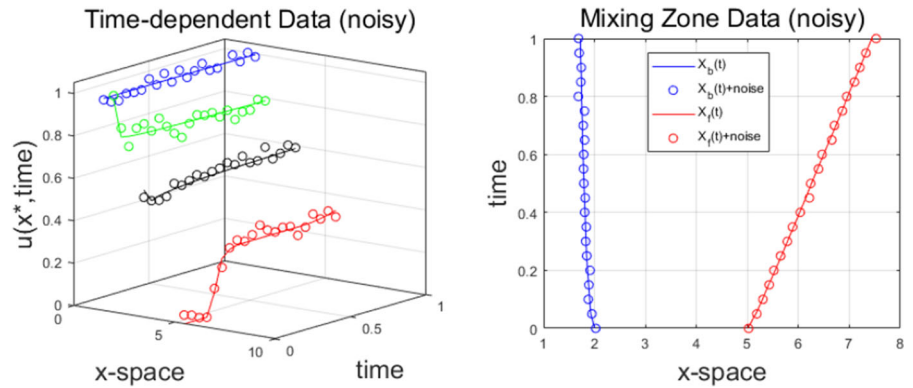
As a test, we consider (A.3) with  $\chi = 120$  and find an excellent match between learned  $f(c)$  and the true (results not shown). Next, we want to increase the num-

ber of intermediate states by considering

$$c_0(x) = \begin{cases} 1.0, & x \in [0, 2] \\ 0.8, & x \in (2, 2.75] \\ 0.6, & x \in (2.75, 3.5] \\ 0.4, & x \in (3.5, 4.25] \\ 0.2, & x \in (4.25, 5] \\ 0, & x \in (5, 10] \end{cases} \quad (30)$$

We used data based on (A.3) with  $\chi = 120$ . The result is shown in Fig. 10 (upper row) and reflects a poorer

**Fig. 6** Left: Illustration of the time-dependent observation data  $\{u(x_i^*, t_j)\}$  with  $i = 1, \dots, 4$  and  $j = 0, \dots, 20$  we use for the training. We have added 3% noise to the data. Right: Time-dependent data in terms of the positions  $x_b^*(t_j)$  and  $x_f^*(t_j)$  with  $j = 0, \dots, 20$



**Fig. 7** Illustration of the learning process for the case where synthetic data are generated from the flux function (16) with  $\chi = 120$ , but only the data set (25) is used. **a** Updated ensemble  $\{\theta_k^{(3)}\}$ . **b** Updated ensemble  $\{\theta_k^{(5)}\}$ . **c** The mean  $\theta^* = \text{mean}\{\theta_k^{(5)}\}$

approximation for  $c > 0.4$  (loss of some finer details of  $f(u)$ ) as compared to, e.g., Fig. 4 suggesting that these smaller initial jumps relatively densely located make the identification process slightly less efficient. Further testing based on (A.3) with  $\chi = 200$  shown in Fig. 10 (middle row) shows the same trend: The higher number of smaller initial jumps included in the initial data slightly impedes the identification of the flux function, see Fig. 10 (column c). In Fig. 10 (panel d and e), we also see the distribution of the  $c$ -values involved in the observation data corresponding to, respectively, learning of  $f(c; \chi = 120)$  (left) and  $f(c; \chi = 200)$  (right). The lack of observation data for  $c \sim (0.6, 0.9)$  appears to be a natural reason for a slightly poorer identification of  $f$  in this interval.

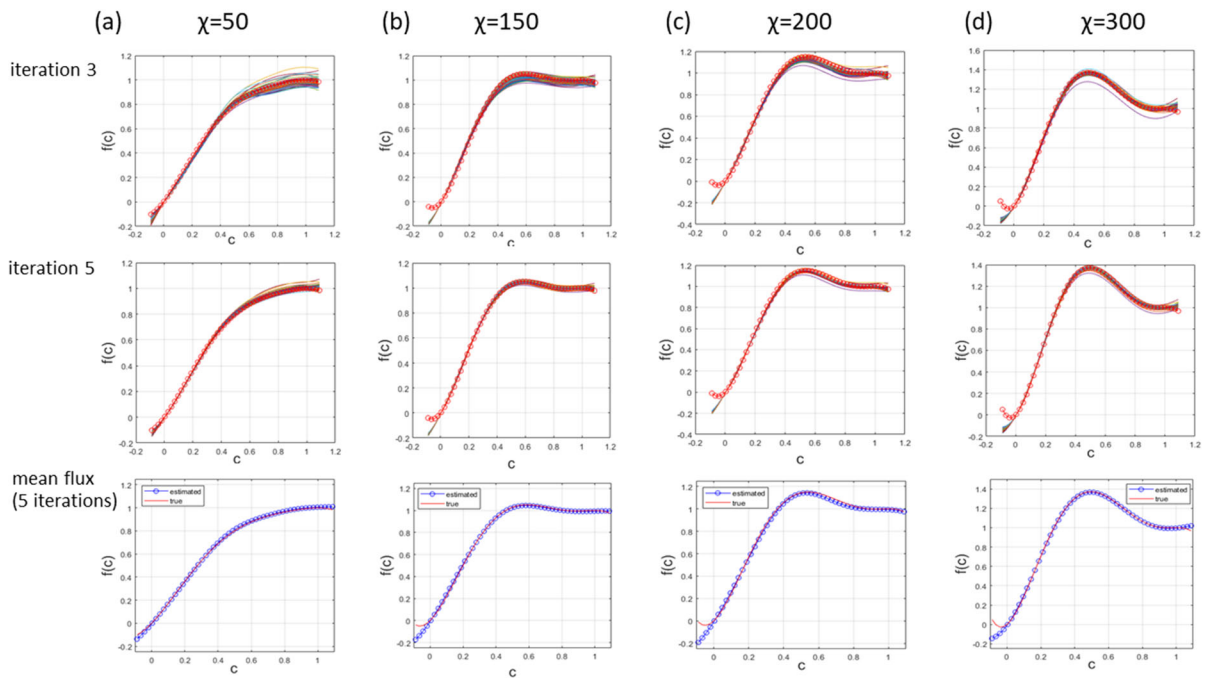
What is the effect of using fewer intermediate states? We consider the initial  $c_0(x)$  given by

$$c_0(x) = \begin{cases} 1.0, & x \in [0, 2] \\ 0.5, & x \in (2, 5] \\ 0, & x \in (5, 10] \end{cases} \quad (31)$$

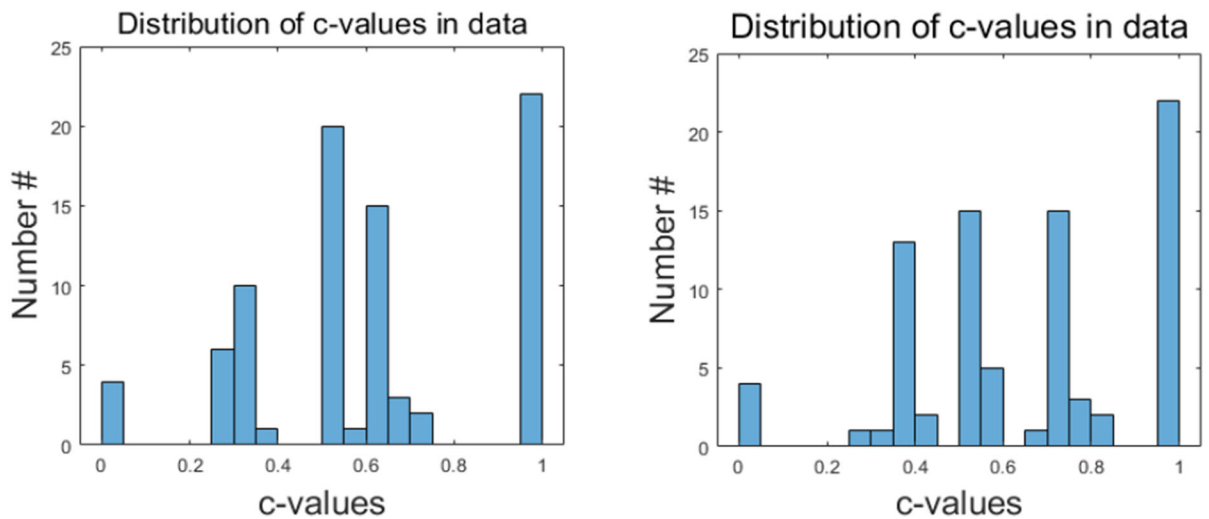
From Fig. 11, column (a) and column (b), we see that through the learning process the updated ensemble of flux functions centers around the true flux, both for  $f(c; \chi = 120)$  (upper row) and  $f(c; \chi = 200)$  (middle row) and identifies the part of the flux which is the harder one to learn. As for the case in Fig. 10, we see that it is challenging to capture the very precise shape of  $f(c)$  for  $c > 0.6$  (column c). The corresponding histograms are shown in Fig. 11d, e, and the lack of observation data for higher values of  $c$  is suggested as a natural reason for the poorer estimation of the flux functions in this region for both cases.

### 4.3 Temporal variation and mixing zone observation data can compensate for sparse distribution of $c$ -values

To what extent is there room for a good identification of the unknown flux function when the observed  $c$ -value distribution in the interval  $[0, 1]$  is sparse? In the



**Fig. 8** Illustration of the learning process for the case where synthetic data is generated from the flux function (16) where  $\chi \in [50, 300]$  combined with initial data (17). **a**  $f(c, \chi = 50)$ ; **b**  $f(c, \chi = 150)$ ; **c**  $f(c, \chi = 200)$ ; **d**  $f(c, \chi = 300)$



**Fig. 9** Distribution of the different  $c$ -values involved in the observed data used for the identification process of  $f(c; \chi = 200)$ . Left: Initial data  $c_0(x; 0.6, 0.3)$  based on (18). Right: Initial data  $c_0(x; 0.7, 0.4)$  based on (18)

following, we explore the effect of using three different observation data sets  $d_k^A, d_k^B$ , and  $d_k^C$  defined as follows:

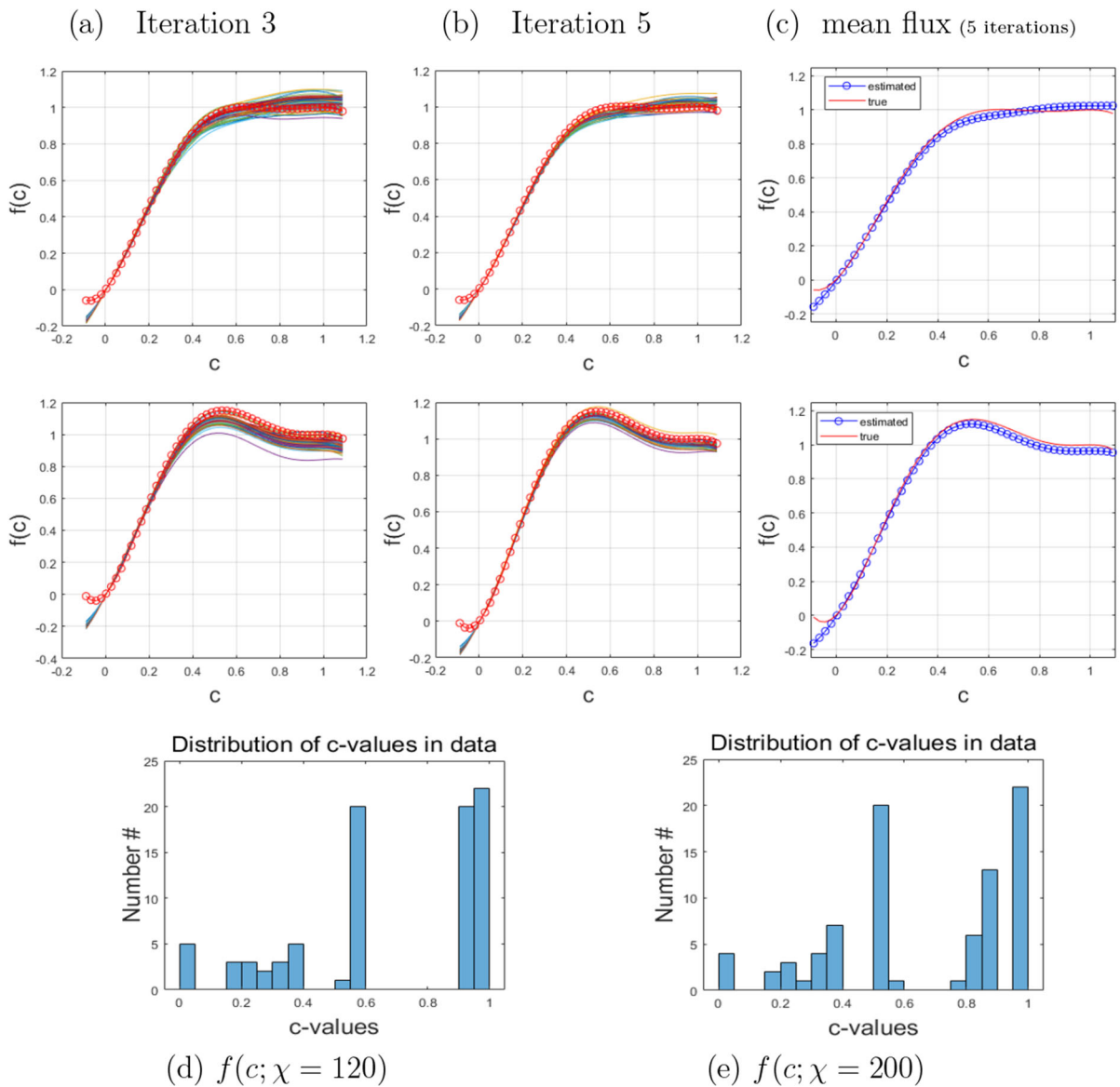
$$d_k^A = \{d_k \text{ given by (27) with } \lambda_1 = \lambda_2 = 1\}$$

$$d_k^B = \{d_k \text{ given by (27) with } \lambda_1 = 1, \lambda_2 = 0\}$$

$$d_k^C = \left\{ d_k \text{ given by (27) with } \lambda_1 = \lambda_2 = 1, \text{ however, (25) is replaced by } d_A^k(t_i^{\text{obs}}) = \left\{ c^k(x_1^*, t_i^{\text{obs}}), c^k(x_2^*, t_i^{\text{obs}}), c^k(x_3^*, t_i^{\text{obs}}), c^k(x_4^*, t_i^{\text{obs}}) \right\} \right\} \quad (32)$$

Hence,  $d_k^B$  ignores information about the mixing zone behavior as expressed by (26), whereas  $d_k^C$





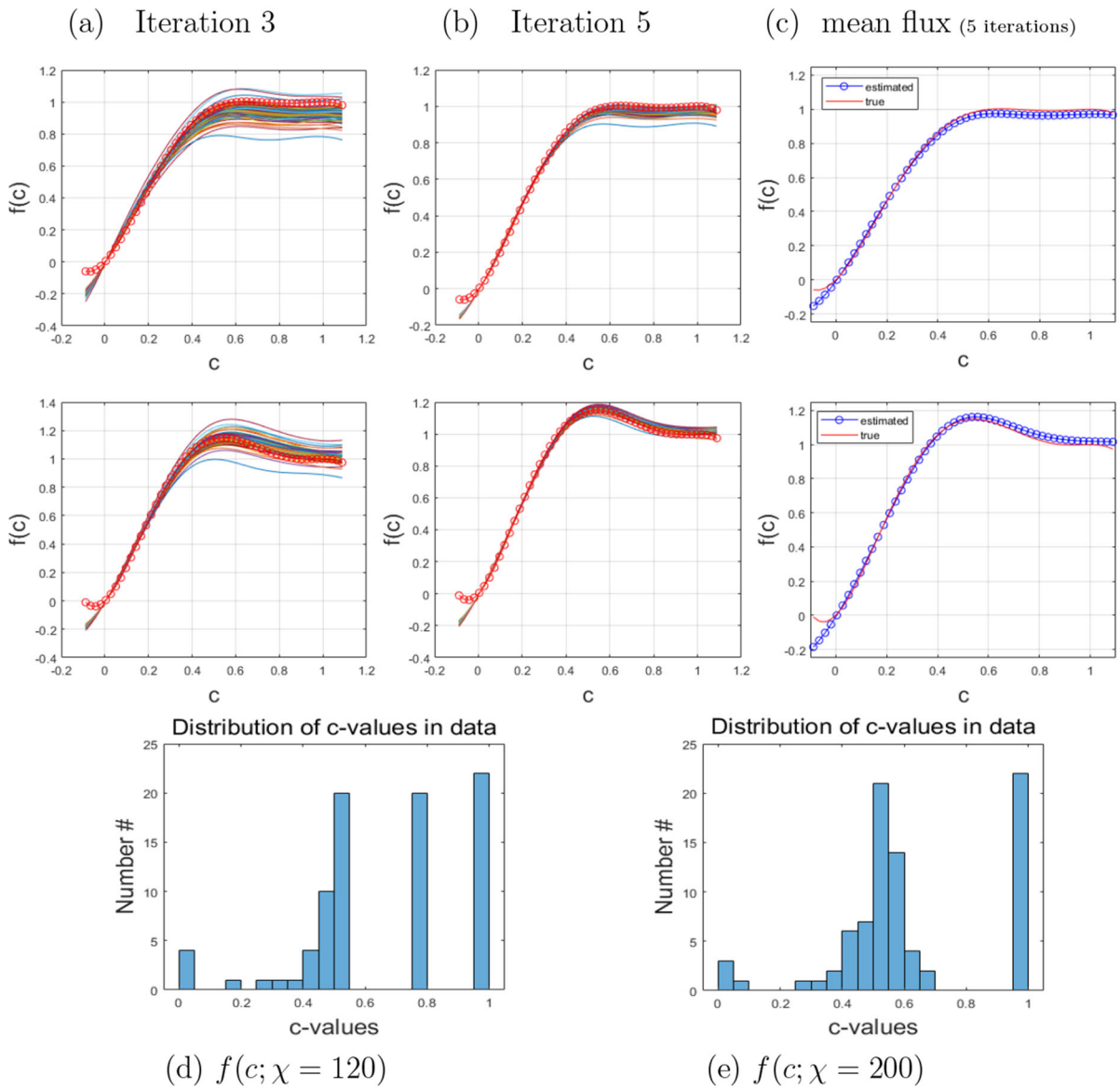
**Fig. 10** Illustration of the learning process where synthetic data are generated from the flux function (16) with  $\chi = 120$  (upper row) and  $\chi = 200$  (middle row) and the data set (27) is used in combination with (30). **a** Updated ensemble  $\{\theta_k^{(3)}\}$ . **b** Updated

ensemble  $\{\theta_k^{(5)}\}$ . **c** The mean  $\theta^* = \text{mean}\{\theta_k^{(5)}\}$ . Distribution of the different  $c$ -values involved in the observed data for **d** case with  $\chi = 120$ ; **e** case with  $\chi = 200$

ignores information about the temporal gradients in the time-dependent observation data as expressed by  $\Delta_t c^k(x_I^*, t_i^{\text{obs}}) = \frac{1}{2} (c^k(x_I^*, t_{i+1}^{\text{obs}}) - c^k(x_I^*, t_{i-1}^{\text{obs}})) / \Delta t^{\text{obs}}$  with  $I = 1, 2, 3, 4$ .

We want to explore the learning process based on, respectively,  $d_k^A$ ,  $d_k^B$ , and  $d_k^C$  for a case when the true flux function involves a rather deep convex “dip” for

smaller values of  $c$ . For that purpose, we consider  $f(c; \chi = -200)$  and we refer to Fig. 13 (third row) for illustration of the flux function. Clearly, this is a potential challenge for the identification process since shock wave solutions (decreasing jumps) depend on the upper concave envelope [4] which ignores information about finer details related to this convex “dip”.



**Fig. 11** Illustration of the learning process where synthetic data are generated from the flux function (16) with  $\chi = 120$  (upper row) and  $\chi = 200$  (middle row) and the data set (27) is used in combination with (31). **a** Updated ensemble  $\{\theta_k^{(3)}\}$ . **b** Updated

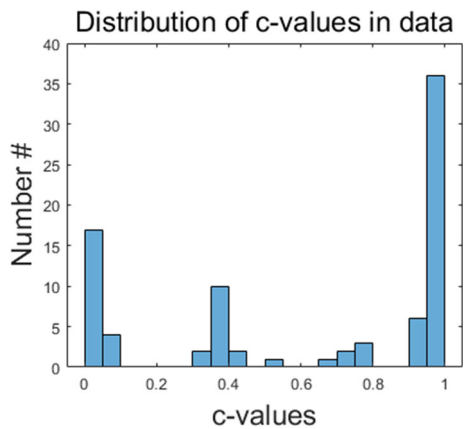
ensemble  $\{\theta_k^{(5)}\}$ . **c** The mean  $\theta^* = \text{mean}\{\theta_k^{(5)}\}$ . Distribution of the different  $c$ -values involved in the observed data for **d** case with  $\chi = 120$ ; **e** case with  $\chi = 200$

We consider the initial state

$$c_0(x) = \begin{cases} 1.0, & x \in [0, 2] \\ 0.4, & x \in (2, 3.5] \\ 0.2, & x \in (3.5, 5] \\ 0, & x \in (5, 10] \end{cases} \quad (33)$$

The resulting observation data distribution is shown in Fig. 12. It reveals that there is lack of observation data both for smaller  $c$ , intermediate, as well as higher.

In Fig. 13, the result of the identification process is shown for the different observation data sets. The result based on observation set  $d_k^A$  is shown in column (a), observation set  $d_k^B$  is shown in column (b), whereas column (c) reflects the learning process based on  $d_k^C$ . See also the figure text for further details of plots in the different panels. Main findings are: (i) The use of data set  $d_k^A$  allows one to identify most of the details of the



**Fig. 12** Distribution of observation data for the case with true flux function  $f(c; \chi = -200)$  combined with initial data (33)

unknown flux function except for the precise level for  $c$ -values close to the end point  $c = 1$ . We see that this learning process is associated with good match between predicted data and true observation data, see bottom panel of column (a); (ii) The price to pay for relying on observation data set  $d_k^B$ , which ignores observation data pertaining to the mixture zone behavior, is a dominating underestimation of the true flux function where details about the local variations for  $c \in [0, 0.3]$  is largely lost, see column (b). Nevertheless, the predicted observation data (bottom panel) largely match the true except from the time-dependent variation associated with  $x_4^*$ ; (iii) Finally, as seen in column (c), ignoring information about the variations in the time-dependent curves as represented by observation data set  $d_k^C$ , gives rise to a similar behavior as seen in column (b). However, now it results in a dominating overestimation of the true flux function. This also results in a poorer prediction of the observation data as seen in the bottom panel of column (c). This example illustrates how sparse distribution of observation data in  $[0, 1]$  can be compensated for through the inclusion of additional information related to the mixture zone behavior and temporal gradients in the observations over time, as represented by data set  $d_k^A$ .

### 5 What is the role played by the selected a priori representation of $f(c)$ ?

So far we have focused on investigations that can reveal insight into the interplay between the selected obser-

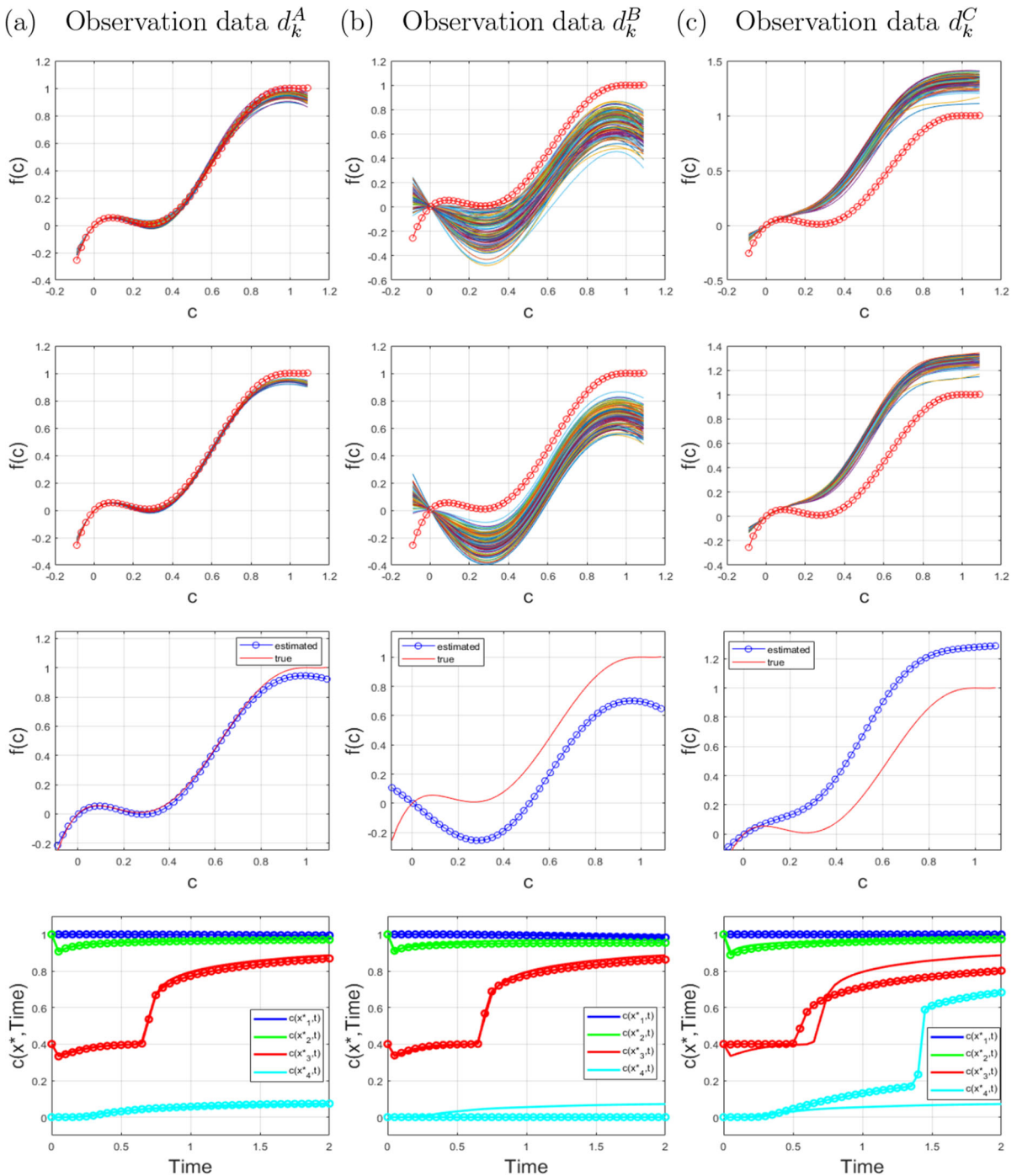
vation data and the possibility to identify the ground truth flux function  $f(c)$ . This issue has largely to do with the special nature of the conservation law (1) and the ambiguity between observation data and the underlying flux function. This is reflected by the entropy solution of (1) where the convex hull of the given flux is employed in the construction of the solution [4]. We first provide some further characterization of the entropy solution before we proceed with testing of different a priori processes to represent  $f(c)$ . For a discontinuity  $(c_l(t), c_r(t))$  moving along a certain path in the  $x - t$  space, the speed  $s$  of this jump is given by the Rankine–Hugoniot condition [4, 39]

$$s = \frac{f(c_l) - f(c_r)}{c_l - c_r}. \tag{34}$$

Furthermore, from (3) another equivalent formulation of the entropy condition can be derived for the valid discontinuity  $(c_l, c_r)$ . This states that for all numbers  $v$  between  $c_l$  and  $c_r$ , the following two inequalities must hold

$$\frac{f(v) - f(c_l)}{v - c_l} \geq s \geq \frac{f(v) - f(c_r)}{v - c_r} \tag{35}$$

where  $s$  is the shock speed velocity (34). It is well known that (8) generally will generate shock wave solutions  $c(x, t)$  in finite time, i.e., solutions that contain one or several discontinuities expressed as a jump  $(c_l, c_r)$  with  $c_l \neq c_r$ , despite the fact that initial data  $c_0(x)$  are smooth [4, 39]. In particular, the specific form of  $f(c)$  in the interval  $[\min(c_l, c_r), \max(c_l, c_r)]$  is not used in the construction of the entropy solution, only the slope  $s = \frac{f(c_l) - f(c_r)}{c_l - c_r}$ . As jumps arise and disappear in the solution over the time period for which observation data are collected, the data may lack information about  $f(c)$  [29, 40]. An illustration of this situation is given in Fig. 14. In the left panel, we plot the flux function  $f(c) = c^2 / (c^2 + (1 - c)^2)$ . In the right panel, the entropy solution after a time  $T = 0.5$  is shown. At time  $t = 0$ , the initial data  $c_0(x)$  involve one jump at  $x = 0$  and another jump at  $x = 1$ . The initial jump at  $x = 0$  is instantly transformed into a solution that is a combination of a continuous wave solution (rarefaction wave) and a discontinuous wave  $(c_l, c_r) \approx (0.3, 1.0)$ , as dictated by the lower convex envelope shown in left panel (green curve) [4]. Similarly, the initial jump at  $x = 1$  is transformed into a solution that is a combination of a continuous wave solution (rarefaction wave) and a discontinuous wave  $(c_l, c_r) \approx (0.7, 0)$ , in accordance with the upper concave envelope illustrated in left panel



**Fig. 13** Illustration of the learning process for the case where synthetic data are generated from the flux function (16) with  $\chi = -200$  combined with initial data (33). **Upper row.** Updated ensemble  $\{\theta_k^{(3)}\}$  after 3 iterations. **Second row.** Updated

ensemble  $\{\theta_k^{(5)}\}$  after 5 iterations. **Third row.** The mean  $\theta^* = \text{mean}\{\theta_k^{(5)}\}$ . **Bottom row.** Observed data versus predicted behavior. Solid line represents observed data, and circles show predicted result

(magenta curve) [4]. From this example, we see that we have no observation data that directly can reveal the shape of  $f(c)$  in the interval  $c \in [0.3, 0.7]$  (approximately). However, as demonstrated in Sect. 4 we can still obtain effective identification of the unknown flux function by careful combination of the type of observation data and sound regularity imposed on the a priori ensemble to represent  $f(c)$ .

In the following, we explore in more detail the role of the a priori distribution through the Gaussian process as described in Sect. 2.1. In particular, we will explore the effect of using an a priori ensemble  $\theta_k = \{\gamma_k, f_k\}$  for  $k = 1, \dots, N_{\text{ens}}$  that is less regular and/or contains stronger variations. In order to pay special focus on the role of the smoothness/regularity of  $f_k(c)$ , we fix  $\gamma_k = 1$ . For the investigations in this section, we employ as our testing ground the case with true  $f$  given by  $f(c; \chi = 300)$  and with initial data  $c_0(x; 0.6, 0.3)$ .

### 5.1 What is the effect of using a priori ensemble with less smoothness/regularity?

We tested use of a shorter correlation length  $l = 10\Delta c$  in (11), see Fig. 15, column (a). Results are comparable to that of using  $l = 20\Delta c$  and reported in Fig. 8 (right column). However, there is a mismatch of the identified flux versus the ground truth for  $c \in [0.4, 0, 6]$ . Looking at the discrepancy between predicted data and true data, we see that the error is very low in the interval  $t \in [0, 1]$  from which observation data are extracted. However, at later times  $t \in (1, 2]$  the consequences of the difference in learned flux versus the true comes to the surface as seen by the observation data at  $x_1^*$  and  $x_4^*$ . This shows that for the selected observation data, there is clearly room for this discrepancy between identified and true  $f$  without being revealed by the loss function error. A natural remedy would be to include more observation data.

The case with  $l = 5\Delta c$  in (11) (or lower values) becomes impractical to deal with since the different members of the a priori ensemble involve flux functions where  $f'(c)$  can take very high values. As seen from the description of the discrete scheme (14) and (15), it is clear that high values of  $f'$  put strong constraints on the size of the discretization step  $\Delta t$ . Noting that  $f'(c)$  has a clear physical interpretation in terms of representing the speed by which  $c$ -values travel with, it is reasonable to use this information when we decide

how to generate the a priori ensemble. For the fluid displacement problem, we know from the experimental set up that  $f'(c)$  cannot take such higher values and therefore, account for this information when we design the a priori process to represent  $f$ . This serves as a justification of using the a priori ensemble with higher correlation length.

What is the effect of using not so smooth a priori distribution, e.g., the Matérn kernel with  $\nu = 3/2$  or  $\nu = 5/2$ , whereas  $l = 20\Delta c$  as before? This test will be quite similar to the above case with shorter correlation length. The Matérn kernel with  $\nu = 5/2$  [41] takes the following form using the same notation as in (11)

$$C_{c_1, c_2}^{\nu=5/2} = \sigma^2 \left( 1 + \sqrt{5}(r/l) + 5(r^2/3l^2) \right) e^{-\sqrt{5}(r/l)},$$

$$r = \sqrt{(c_1 - c_2)^2}. \quad (36)$$

Results are shown in Fig. 15, column (b). We found that the identified flux  $f(c)$  contained a systematic difference to the true flux for  $c \in [0.4, 1.0]$ . However, looking at the predicted data curves (bottom row) we find that they largely fit the observed data curves in the time period  $t \in [0, 1]$ . This example again illustrates the challenge when the entropy solution, as characterized by (35) and illustrated in Fig. 14, is the source of the observation data. An improved learning of the flux function could most likely be obtained by incorporating more observation data that potentially can reveal the gap between the entropy solution generated by the learned and true flux function, respectively.

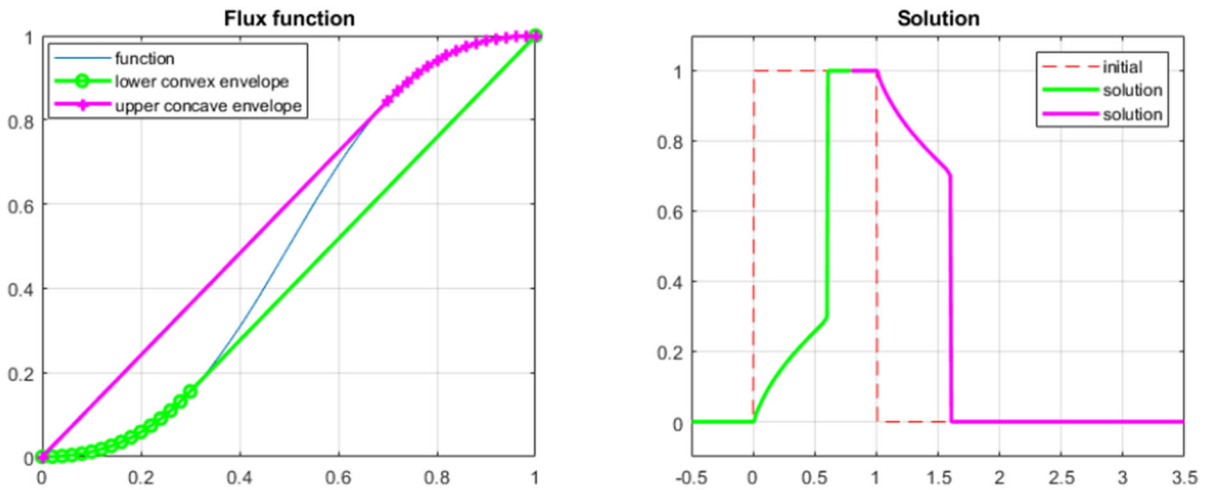
We also tested the use of a priori Matérn kernel with  $\nu = 3/2$

$$C_{c_1, c_2}^{\nu=3/2} = \sigma^2 \left( 1 + \sqrt{3}(r/l) \right) e^{-\sqrt{3}(r/l)},$$

$$r = \sqrt{(c_1 - c_2)^2}. \quad (37)$$

Results are illustrated in Fig. 15, column (c). The deviation between identified flux and true is clearly larger, whereas the difference between predicted behavior and observed data is still low in the time period  $[0, 1]$  (bottom row). This illustrates that adding more variations and/or less regularity to the a priori distribution may increase mismatch pertaining to the underlying flux functions without showing this in the loss function behavior. This is due to the construction of the entropy solution which only uses the convex hull of the involved  $f(c)$  as indicated by the example in Fig. 14.





**Fig. 14** Left: Example of nonlinear flux function  $f(c) = \frac{c^2}{c^2+(1-c)^2}$  (blue curve). Upper concave envelope (magenta curve) and lower convex envelope (green curve) are also included which

are identified by the entropy condition (35) as the part of the flux function involved in the construction of the solution for this initial data. Right: The solution of (8) at time  $T = 0.5$  is shown together with its initial data  $c_0(x)$  (red dashed line)

### 5.2 What is the role played by the size $N_{\text{ens}}$ of the ensemble $\{\theta_k\}_{k=1}^{N_{\text{ens}}}$ ?

We have consistently used an ensemble of size  $N_{\text{ens}} = 100$ . We also explore the identification by using a lower number  $N_{\text{ens}} = 50$  and  $N_{\text{ens}} = 25$ . We use the Gaussian process with  $l = 20\Delta c$  and explore identification for the same case as in Sect. 5.1. In Fig. 16, a comparison is made of the identification based on different ensemble numbers  $N_{\text{ens}} = 100$  (a),  $N_{\text{ens}} = 50$  (b), and  $N_{\text{ens}} = 25$  (c). Main observation is that reducing the ensemble number  $N_{\text{ens}}$  increases the width of the updated ensemble parameter vector representing the flux function, see column (b) and (c) versus column (a). The resulting mean flux function with lower  $N_{\text{ens}}$ , i.e., column (b) and (c), gives less accurate learning of the true flux function, suggesting that  $N_{\text{ens}} = 100$  is a proper choice for the problem under consideration.

### 5.3 How robust is the identification to larger noise in the observation data?

We explore how identification of the flux is affected when noise  $\eta$  in (4) is increased from 0.03 to 0.06, 0.09, and 0.15, respectively. We use the Gaussian process with  $l = 20\Delta c$  and explore identification for the same case as in Sect. 5.1. Results are shown in Fig. 17.

We see that increasing noise in the observation data is manifested in the updated parameter vector representing the flux through a wider band width. However, the mean of the updated parameter vector gives the same accurate identification, see third row. Finally, we also tested sensitivity to different realizations of the a prior ensemble generated to represent  $f(c)$  for the case with  $l = 20\Delta c$ . We found that the identified flux deviated minorly from one realization to another (results not shown).

### 5.4 Comparison with standard Monte Carlo Markov Chain (MCMC) algorithm

It is instructive and interesting to do a comparison of EnKF with the standard MCMC algorithm. Following [30,42], we use the Metropolis–Hastings method to approximate the posterior flux. Let

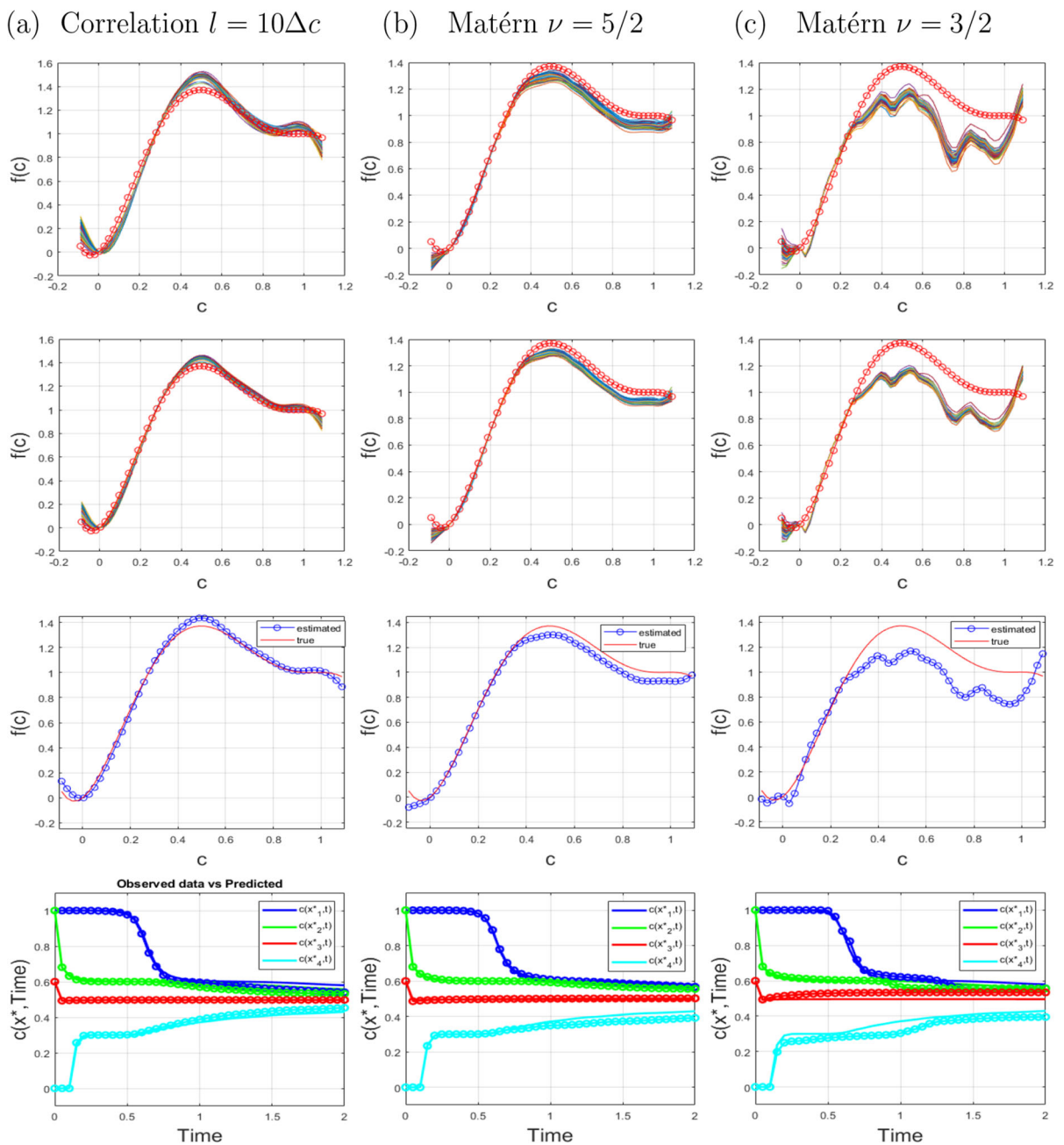
$$a(u, v) = \min\{1, \exp(I(u) - I(v))\} \tag{38}$$

where

$$I(u) = I(u; d) = \Phi(u; d) + \frac{1}{2} \|C^{-1/2}u\|^2,$$

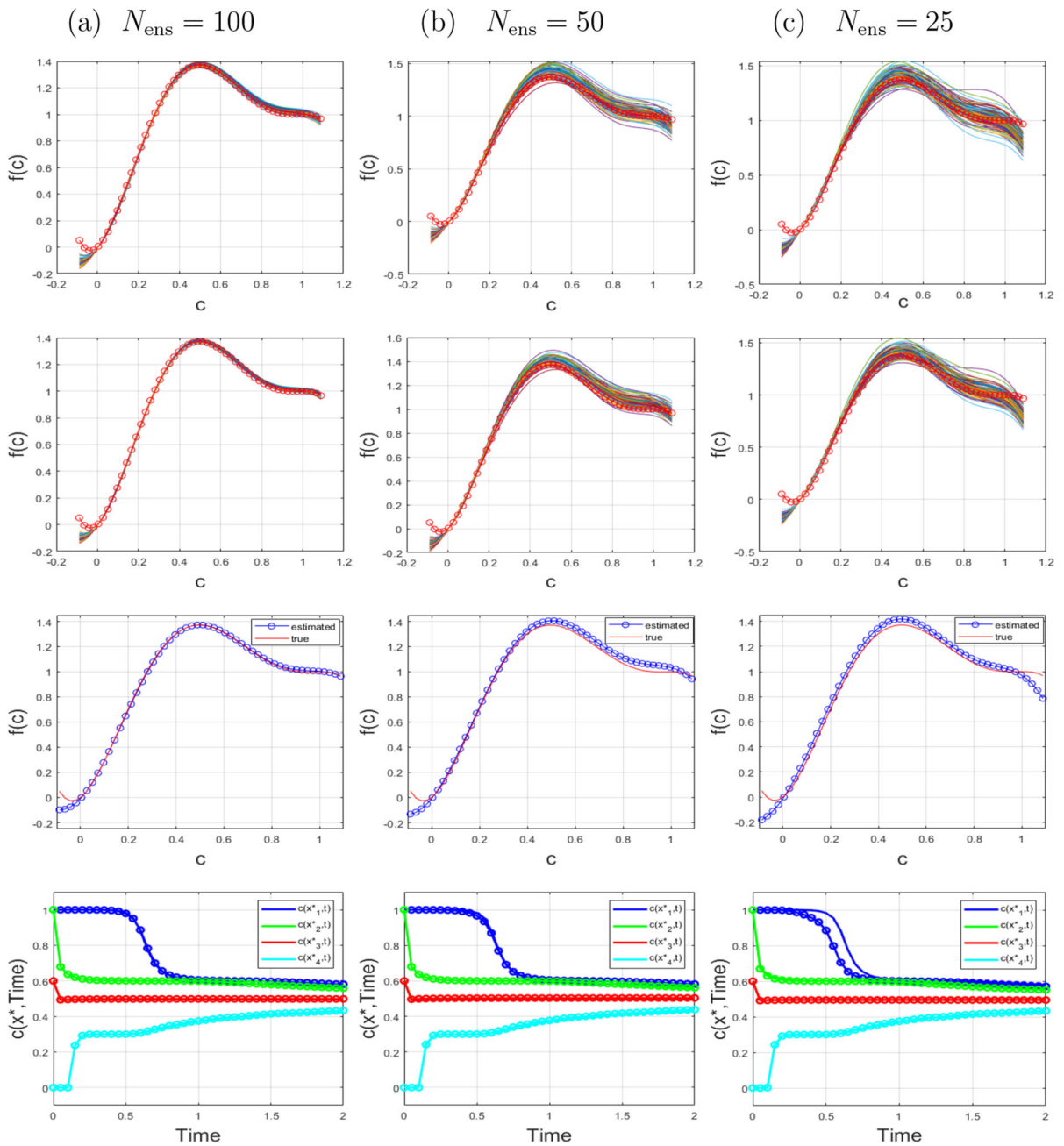
$$\Phi(u; d) = \frac{1}{2} \|\Gamma^{-1/2}(h(u) - d)\|^2.$$

Herein,  $\Gamma$  is the covariance matrix for the measurement error  $\eta \sim N(0, \gamma^2\mathcal{I})$  associated with observations  $d$



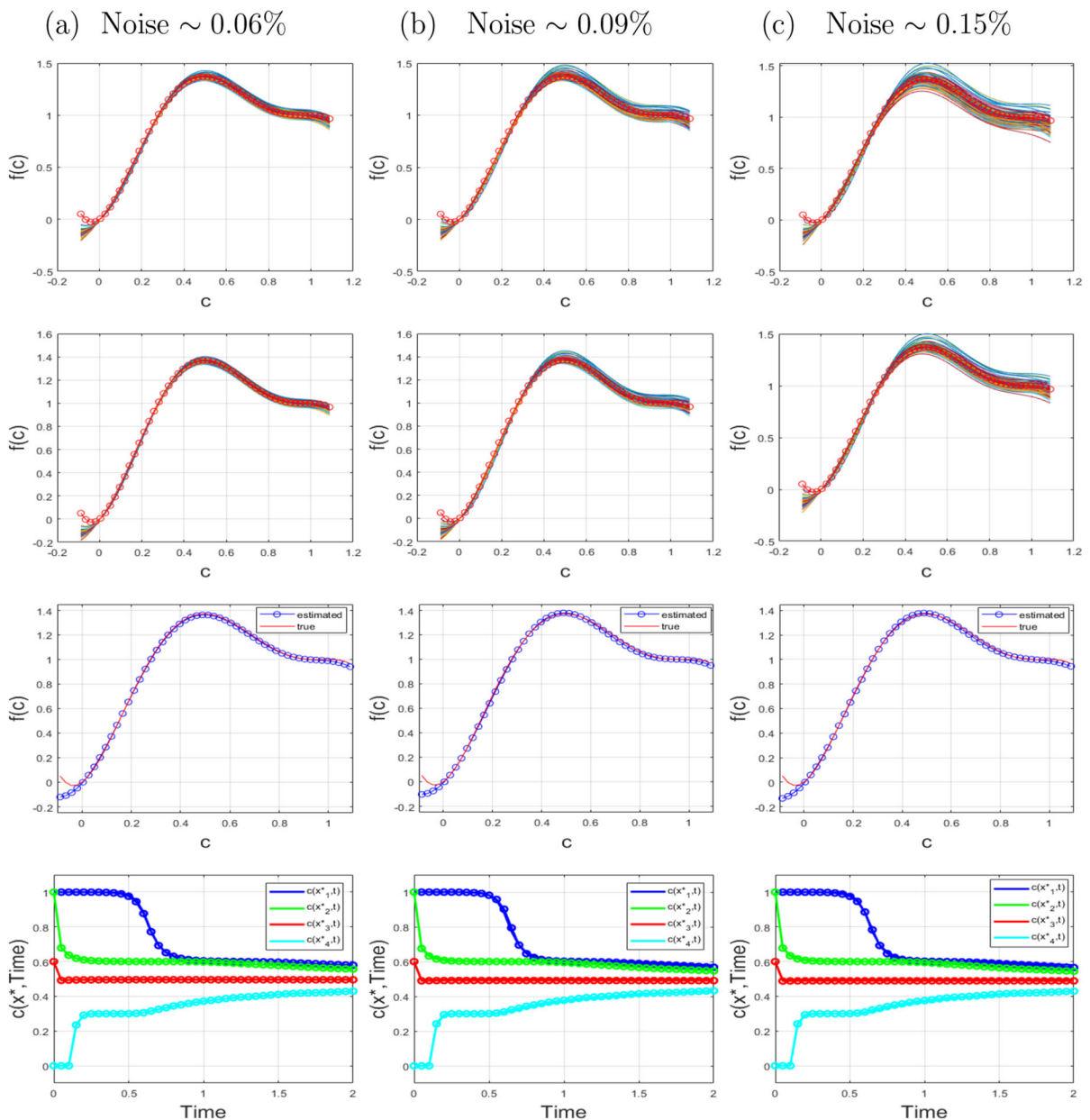
**Fig. 15** Illustration of the learning process for the case where synthetic data are generated from the flux function (16) with  $\chi = 300$  combined with initial data (18) corresponding to  $c_0(x, 0.6, 0.3)$ . The different columns refer to a priori ensemble distribution of different types: **a** Gaussian with correlation length  $l = 10\Delta c$ ; **b** Matérn kernel with correlation  $l = 20\Delta c$

and  $\nu = 5/2$ ; **c** Matérn kernel with correlation  $l = 20\Delta c$  and  $\nu = 3/2$ . Upper row. Updated ensemble  $\{\theta_k^{(3)}\}$  after 3 iterations. Second row. Updated ensemble  $\{\theta_k^{(5)}\}$  after 5 iterations. Third row. The mean  $\theta^* = \text{mean}\{\theta_k^{(5)}\}$ . Bottom row. Observed data versus predicted behavior. Solid line represents observed data, and circles show predicted result



**Fig. 16** Illustration of the learning process for the case where synthetic data are generated from the flux function (16) with  $\chi = 300$  combined with initial data (18) corresponding to  $c_0(x, 0.6, 0.3)$ . The different columns refer to a priori ensemble distribution of different number: **a**  $N_{\text{ens}} = 100$ ; **b**  $N_{\text{ens}} = 50$ ;

**c**  $N_{\text{ens}} = 25$ . Upper row. Updated ensemble  $\{\theta_k^{(3)}\}$  after 3 iterations. Second row. Updated ensemble  $\{\theta_k^{(5)}\}$  after 5 iterations. Third row. The mean  $\theta^* = \text{mean}\{\theta_k^{(5)}\}$ . Bottom row. Observed data versus predicted behavior



**Fig. 17** Synthetic data are generated from the flux function (16) with  $\chi = 300$  and initial data (18) corresponding to  $c_0(x, 0.6, 0.3)$ . The different columns refer to different levels of noise in observation data: **a** Noise  $\sim 0.06\%$ ; **b** Noise  $\sim 0.9\%$ ;

**c** Noise  $\sim 0.15\%$ . Upper row. Updated ensemble  $\{\theta_k^{(3)}\}$  after 3 iterations. Second row. Updated ensemble  $\{\theta_k^{(5)}\}$  after 5 iterations. Third row. The mean  $\theta^* = \text{mean}\{\theta_k^{(5)}\}$ . Bottom row. Observed data versus predicted behavior

in (4) where  $\gamma = 0.03$  and  $\mathcal{I}$  is the identity matrix. For the sake of comparison, we focus on the the same case as explored in Sect. 5.1. In particular, we use the Gaussian process with  $l = 20\Delta c$  for the initial a priori

distribution to represent  $f$  as well as for the random walk update step. The MCMC algorithm is then given by the following standard random walk steps:

1. Set  $k = 0$  and select  $f^{(0)}(c_i)$  from a priori distribution  $N(0, \mathcal{C})$  where  $\mathcal{C}$  amounts to (11) with  $l = 20\Delta c$  and  $\sigma = 0.25$ .
2. Propose  $g^{(k)} = f^{(k)} + \beta\xi^{(k)}$ ,  $\xi^{(k)} \sim N(0, \mathcal{C})$  where  $\mathcal{C}$  amounts to (11) with  $l = 20\Delta c$  and  $\sigma = 0.25$ .
3. Compute  $a(f^{(k)}, g^{(k)})$ . Then, draw a number  $i^{(k)}$  in  $[0, 1]$  from a uniform distribution.  
If  $i^{(k)} \leq a(f^{(k)}, g^{(k)})$ , then we set  $f^{(k+1)} = g^{(k)}$ .
4. Set  $f^{(k+1)} = f^{(k)}$  otherwise.
5.  $k \rightarrow k + 1$

The algorithm has three scalar hyperparameters  $(\beta, b, \tau)$  which need to be specified [30,42]. First, the stepsize  $\beta$  which controls the size of the move, second the burn-in  $b$ , i.e., the number of samples which are discarded in order to minimize the contribution of the initial value  $f^0$ , and the sample interval  $\tau$  which is the number of states which are discarded between two observations. For the presented simulation results shown in Fig. 18, we have used  $(\beta, b, \tau) = (0.05, 400, 20)$ . After the burn-in period, we generate a Markov chain of length 2000 and obtain an ensemble of 100 samples using the sample interval  $\tau = 20$ , see Fig. 18a. The main finding is that MCMC gives mean posterior flux (Fig. 18b) which is essentially of a similar quality as the one EnKF gave, see for example, Fig. 16 (left column). In particular, the predicted data shown in panel 18c are captured well. The main difference is that MCMC has involved 2400 simulation steps, whereas EnKF has employed 600. We may expect that results are to some extent sensitive to different parameters. That is, other choices of a priori distribution and/or covariance structure of  $\xi^{(k)}$  most likely could have some impact on the result. E.g., a higher burn-in value  $b$  would ensure that a larger portion of the posterior samples lies closer to the true flux.

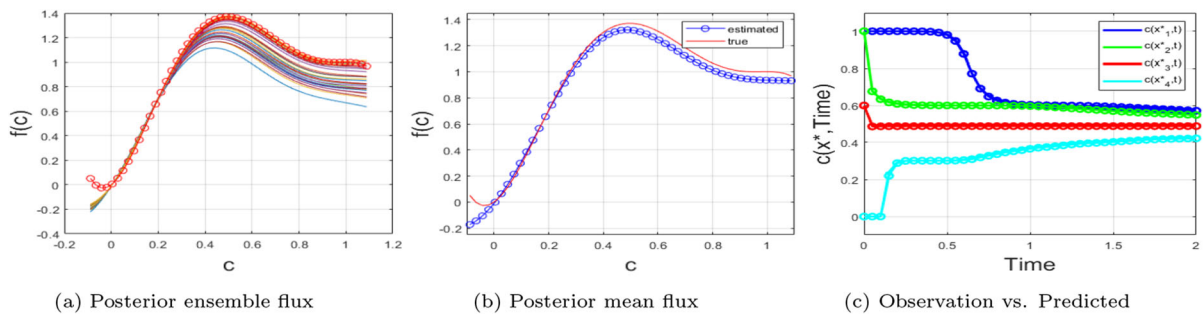
## 6 Conclusion

In this work, we have explored a methodology that combines representing the unknown flux function as a Gaussian random process through a parameter vector and updating this parameter vector based on the ESMDA method. The physical setting we consider is displacement of one fluid by another in a vertical domain. The parameter  $\chi$  is a key parameter that expresses the ratio of buoyant to viscous stresses. As indicated in Sect. 2, it is possible to derive a nonlinear conservation

law of the form (8) where a functional form  $f(c; \chi)$  of the flux function is obtained. Depending on the parameter  $\chi$ , different flow regimes are involved ranging from stable to density-unstable displacement. Our aim has been to explore an ensemble-based approach as a means to extract from observation data the unknown nonlinear flux function. Main findings from the investigations are:

- By relying on a sparse set of time-dependent data extracted from a few fixed positions in space, combined with some care in the choice of the initial state, we can to a large extent provide an effective identification of the nonlinear unknown flux functions  $f(c; \chi)$  for different flow regimes corresponding to  $\chi \in [-200, 300]$ . This includes highly non-convex flux functions.
- The use of Gaussian random fields to represent the unknown flux function  $f(c)$  for  $c \in [0, 1]$  adds additional regularity and information which is exploited in the ensemble-based learning process. That is, we have seen that a relative effective identification of  $f(c)$  is possible despite the fact that the time series observation data only partly cover the domain  $[0, 1]$  on which  $f(c)$  is described (due to the occurrence of discontinuities in the solutions). We have found that this is a direct consequence of the inclusion of additional observation data related to temporal variations, as expressed by the second part (second line) in (25), and information about the front and back position of the mixing zone as expressed by (26).
- Further investigations showed that the regularity of the a priori ensemble distribution was quite essential for a good identification in the current study where a sparse amount of observation data was assumed available. The assumed smoothness of the a priori ensemble we applied is reasonable from the setup of the physical process under consideration. By including more observation data, one may expect to see higher robustness with respect to a prior regularity/smoothness. More precise understanding of this topic seems to be an interesting direction for further research. The identification was robust regarding level of noise in the observation data and to a little extent sensitive to different realizations of the a prior ensemble generated to represent  $f(c)$ . Finally, comparison with a standard MCMC algorithm to compute posterior flux





**Fig. 18** Results obtained by using MCMC. Synthetic data are generated from the flux function (16) with  $\chi = 300$  and initial data (18) corresponding to  $c_0(x, 0.6, 0.3)$ . **a** A posterior flux.

**b** Mean of a posterior flux. **c** Observed data versus predicted behavior. Solid line represents observed data, and circles show predicted result

confirmed the effectiveness of EnKF as a reliable method to identify the unknown flux function for the chosen set of observation data.

In conclusion, we have seen that the ensemble-based method explored in this work provides a practical tool for recovering an unknown nonlinear conservation law from observation data through point observations extracted over a time period. This setting is realistic in view of the fact that sensor data are commonly obtained in terms of time series. In this study, the choice of the observation data was motivated by the specific fluid mechanical process under consideration. More generally, there seems to be many unexplored aspects of the use of Bayesian methods to learn a nonlinear conservation law from observations due to the entropy solution concept which depends on the underlying flux function in a highly non-trivial manner.

**Funding** Open access funding provided by University of Stavanger & Stavanger University Hospital. The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

**Data Availability** All data generated or analyzed during this study are included in this article.

#### Declarations

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

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or

other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

#### Appendix A. A nonlinear conservation law for viscous displacements based on lubrication scaling

For vertical displacements that are dominated by viscous and buoyant stresses, a reduced displacement model may be derived by suitable scaling of the variables in the governing Eqs. (5)–(7). To this end, we consider the model problem shown in Fig. 1, where the displacing fluid advances vertically downward through the center of a plane channel with a gap of width  $D$ . This model geometry is analogous to that of a perfectly vertical, concentric annulus. We assume no flow in the azimuthal direction, and consider the displacement as effectively two-dimensional, i.e., along the axial and radial directions only. This is considered applicable for the vertical and concentric annulus, and for “stable” displacements that are density-stable or dominated by viscous stresses. To this end, a rectangular coordinate system is oriented as indicated in Fig. 1, with the origin in the center of the channel and the  $x$ -axis parallel to the direction of imposed flow. The displacing fluid is injected at a constant volumetric rate  $q$  per circumferential length of the channel, corresponding to a bulk axial velocity  $U^* = q/D$  in the gap. As pointed out above, the displacing and the displaced fluids are assumed to

be constant viscosity fluids, with a common viscosity  $\mu$ . Further, the fluids are assumed to be incompressible, with constant mass densities  $\rho_1$  and  $\rho_2$  for the displacing and displaced fluid, respectively. In what follows, a subscript  $i = 1$  refers to displacing fluid and  $i = 2$  to the displaced fluid, in accordance with the above and Fig. 1.

A lubrication scaling equal to that by Lajeunesse et al. [43] is next invoked, where it is assumed that the flow is essentially unidirectional and parallel to the walls of the geometry, and that the gap width  $D$  is small compared to a characteristic axial length scale of the duct,  $L^*$ , [44]. Defining  $\delta = D/L^*$  as the aspect ratio of the channel, the above assumption implies that  $\delta \ll 1$ . The problem is non-dimensionalized by taking  $L^*$  and  $D$  as length scales for the  $x$  and  $y$  coordinate, respectively, and using  $U^*$  as a characteristic scale for the velocity component in the vertical direction. Then,  $t^* = L^*/U^*$  is a natural time scale for the displacement process. As per Lajeunesse et al., we focus on displacements that occur at sufficiently high flow rates for diffusion to be negligible, i.e., it is assumed that  $Pe = \delta U D / \mathcal{D} \gg 1$ , with  $Pe$  the Péclet number. Further,  $\tau^* = \mu U^*/D$  is used as characteristic scale for viscous stresses. The pressure is non-dimensionalized by the viscous pressure scale  $P^* = \tau^*/\delta$ , [44]. Taking the fluids to be incompressible and denoting by  $u^x$  and  $u^y$  the axial and transverse dimensionless velocity components, one can show that the zero-order governing equations are

$$\partial_x u^x + \partial_y u^y = 0 \tag{A.1a}$$

$$-\partial_x p + \partial_y^2 \partial_y u^x + \phi_i \frac{Re}{Fr^2} = O(\delta^2, \delta Re) \tag{A.1b}$$

$$-\partial_y p = O(\delta^2, \delta^3 Re) \tag{A.1c}$$

$$\partial_t C + \partial_x(u^x C) + \partial_y(u^y C) = 0. \tag{A.1d}$$

Here,  $Re = \bar{\rho} U^* D / \mu$  is a Reynolds number based on the average fluid mass density,  $\bar{\rho} = (\rho_1 + \rho_2)/2$ , and  $Fr^2 = U^{*2}/(AtgD)$  is the densimetric Froude number expressing the ratio of inertial and buoyant stresses. The Atwood number  $At = (\rho_1 - \rho_2)/(\rho_1 + \rho_2)$  is introduced in the definition of the Froude number, which can be either positive or negative depending on whether the displacement flow is density-unstable or density-stable. Finally,  $\phi_1 = 1$  and  $\phi_2 = -1$ . Averaging the convection Eq. (A.1d) across  $y$ , i.e., between the walls of the channel, while assuming a sharp and single-valued position of the interface, a hyperbolic conser-

vation equation is found for the average fluid concentration  $c$ :

$$c(x, t)_t + \partial_x \int_{-h}^h u^x dy \equiv c(x, t)_t + f(c; \chi, m)_x = 0, \tag{A.2}$$

where it is seen that the flux function is equal to the flux of the *displacing* fluid. Lajeunesse et al. [43] and Amiri et al. [45] obtained an analytical expression for the flux function by solving Eq. (A.1b) for the velocity  $u^x$ , with the result being

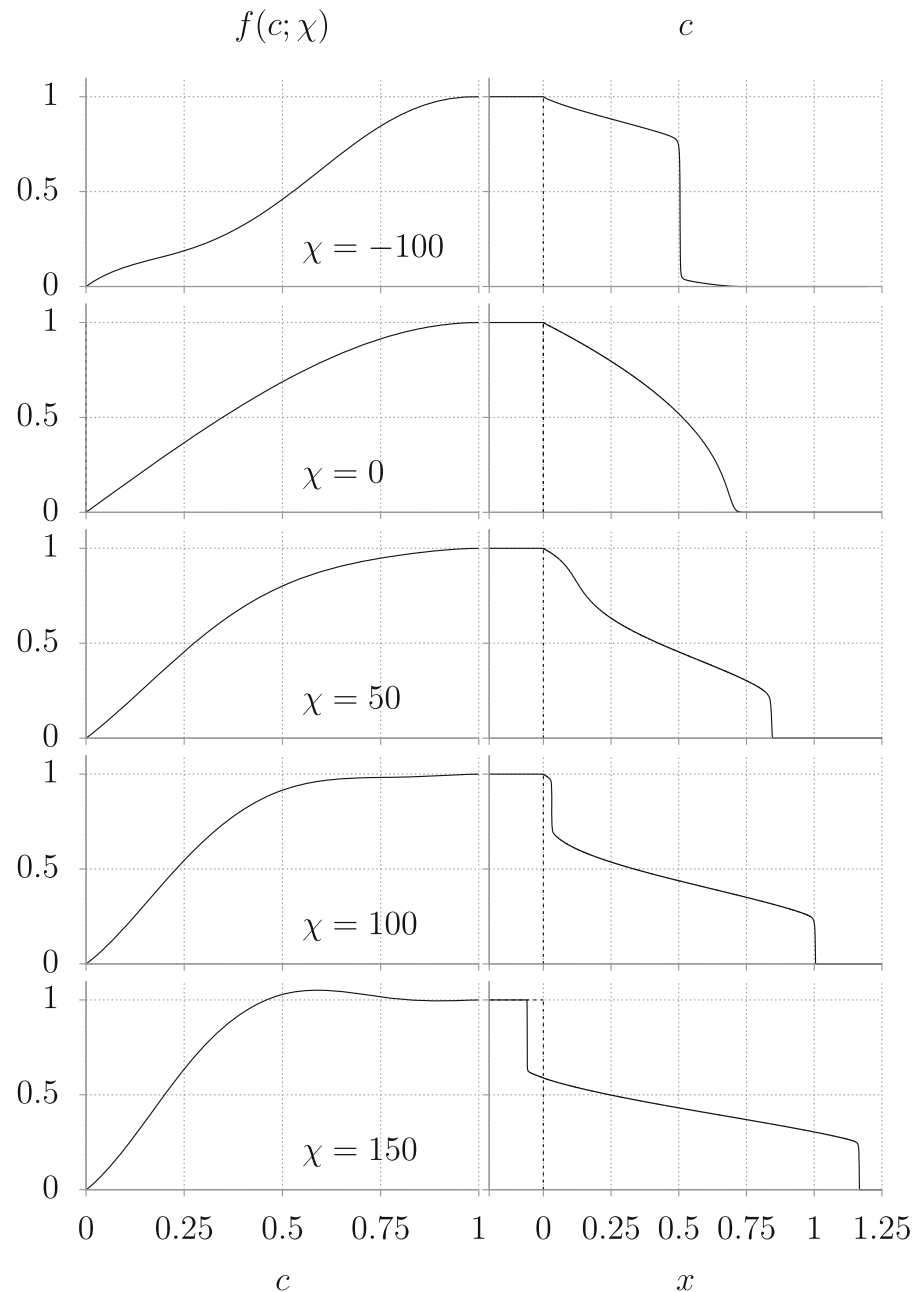
$$f(c; \chi) = \frac{c}{2}(3 - c^2) + \frac{\chi c^2}{12} \left( \frac{3}{4} - 2c + \frac{3}{2}c^2 - \frac{1}{4}c^4 \right). \tag{A.3}$$

Here,  $\chi = 2Re/Fr^2$  which is the ratio of buoyant to viscous stress scales. We note that Lajeunesse et al. accounted for different viscosity fluids and derived a more general form of the flux function in Eq. (A.3). Finally, we point out that a generalization to displacements involving pairs of yield stress and shear thinning fluids was presented by Zare et al. [46].

In Fig. 19, the impact of varying the sign and magnitude of  $\chi$  on the flux function and the characteristic interface profile is illustrated.

For iso-dense fluids,  $\chi = 0$ , and this limit is equivalent to that of single-phase flow, since now both density and viscosity are equal for the two fluids. The flux function has  $f'' < 0$  for all  $c$ , resulting in a rarefaction wave that is advected according to the Newtonian velocity profile in a plane channel. Density-stable displacements were considered by Lajeunesse et al. [43] and are illustrated in the top row of Fig. 19 for the case of  $\chi = -100$ . The flux function now contains a convex region for intermediate values of  $c$ , and this manifests in a jump discontinuity (shock) in the profile  $c$ , seen in the top right panel of Fig. 19. For a positive nonzero value of  $\chi$ , the flux function is convex in the lower end of the  $c$  axis and, for increasing values of  $\chi$ , eventually develops a second convex region at the opposite side of the  $c$  axis. The resulting interface profile therefore consists of either one or two jump discontinuities, as shown in the right panels in Fig. 19. For  $\chi \approx 117.8$ , the second jump discontinuity becomes stationary, due to the exact balance of buoyant and viscous stresses. At larger values of  $\chi$ , buoyant stresses are sufficiently

**Fig. 19** Flux functions and corresponding concentration profiles for iso-viscous displacements, and different density differences between the fluids. The density-unstable configuration results in a discontinuity in  $c$  for low values of the concentration. This corresponds to fast propagation of displacing fluid through the center of the channel. At larger density differences, a second discontinuity emerges at high values of  $c$ . For  $\chi > 117.8$ , the second discontinuity reverses direction, resulting in back-flow of the less dense fluid



large to overcome the viscous drag of the displacing fluid, resulting in reverse flow of the lighter fluid. This is illustrated for  $\chi = 150$  in Fig. 19.

## References

1. Berg, J., Nystrom, K.: Data-driven discovery of PDEs in complex datasets. *J. Comput. Phys.* **384**, 239–252 (2019)
2. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **378**, 686–707 (2019)
3. Long, Z., Lu, Y., Dong, B.: PDE-Net 2.0: Learning PDEs from data with a numeric-symbolic hybrid deep network. *J. Comput. Phys.* **399**, 108925 (2019)

4. LeVeque, R.J.: *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics, Cambridge (2007)
5. Hesthaven, J.S.: *Numerical Methods for Conservation Laws from Analysis to Algorithms*. Computational Science & Engineering, SIAM, Philadelphia (2017)
6. Kröene, D.: *Numerical Schemes for Conservation Laws*. Wiley-Teubner Series Advances in Numerical Mathematics, Wiley, New York (1997)
7. Emerick, A.A., Reynolds, A.C.: Ensemble smoother with multiple data assimilation. *Comput. Geosci.* **55**, 3–15 (2013)
8. Oliver, D.S., Fossun, K., Bhakta, T., Sandø, I., Nævdal, G., Lorentzen, R.J.: 4D seismic history matching. *J. Petrol. Sci. Eng.* **207**, 109119 (2021)
9. Evensen, G.: Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *J. Geophys. Res.* **99**(C5), 10143–10162 (1994)
10. Nævdal, G., Mannseth, T., Vefring, E.H.: Near-well reservoir monitoring through ensemble Kalman filter. In *SPE/DOE Improved Oil Recovery Symposium*, Tulsa, Oklahoma, SPE75235 (April 2002)
11. Haugen, V., Nævdal, G., Natvik, L.J., Evensen, G., Berg, A., Flornes, K.: History matching using the ensemble Kalman filter on a North Sea field case. *SPE J.* **13**, 382–391 (2008)
12. Bianco, A., Cominelli, A., Dovera, L., Nævdal, G., Vallès, B.: History matching and production forecast uncertainty by means of the ensemble Kalman filter: A real field application. In *SPE Europe/EAGE Annual Conference and Exhibition*, London, UK, 11–14 (June 2007). Society of Petroleum Engineers. SPE107161
13. Skjervheim, J.-A., Evensen, G., Hove, J., Vabø, J.G.: An ensemble smoother for assisted history matching. In *SPE Reservoir simulations symposium*, The Woodlands, Texas, (February 21–23 2011). SPE141929-MS
14. Chen, Yan, Oliver, Dean: Ensemble randomized maximum likelihood method as an iterative ensemble smoother. *Math. Geosci.* **44**(1), 1–26 (2012)
15. Chen, Yan, Oliver, Dean S.: Levenberg-Marquardt forms of the iterative ensemble smoother for efficient history matching and uncertainty quantification. *Comput. Geosci.* **17**, 689–703 (2013)
16. Iglesias, M.A., Law, K.J.H., Stuart, A.M.: Ensemble Kalman methods for inverse problems. *Inverse Probl.* **29**, 045001 (2013)
17. Iglesias, M.A.: A regularizing iterative ensemble Kalman method for PDE-constrained inverse problems. *Inverse Probl.* **32**, 025002 (2016)
18. Chada, N.K., Iglesias, M.A., Roininen, L., Stuart, A.M.: Parameterizations for ensemble Kalman inversion. *Inverse Probl.* **34**, 055009 (2018)
19. Herty, M., Visconti, G.: Continuous limits for constrained ensemble Kalman filter. *Inverse Probl.* **36**, 075006 (2020)
20. Kang, H., Tanuma, K.: Inverse problems for scalar conservation laws. *Inverse Probl.* **21**(3), 1047–1059 (2005)
21. James, F., Sepúlveda, M.: Convergence results for the flux identification in a scalar conservation law. *SIAM J. Control. Optim.* **37**(3), 869–891 (1999)
22. Castro, C., Zuazua, E.: Flux identification for 1-d scalar conservation laws in the presence of shocks. *Math. Comput.* **80**(276), 2025–2070 (2011)
23. Holden, H., Priuli, F.S., Risebro, N.H.: On an inverse problem for scalar conservation laws. *Inverse Probl.* **30**, 035015 (2014)
24. Bustos, M.C., Concha, F., Bürger, R., Tory, E.M.: *Sedimentation and Thickening-Phenomenological Foundation and Mathematical Theory*. Kluwer Academic Publishers, Dordrecht (1999)
25. Diehl, S.: Estimation of the batch-settling flux function for an ideal suspension from only two experiments. *Chem. Eng. Sci.* **62**, 4589–4601 (2007)
26. Bürger, R., Diehl, S.: Convexity-preserving flux identification for scalar conservation laws modelling sedimentation. *Inverse Prob.* **29**, 045008 (2013)
27. Betancourt, F., Bürger, R., Diehl, S., Mejías, C.: Advanced methods of flux identification for clarifier-thickener simulation models. *Miner. Eng.* **63**, 2–15 (2014)
28. Bürger, R., Careaga, J., Diehl, S.: Flux identification of scalar conservation laws from sedimentation in a cone. *IMA J. Appl. Math.* **83**, 526–552 (2018)
29. Li, Q., Evje, S.: Learning the nonlinear flux function of a hidden scalar conservation law from data. *Netw. Heterog. Media* **18**, 48–79 (2023)
30. Mishra, S., Ochsner, D., Ruf, A.M., Weber, F.: Well-Posedness of Bayesian Inverse Problems for Hyperbolic Conservation Laws. *Seminar for Applied Mathematics*, ETH, 24, (2021)
31. Duong, D. L.: *Inverse problems for hyperbolic conservation laws. A Bayesian approach*. University of Sussex PhD thesis, (2020)
32. Nelson, E.B., Guillot, D. (eds.): *Well Cementing*, 2nd edn. Schlumberger, Sugar Land (2006)
33. Skadsem, H.J., Kragset, S.: A numerical study of density-unstable reverse circulation displacement for primary cementing. *J. Energy Resour. Technol.* **144**, 123008 (2022)
34. Vetra-Cervalho, S., van Leeuwen, P.J., Nerger, L., Barth, A., Altaf, M.U., Brasseur, P., Kirchgessner, P., Beckers, J.-M., Tellus, A.: State-of-the-art stochastic data assimilation methods for high-dimensional non-Gaussian problems. *Dyn. Meteorol. Oceanogr.* **70**(1), 1445364 (2018)
35. Bishop, C.H., Etherton, B.J., Majumdar, S.J.: Adaptive sampling with the ensemble transform Kalman filter. part I: Theoretical aspects. *Mon. Weather Rev.* **129**, 420–436 (2001)
36. Hunt, B.R., Kostelich, E.J., Szunyogh, I.: Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter. *Physica D* **230**, 112–126 (2007)
37. Livings, David M., Dance, Sarah L., Nichols, Nancy K.: Unbiased ensemble square root filters. *Physica D* **237**, 1021–1028 (2008)
38. Sakov, P., Oke, P.R.: Implications of the form of the ensemble transformation in the ensemble square root filters. *Mon. Wea. Rev.* **136**, 1042–1053 (2008)
39. Holden, H., Risebro, N.H.: *Front Tracking for Hyperbolic Conservation Laws*. Springer, Berlin (2011)
40. Diehl, Stefan: Numerical identification of constitutive functions in scalar nonlinear convection-diffusion equations with application to batch sedimentation. *Appl. Numer. Math.* **95**, 154–172 (2015)
41. Rue, H., Held, L.: *Gaussian Markov Random Fields. Theory and Applications*. Chapman & Hall/CRC, Boca Raton (2005)

42. Cotter, S.L., Roberts, G.O., Stuart, A.M., White, D.: MCMC methods for functions: Modifying old algorithms to make them faster. *Stat. Sci.* **28**, 424–446 (2013)
43. Lajeunesse, E., Martin, J., Rakotomalala, N., Salin, D., Yortsos, Y.C.: Miscible displacement in a Hele-Shaw cell at high rates. *J. Fluid Mech.* **398**, 299–319 (1999)
44. Gary Leal, L.: *Advanced Transport Phenomena: Fluid Mechanics and Convective Transport Processes*. Cambridge Series in Chemical Engineering, Cambridge University Press, Cambridge (2007)
45. Amiri, A., Larachi, F., Taghavi, S.M.: Buoyant miscible displacement flows in vertical pipe. *Phys. Fluids* **28**(10), 102105 (2016)
46. Zare, M., Roustaei, A., Frigaard, I.A.: Buoyancy effects on micro-annulus formation: density stable displacement of Newtonian-Bingham fluids. *J. Nonnewton. Fluid Mech.* **247**, 22–40 (2017)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.