**Computing and Visualization in Science manuscript No.**
(will be inserted by the editor)

# Least Squares Approach for Initial Data Recovery in Dynamic Data-Driven Applications Simulations

**C. Douglas[1,2], Y. Efendiev[3,4], R. Ewing[3,4], V. Ginting[4], R. Lazarov[3,4], M. Cole[5], and G. Jones[5] ⋆**

[1] Department of Computer Science, University of Kentucky, Lexington, KY 40506-0045

[2] Department of Computer Science, Yale University, New Haven, CT 06520-8285

[3] Department of Mathematics, Texas A & M University, College Station, TX 77843-3404

[4] Institute for Scientific Computation, Texas A & M University, College Station, TX 77843-3404

[5] Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, UT 84112

The date of receipt and acceptance will be inserted by the editor

**Abstract.** In this paper, we consider the initial data recovery and the solution update based on the local measured data that are acquired during simulations. Each time new data is obtained, the initial condition, which is a representation of the solution at a previous time step, is updated. The update is performed using the least squares approach. The objective function is set up based on both a measurement error as well as a penalization term that depends on the prior knowledge about the solution at previous time steps (or initial data). Various numerical examples are considered, where the penalization term is varied during the simulations. Numerical examples demonstrate that the predictions are more accurate if the initial data are updated during the simulations.

**Key words** Initial data recovery, dynamic data-driven applications simulations (DDDAS), least squares, parameters update

## 1 Introduction

Dynamic data-driven applications simulations (DDDAS) are important for many practical applications. Consider an extreme example of a disaster scenario in which a major waste spill occurs in a subsurface near a clean water aquifer. Sensors can now be used to measure where the contaminant was spilled, where the contamination is, where the contaminant is going to go, and to monitor the environmental impact of the spill. One of the objectives of DDDAS is to incorporate the sensor data into the real-time computer simulations. A number of important issues are involved in DDDAS, and they are described

⋆ Research of the authors is partially supported by NSF grant ITR-0219627.

in [7]. In this paper, our goal is to devise DDDAS with a procedure for recovering initial data based upon an available set of sensor measurements and to use them for solution forward prediction. In particular, the procedure involves updating the initial data (the solution at previous time steps) as the sensors measurements are acquired.

As new data are obtained from sensors measurements, the initial data needs to be updated. This update reduces the computational errors associated with incorrect initial data and improves the predictions. In this paper, we consider linear subsurface flows involving convection and diffusion. Initial data is sought in a finite dimensional space. Using the first set of measurements, the approximation of the initial data is recovered. As new data are incorporated into the simulator, we update the initial data using an objective function. We note that the formulated problem is ill-posed. Two facts can be attributed to this ill-posedness. First, the data gathered from the sensor measurements always contain some defects that come from factors such as human errors and inherent factory errors of the sensors. Secondly, the number of sensors that can be installed are limited, and in general are much fewer than the finite dimensional space describing the initial data. For the latter, we can regularize the problem by using the prior information about the initial data. This prior information is the updated initial data. The penalization constants depend on time of update and can be associated with the relative difference between simulated and measured values.

A number of numerical examples are presented. The main objective of these examples is to show how crucial the initial data update is. In particular, with the correct choice of penalty terms, one can improve the prediction of the initial data. The improved predictions of the initial data provides us with more accurate solution at later times. In the paper, we only consider a few updates because, in dynamic data-driven applications simulations, our approach will be used only locally in time by updating the solution at the previous time steps rather than at time zero. The latter can provide significant computational savings.

As mentioned earlier, sensor measurements contain errors and uncertainties. In our numerical simulations, we take into account these uncertainties by sampling the sensor data from the known distribution. As a result, one obtains various realizations of the initial data. In our subsequent work [6], we will employ the least squares approach in developing Bayesian methods. To quantify uncertainties in the measurements and *a priori* knowledge about the initial data, the Markov Chain Monte Carlo method (MCMC) can be used. Because this method is expensive due to rejection of the proposals, we propose an approach that combines the least squares method with Bayesian approaches that will give high acceptance rates.

## 2 Initial Data Recovery and Update Based on Multiple Sets of Measurements

The model problem that we consider is a convection-diffusion problem

$$\frac{\partial C}{\partial t} + v \cdot \nabla C - \nabla \cdot (D \nabla C) = 0 \quad \text{in } \Omega, \tag{2.1}$$

where by Darcy's Law, we have $v = -k\nabla p$, with the pressure $p$ satisfies

$$-\nabla \cdot (k \nabla p) = 0 \tag{2.2}$$

with some prescribed boundary conditions and initial condition/data $C(\mathbf{x}, 0) = C^0(\mathbf{x})$. This initial boundary value problem models the tansport of a contaminant through the porous medium $\Omega$. There have been many researches devoted to studying this model [1,2]. In addition, many aspects of numerical simulation of this problem are investigated in [3,9] and references therein. Here the variable $C(\mathbf{x}, t)$ is designated for a contaminant concentration over the porous medium $\Omega$ and at time level $t$, $k$ is the permeability of the porous medium, and $D$ is the diffusion coefficient.

In standard practice, given the boundary and initial data, one solves (2.1) and (2.2) forward in time to obtain the concentration $C$ at a certain time level $t$. One of the problems in dynamic data-driven simulation is the estimation of the initial condition $C^0(\mathbf{x})$ given a set of spatially sparse concentration measurements at certain time levels. In other words, instead of solving for $C$, our task is to solve for $C^0(\mathbf{x})$ given sparse values of $C$ and with the assumption that the boundary data are known. We note that in this study, the permeability $k$ is randomly generated with a pre-assigned statistical variogram and exhibit a certain correlation structure [10,4].

Before presenting the procedure, we shall introduce several pertaining notations. Let $N_s$ be the number of sensors installed in various points in the porous medium and $\{\mathbf{x}_j\}_{j=1}^{N_s}$ denote such points. Let $N_t$ be the number of how many times the concentration is measured in time and $\{t_k\}_{k=1}^{N_t}$ denote such time levels. Furthermore, $\gamma_j(t_k)$ denotes the measured concentration at sensor located in $\mathbf{x}_j$ and at time $t_k$. We set

$$M(\gamma, t_k) = \{\gamma_j(t_k), j = 1, \cdots, N_s\}. \tag{2.3}$$

Now let $\Omega_c \subset \Omega$ be the predicted location at which the initial data exist inside the porous medium $\Omega$. Moreover, we denote by $W_c$ a finite dimensional space that lives on $\Omega_c$ and let $N_c$ be its dimension. The finite dimensional space $W_c$ is equipped with a set of linearly independent functions that we denote by $\{\tilde{C}_i^0(\mathbf{x})\}_{i=1}^{N_c}$. Obviously, any function that belongs to $W_c$ can be written as a linear combination of $\{\tilde{C}_i^0(\mathbf{x})\}_{i=1}^{N_c}$.

The idea explored in this paper is to seek an approximation of the initial data $C^0(\mathbf{x})$ in the finite dimensional space $W_c$. To see this, we denote that approximation by $\tilde{C}^0(\mathbf{x})$, which is represented as

$$\tilde{C}^0(\mathbf{x}) = \sum_{i=1}^{N_c} \alpha_i \tilde{C}_i^0(\mathbf{x}), \tag{2.4}$$

for some $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_{N_c})$. Furthermore, let $\tilde{C}_i(\mathbf{x}, t)$ be the solution of (2.1) using an initial condition $\tilde{C}_i^0(\mathbf{x})$. Then by superposition principle, the solution of (2.1) using $\tilde{C}^0(\mathbf{x})$ in (2.4) as an initial condition has the following form:

$$\tilde{C}(\mathbf{x},t) = \sum_{i=1}^{N_c} \alpha_i \tilde{C}_i(\mathbf{x},t). \tag{2.5}$$

Having written the approximation of the initial data as in (2.4), we transform the task of recovering the initial condition of (2.1) into a problem of finding the "best" $\alpha$ such that $\tilde{C}(\mathbf{x},t) \approx C(\mathbf{x},t)$. This is materialized through the formulation of an objective function that quantifies the difference between the measured concentration, $M(\gamma,t)$, and the simulated concentration, $\tilde{C}(\mathbf{x},t)$. As already mentioned in the introduction, in general the number of the sensors are less than the dimension of $W_c$ (i.e., $N_s < N_c$). Hence, an attempt to minimize an objective function that only contains the difference between measurements and simulations will lead to an ill-posed problem. To regularize the problem, we add a penalty term that contains the prior information related to the initial data, and consider the following objective function

$$F : \mathbb{R}^{N_c} \to \mathbb{R}, \tag{2.6}$$

such that

$$F(\alpha) = \sum_{j=1}^{N_s} \left( \sum_{i=1}^{N_c} \alpha_i \tilde{C}_i(\mathbf{x}_j,t) - \gamma_j(t) \right)^2 + \sum_{i=1}^{N_c} \kappa_i \left(\alpha_i - \beta_i\right)^2 . \tag{2.7}$$

Here $\kappa = (\kappa_1, \kappa_2, \cdots, \kappa_{N_c})$ is the penalty coefficients for an *a priori* vector $\beta = (\beta_1, \beta_2, \cdots, \beta_{N_c})$. This prior information will be updated during the simulation to achieve higher accuracy. Minimization of the target function (2.7) is done by setting

$$\frac{\partial F(\alpha)}{\partial \alpha_m} = 0, \qquad m = 1, \cdots, N_c, \tag{2.8}$$

which gives the linear system

$$A\alpha = R, \tag{2.9}$$

where for $m, n = 1, \cdots, N_c$,

$$A_{mn} = \sum_{j=1}^{N_s} \tilde{C}_m(\mathbf{x}_j,t)\tilde{C}_n(\mathbf{x}_j,t) + \delta_{mn}\kappa_m, \tag{2.10}$$

with $\delta_{mn} = 1$ if $m = n$ and zero otherwise, and

$$R_m = \sum_{j=1}^{N_s} \tilde{C}_m(\mathbf{x}_j,t)\gamma_j(t) + \kappa_m \beta_m. \tag{2.11}$$

In the following, we describe several ingredients pertaining to the implementation of the least squares described above. This approach relies on some prior knowledge inherent in the predicted initial data. This set of knowledge is referred to as *priors*. In the formulation above, the priors are represented by the variables $\Omega_c$, $N_c$, $\beta$, and $\kappa$. At the preliminary stage, one needs to determine possible spatial location of the initial data and its support in the porous medium $\Omega$. In the worst possible scenario, when we do not have prior information on the location of the initial data, we may set $\Omega_c = \Omega$. The next step is to choose the finite dimensional space $W_c$ in which we want to seek our approximate initial data. Mainly, we will need to decide what kind of linearly independent functions we will use along with the dimension, $N_c$. If there is a reasonable level of certainty on the location of initial data and its support, then a smaller number of $N_c$ may be used. On the other hand, a larger $N_c$ has to be used if one has a low certainty level of the location, which results in predicted initial condition whose support occupies a larger portion of the porous medium. Closely connected to the location of initial data are whether or not we have some prediction (or knowledge) on the initial data values at the specified points. These values are represented by $\beta$. Provided that we have this information, our aim is to obtain $\alpha$ from the above least squares system that are in the neighborhood of $\beta$. And finally, the closeness of the presumed $\beta$ to $\alpha$ is controlled by the vector $\kappa$.

As already pointed out, it is, in general, impossible to provide priors with certainty. Consequently, a procedure has to be proposed with the goal of increasing the priors' level of certainty in the real-time computer simulation. For this purpose, the least squares approach presented earlier can be suitably fitted into the framework of dynamic data-driven applications simulations. One of the objectives of DDDAS is to incorporate the measurement data into the real-time computer simulations. Several issues of DDDAS designed in the framework of contaminant prediction and tracking were discussed in [7,8].

The procedure that we propose in this paper is related to the sequence of updating sensor data described in [5]. We assume that multiple sets of measurement data are available. To explain the idea in a rigorous way, we concentrate on updating $\beta$ in $n > 1$ steps real-time simulations, namely at time $t_1, t_2, t_3, \cdots, t_n$. We note that in this case there are $n$ sets of measurement data at our disposal. At the very beginning of the simulation (at time level $t_1$), we come up with prior values of $\beta = \beta^0$. Running the simulation at time level $t_1$ gives us the predicted $\alpha^1$. Now the aim is to update $\beta$ to be used in the simulation at time level $t_2$, and for this we use the value of $\alpha^1$, which is already obtained. The updating sequence is then similarly proceeded up to time level $t_n$ (see Figure 1 for a schematic illustration of this procedure). Obviously, in conjunction with updating $\beta$, we may also update the value of $\kappa$ with the assumption that the updated $\beta$ are converging to the predicted $\alpha$.

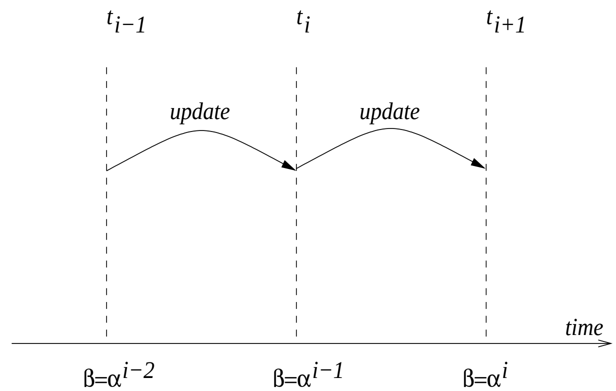


**Fig. 1.** Procedure for updating the prior information

In practical applications, the data acquired from measurements contain inherent errors/noise that add up the uncertainty in the whole simulation process. This noise affects the accuracy of the initial data recovery. One way to quantify this uncertainty is to sample the measurement data into several realizations and use them in the simulation to obtain the corresponding set of realizations of initial conditions. Another way is to use the Bayesian framework. This method will be presented elsewhere [6].

## 3 Numerical Examples

Below we present several synthetic examples to test the performance of the method described in the previous section. We use $\Omega = [0,1] \times [0,1]$. The boundary conditions in the subsurface flow for the pressure equation (2.2) are given pressure at the inlet and outlet edges (i.e., $x = 0$ and $x = 1$, respectively), and no flow at the bottom and top edges (i.e., $z = 0$ and $z = 1$, respectively). The permeability $k$ is generated with given correlation length

$l_x = 0.2$ and $l_z = 0.02$, with a spherical variogram using GSLIB algorithms [4].

For the convection-diffusion equation (2.1), we set the diffusion coefficient $D = 0.1$ over all domain. We assume zero concentration at the inlet, bottom, and top edges, and a zero diffusion, i.e., $(D\nabla C) \cdot \mathbf{n} = 0$, at the outlet edge, with $\mathbf{n}$ being the unit normal vector pointing outward on the outlet edge. In numerical investigations, we use the following form of true initial condition:

$$C^0(x,z) = \begin{cases} \sum_{i=1}^{n} c_i \phi_i(x,z), & \text{if } (x,z) \in Q \subset \Omega \\ 0, & \text{otherwise,} \end{cases} \tag{3.1}$$

where $\phi_i(x,z)$ are the standard bilinear functions of nodes $i = 1, \cdots, n$ in $Q$. One example of true initial condition that we want to recover uses $Q = (0.2, 0.4) \times (0.2, 0.4)$ with $n = 9$, and $c_1 = c_3 = c_7 = c_9 = 0.5625$, $c_2 = c_4 = c_6 = c_8 = 0.75$, and $c_5 = 1$. See left side of Figure 2 for schematic illustration of these nodes. Other example of true initial condition is defined similarly.

Both pressure and convection-diffusion equations are solved by the finite volume method on rectangular grids. For all cases we discretize the domain into $100 \times 100$ elements, i.e., 100 elements in each direction. A sparse portion of the numerical solution of (2.1) using the true initial condition is treated as the measured concentration obtained from the sensors. We use time step $\Delta t = 0.01$. The data from measurement $M(\gamma, t_k)$ are taken from multiple set of numerical simulations with the initial condition mentioned earlier. The measurement is assumed to be conducted at time level $t_1 = 0.1$, $t_2 = 0.2$, $t_3 = 0.3$, and $t_4 = 0.4$. A number of sensors are installed at various locations in the porous medium. Figure 3 shows the concentration profile at each of these time levels along with the sensor locations which are denoted by (x) indicator.

Regarding priors, we consider several scenarios. The first one is assuming that we know the support of the exact initial condition. Hence, we use $\Omega_c = Q = (0.2, 0.4) \times (0.2, 0.4)$ and $N_c = 9$. As prior information for $\beta$ we use $\beta_i^0 = 0$, $i = 1, \cdots, 9$. In this case, only five sensor data depicted in Figure 3 are used, i.e., all the four corner sensors and one at the very center.

The second case assumes we do not know exactly the support of the true initial condition. We use $\Omega_c = (0.05, 0.55) \times (0.2, 0.4)$ and $N_c = 27$, where the nodes now consist not only on the whole portion of the exact initial condition support, but also on the outside of it, namely, on the left and right boundaries of the exact initial condition support (see Figure 4). For this case all $\beta_i^0$ are nonzero.

In all the figures below, we show the sequence of updated initial condition corresponding to the sequence of time levels of measurement mentioned earlier. The first two figures below correspond to the prior scenario described in the last two paragraphs with the measurement data $M(\gamma, t_k)$ assumed to be deterministic, so that they are imposed into the least squares system as *hard* constraint. We use $\kappa_i = 2 \times 10^{-12}$ for all $i$ in these two cases. Figure 5 shows the updated initial data where the support is known exactly (i.e., $\Omega_c = Q$). We can see from the sequence in this figure that updating $\beta$ clearly improve the accuracy of the predicted initial condition.

Figure 6 shows the sequence of updated initial condition using larger support prior. We note that at each update $k$, there are negative values of $\alpha_i^k$ for nodes $i$ located outside the support of the exact initial data. As a part of the updating procedure, we truncate these negative values to zero, and the resulting vector is used as $\beta^{k+1}$. The reason for this is that the quantity of interest represents the concentration that cannot be negative. We observe in the simulation that as the time level progresses the magnitude of these negative values is decreasing, which results in the improvement of the predicted initial data. In particular, during the first update we see the recovered initial data has larger support than the true initial data. At later updates this support converges to the support of the true initial condition.

In addition to updating $\beta$, we can update $\kappa$. In particular, $\kappa$ is increased during the simulations to reflect the fact that the updated initial data is a better representation of the true initial data. In general, one can change $\kappa$ in various ways. In our examples, we update $\kappa_i$ by multiplying it with certain factor after each simulation update. Figure 7 shows the updated initial condition with both $\beta$ and $\kappa$ updated for the case of larger support. The prior for $\kappa$ is $\kappa_i^0 = 2 \times 10^{-12}$ for all $i$, and when updated it is multiplied by ten. The figure shows significant improvement on the predicted initial data.

Figure 8 is another example showing the advantage of updating $\kappa$ in conjunction with updating $\beta$. For this example we have used a different true initial data. As before, true initial data employed here follows the form in (3.1), where now we use $Q = (0.2, 0.4) \times (0.2, 0.4) \cup (0.1, 0.3) \times (0.6, 0.8)$, such that the true initial data exhibits two peaks and two distinct supports. The priors are chosen such that the predicted initial data has a spatial support that covers the supports of both peaks, namely we use $\Omega_c = (0.0, 0.5) \times (0.2, 0.8)$. Moreover, we have chosen the time instances for the simulation update to be $t_1 = 0.04$, $t_2 = 0.08$, $t_3 = 0.12$, and $t_4 = 0.16$. The numerical results presented in Figure 8 shows that updating both $\kappa$ and $\beta$ gives significant improvement.

Next, we present the case when we use $\Omega_c = \Omega$ as our prior. The true initial data we want to recover is the same as the one predicted in Figure 8. Obviously the finite dimensional space $W_c$ for this problem is much larger. Instead of updating four times in the numerical simulation, we have updated the prior seven times, i.e., at $t_1 = 0.04$, $t_2 = 0.08$, $t_3 = 0.12$, $t_4 = 0.16$, $t_1 = 0.20$, $t_2 = 0.24$, and $t_3 = 0.28$. Four of the updated initial data are depicted in Figure 10. Again, similar improvement on the updated initial data is evident.

As already mentioned in the introduction, another key purpose of DDDAS is to perform forward predictions based upon the approximated initial data. We use the predicted initial data shown in Figure 8 to illustrate this forward prediction. The comparison is shown in Figure 9. The top plot of this figure shows the true concentration at $t = 0.2$. In the bottom left plot of this figure we use initial xdata obtained after first update (see upper left plot of Figure 8) to predict the concentration at $t = 0.2$. We see a significant difference both in the form as well as the magnitude between true and the predicted concentrations. In the bottom right figure we use the initial data obtained at the fourth update (see lower right plot of Figure 8). This plot clearly shows that the predicted concentration is more accurate if we use the fourth update. This example gives an illustration of the advantage and importance of updating prior information for improving predictions in later times and its relevance in dynamic data-driven applications simulations.

Finally, we consider examples when the measurements are sampled. Sampling the measurements reflects the fact that the data acquired from measurement contain noise. Thus it is not reliable to impose it as a hard constraint. To impose it as a *soft* constraint, we sample the measurement data into a set of realizations and obtain the corresponding realizations of initial data using the least squares and updating procedure. Here, we sample the measurement data from the Gaussian distribution whose mean is the measured concentration corresponding to the true solution and variance is approximately five percent of this value. Figures 11 and 12 show two realizations of the updated initial data with the two peaks presented earlier. In each realization we see although the simulations tend to under-predict the true initial data, there is relative improvement as time level progresses.

## References


1. J. Bear, *Dynamics of Fluids in Porous Media*, Elsevier, 1972.
2. J. Bear, *Modeling transport phenomena in porous media*, in Environmental studies (Minneapolis, MN, 1992), vol. 79 of IMA Vol. Math. Appl., Springer, New York, 1996, pp. 27–63.
3. G. Chavent and J. Jaffré, *Mathematical Models and Finite Elements for Reservoir Simulation*, no. 17 in Studies in Mathematics and its Applications, North-Holland, Amsterdam, 1986.
4. C. V. Deutsch and A. G. Journel, *GSLIB: Geostatistical software library and user's guide, 2nd edition*, Oxford University Press, New York, 1998.
5. C. Douglas, C. Shannon, Y. Efendiev, R. Ewing, V. Ginting, R. Lazarov, M. Cole, G. Jones, C. Johnson, and J. Simpson, *A note on data-driven contaminant simulation*, Lecture Notes in Computer Science, Springer-Verlag, 3038 (2004), pp. 701–708.
6. C. C. Douglas, Y. Efendiev, R. Ewing, V. Ginting, and R. Lazarov, *Bayesian approaches for initial data recovery in dynamic data-driven simulations*, in preparation.
7. C. C. Douglas, Y. Efendiev, R. Ewing, R. Lazarov, M. R. Cole, C. R. Johnson, and G. Jones, *Virtual telemetry middleware for dddas*, Computational Sciences - ICCS 2003, P. M. A. Sllot, D. Abramson, J. J. Dongarra, A. Y. Zomaya, and Yu. E. Gorbachev (eds.), 4 (2003), pp. 279–288.
8. C. C. Douglas, C. Shannon, Y. Efendiev, R. Ewing, V. Ginting, R. Lazarov, M. R. Cole, G. Jones, C. R. Johnson, and J. Simpson, *Using a virtual telemetry methodology for dynamic data driven application simulations*, in Dynamic Data Driven Applications Systems, F. Darema (ed.), Kluwer, Amsterdam, 2004.
9. R. E. Ewing, ed., *The mathematics of reservoir simulation*, vol. 1 of Frontiers in Applied Mathematics, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1983.

10. H. WACKERNAGLE, *Multivariate geostatistics: an introduction with applications*, Springer, New York, 1998.

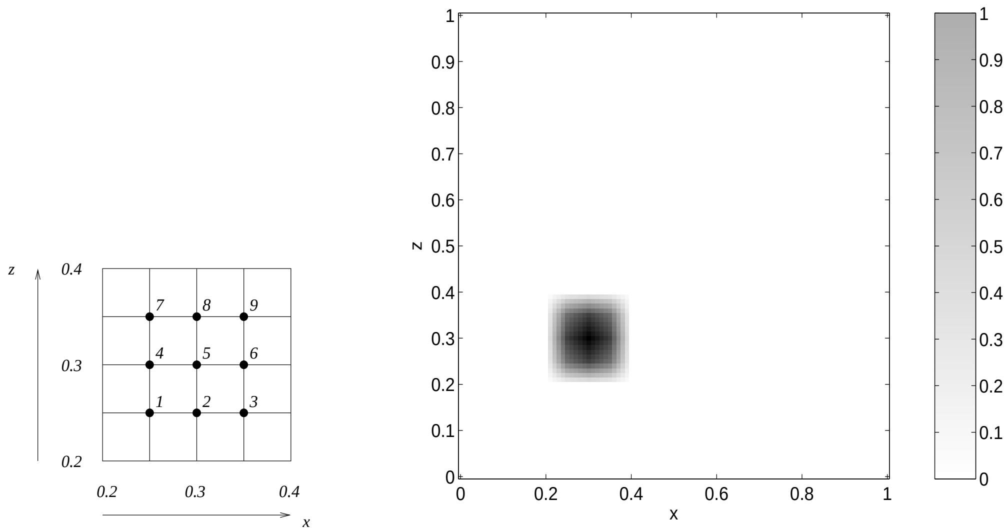


**Fig. 2.** (left) True initial condition is formed as a linear combination of nodal basis functions with known nodal values, (right) the initial condition profile.

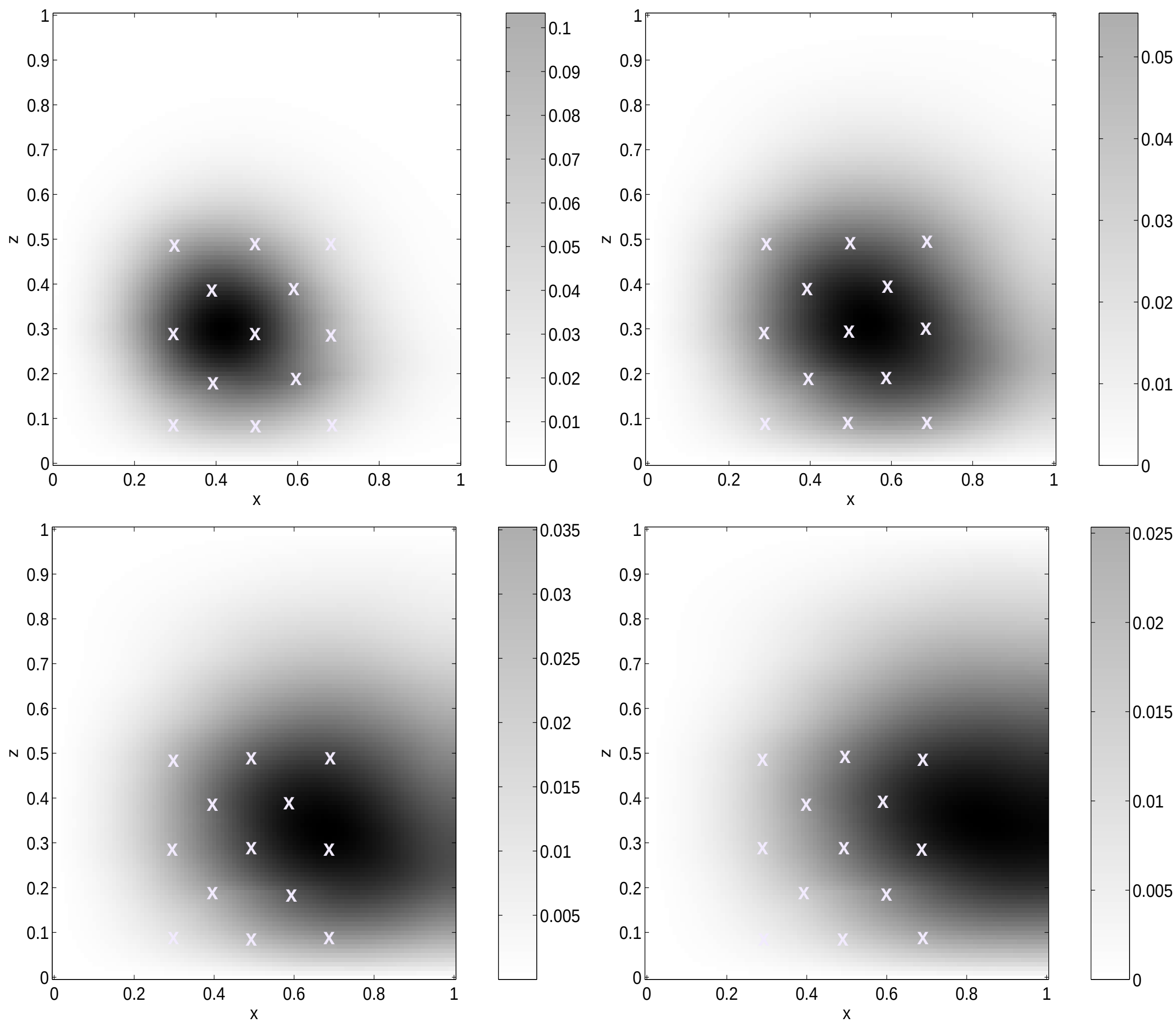

**Fig. 3.** Concentration at several time levels: $t = 0.1$ (top left), $t = 0.2$ (top right), $t = 0.3$ (bottom left), $t = 0.4$ (bottom right). The (x) indicates the sensor location.

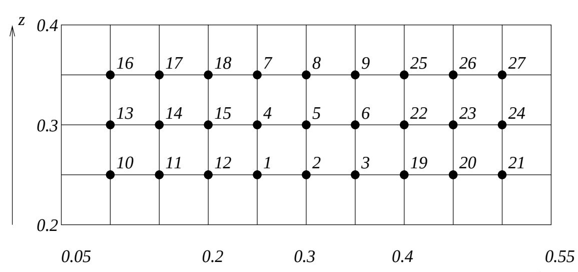


**Fig. 4.** Prior configuration when the support is not known

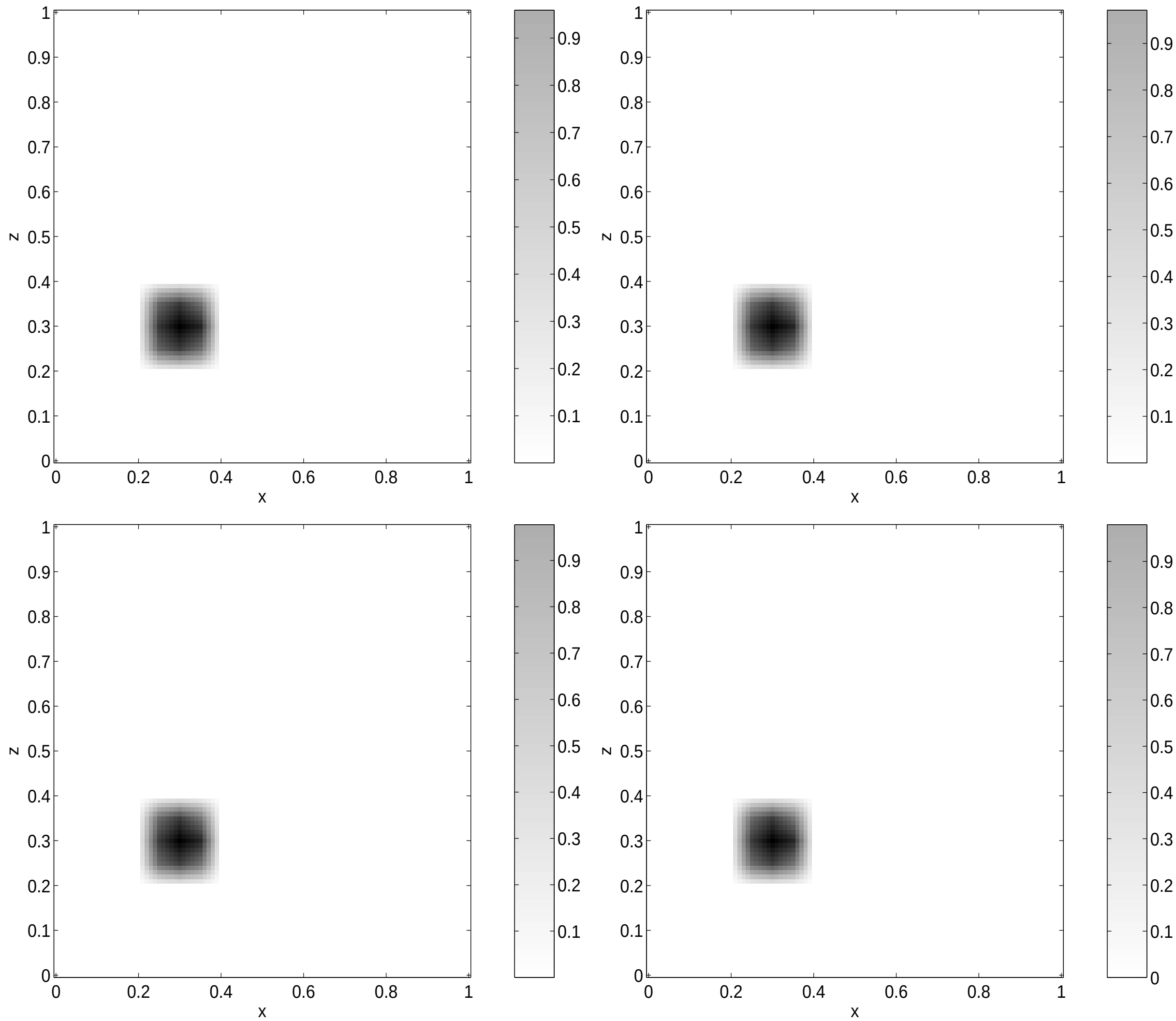


**Fig. 5.** Updated initial data: $t = 0.1$ (top left), $t = 0.2$ (top right), $t = 0.3$ (bottom left), $t = 0.4$ (bottom right). The prior assumes a known support.

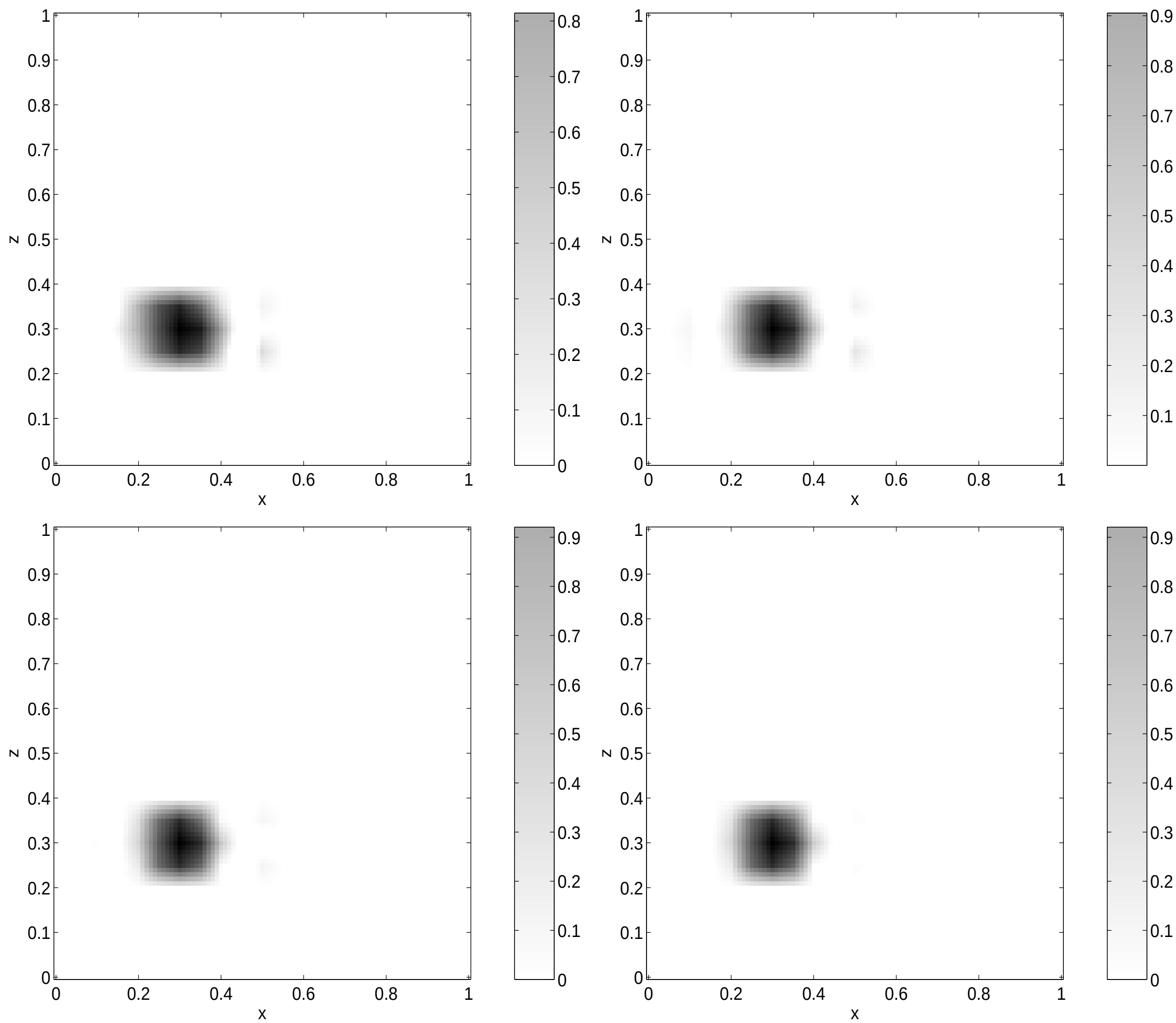


**Fig. 6.** Updated initial data: $t = 0.1$ (top left), $t = 0.2$ (top right), $t = 0.3$ (bottom left), $t = 0.4$ (bottom right). The prior for $\beta$ assumes a larger support.

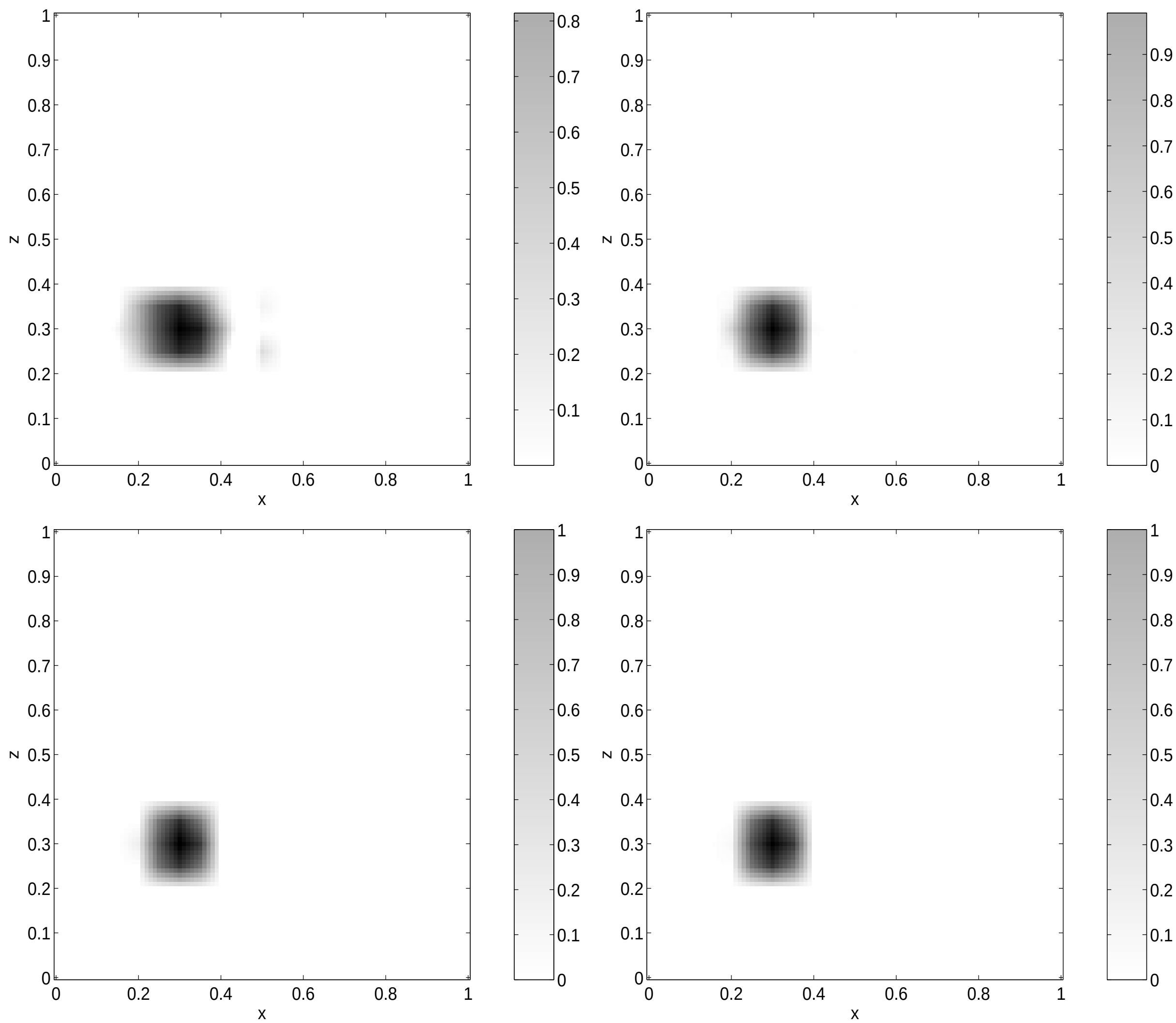


**Fig. 7.** Updated initial data: $t = 0.1$ (top left), $t = 0.2$ (top right), $t = 0.3$ (bottom left), $t = 0.4$ (bottom right). The prior for $\beta$ assumes a larger support. Both $\beta$ and $\kappa$ are updated.

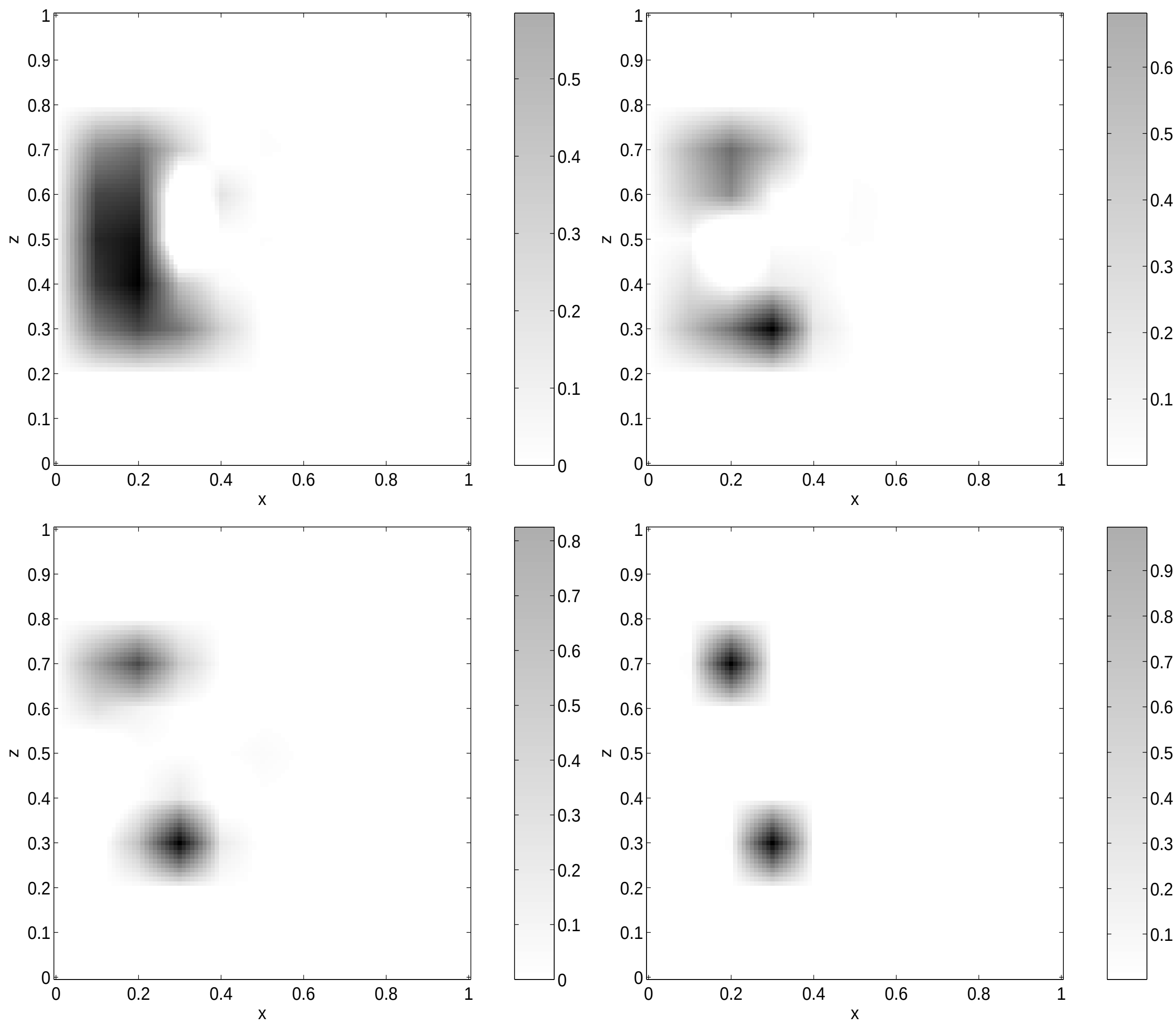


**Fig. 8.** Updated initial data for two-peak initial data: $t = 0.04$ (top left), $t = 0.08$ (top right), $t = 0.12$ (bottom left), $t = 0.16$ (bottom right). The prior for $\beta$ assumes a larger support that covers the support of both peaks. Both $\beta$ and $\kappa$ are updated.

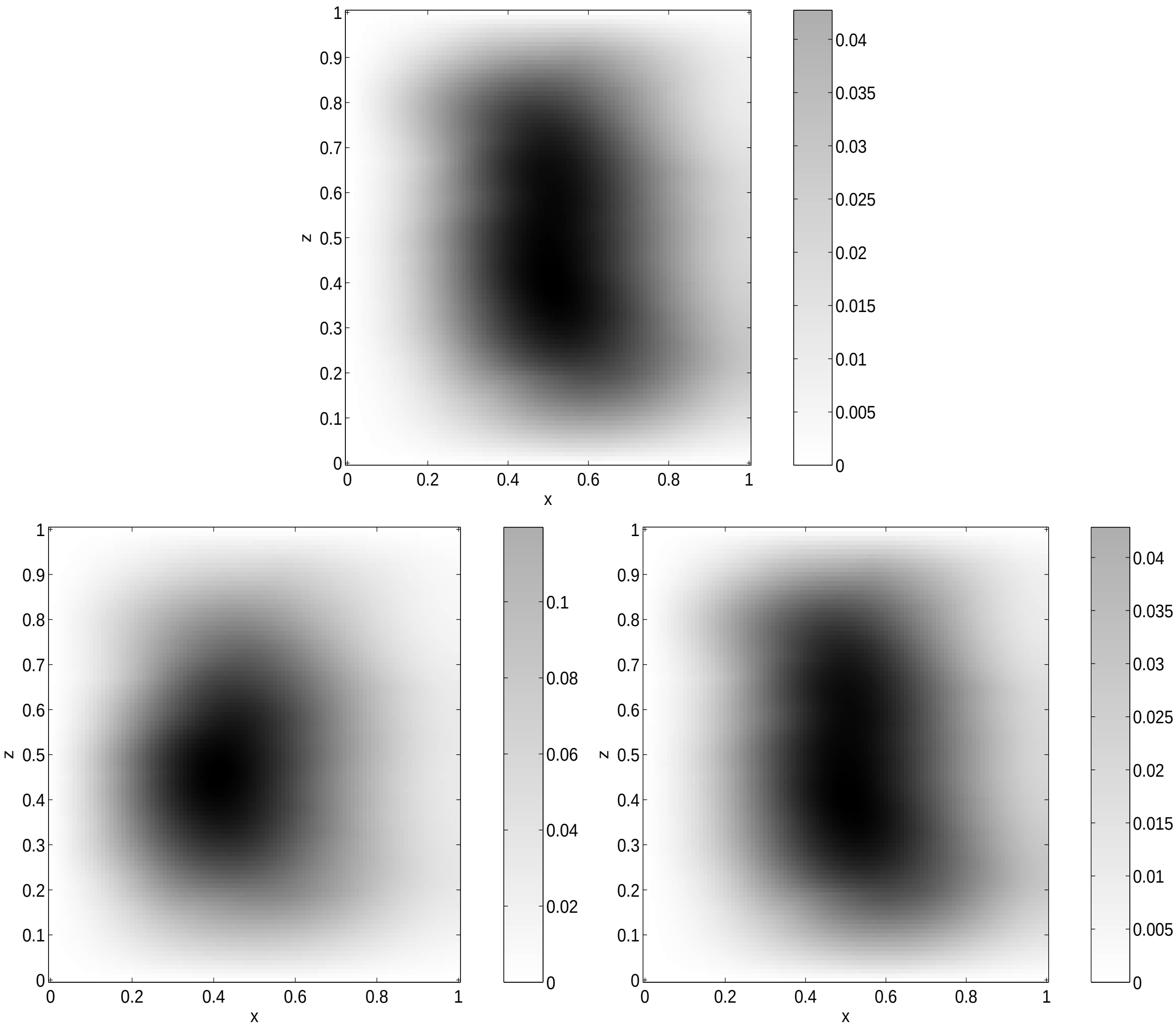


**Fig. 9.** Comparison of predicted concentration at $t = 0.2$ for the case presented in Figure 8: (top) true concentration, (bottom left) concentration using first update, (bottom right) concentration using fourth update.

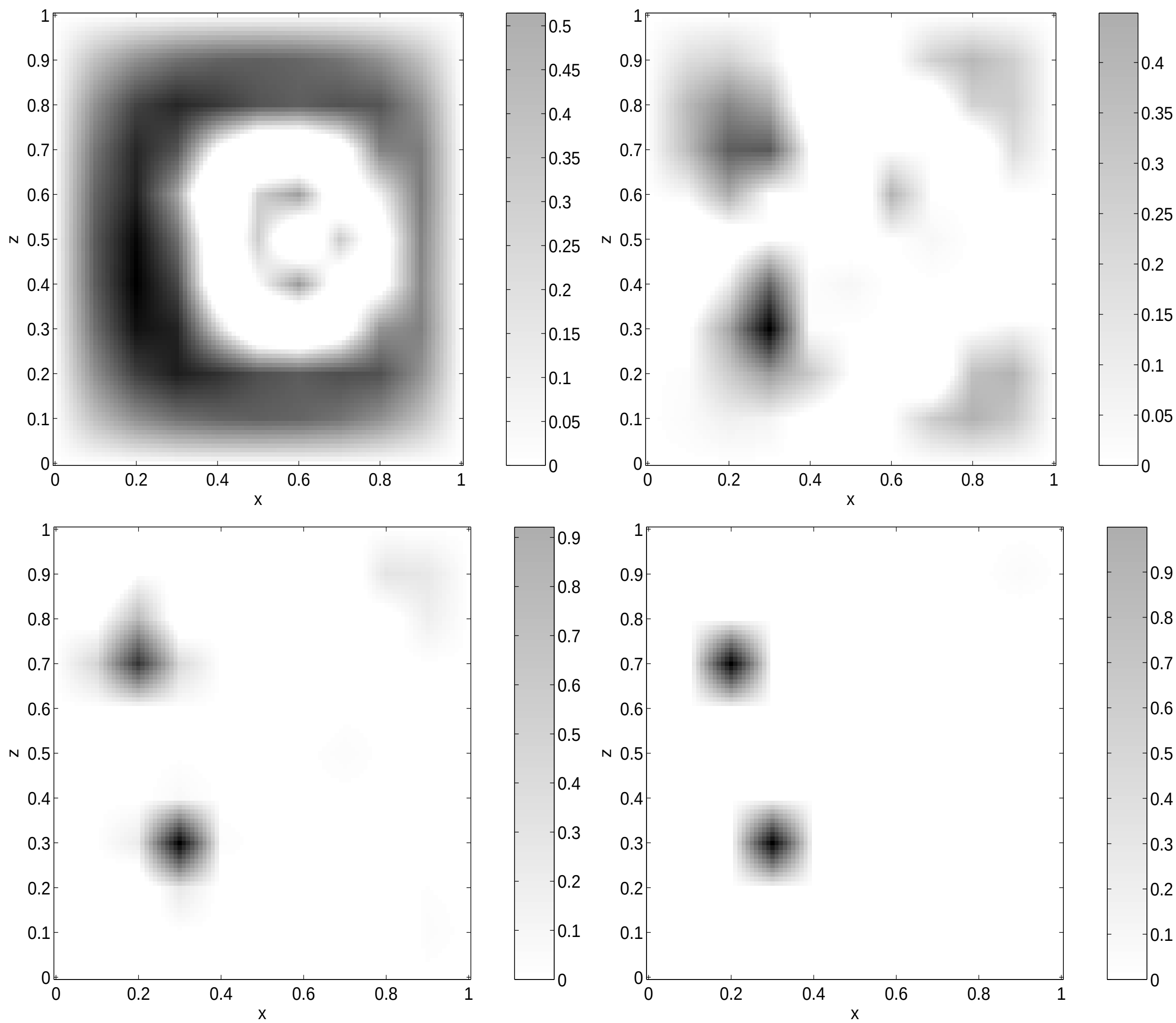


**Fig. 10.** Updated initial data for two-peak initial data: $t = 0.04$ (top left), $t = 0.12$ (top right), $t = 0.20$ (bottom left), $t = 0.28$ (bottom right). The prior for $\beta$ assumes a support equal to $\Omega$ ($\Omega_c = \Omega$). Both $\beta$ and $\kappa$ are updated.

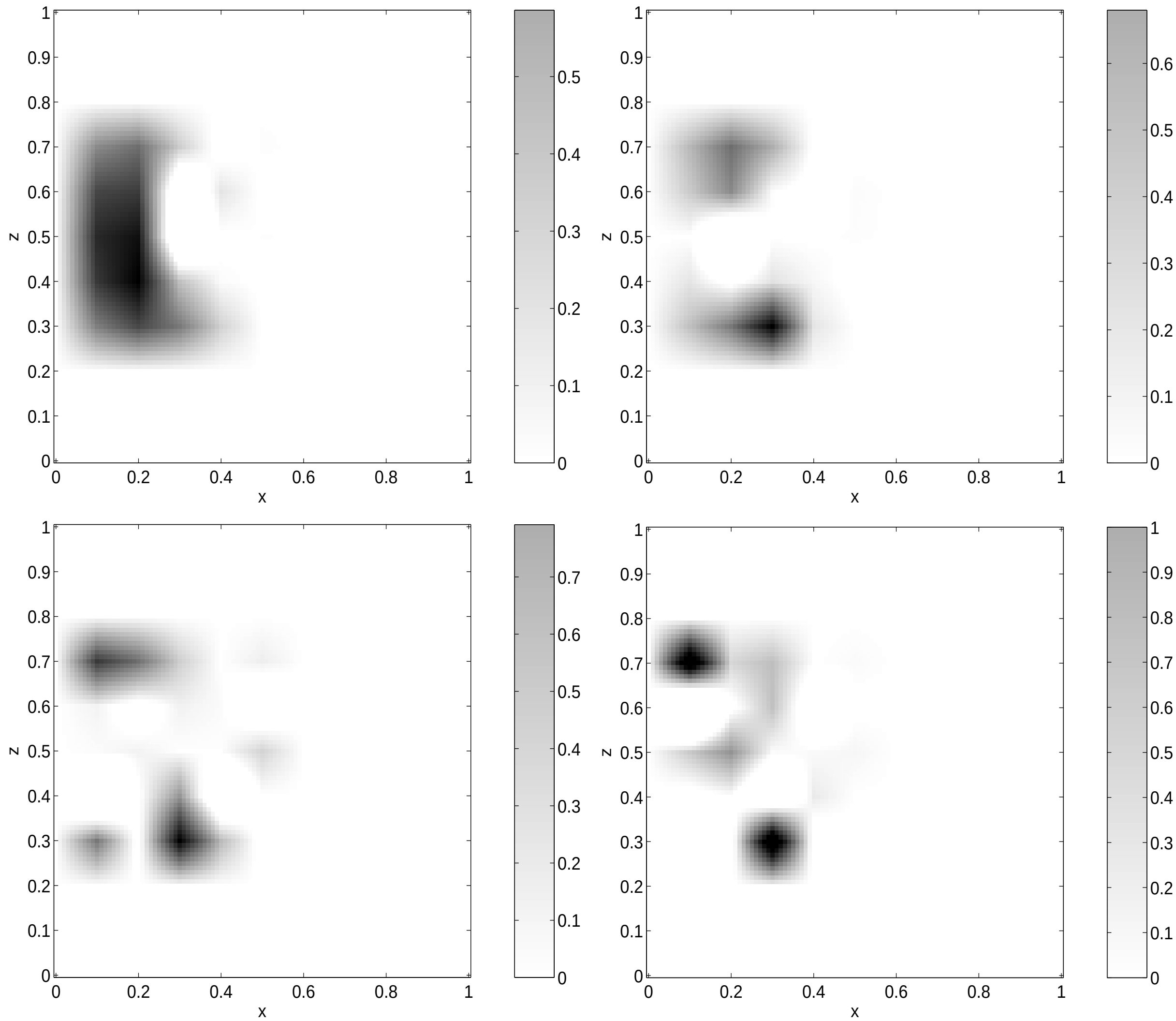


**Fig. 11.** First realization of updated initial data: $t = 0.1$ (top left), $t = 0.2$ (top right), $t = 0.3$ (bottom left), $t = 0.4$ (bottom right).

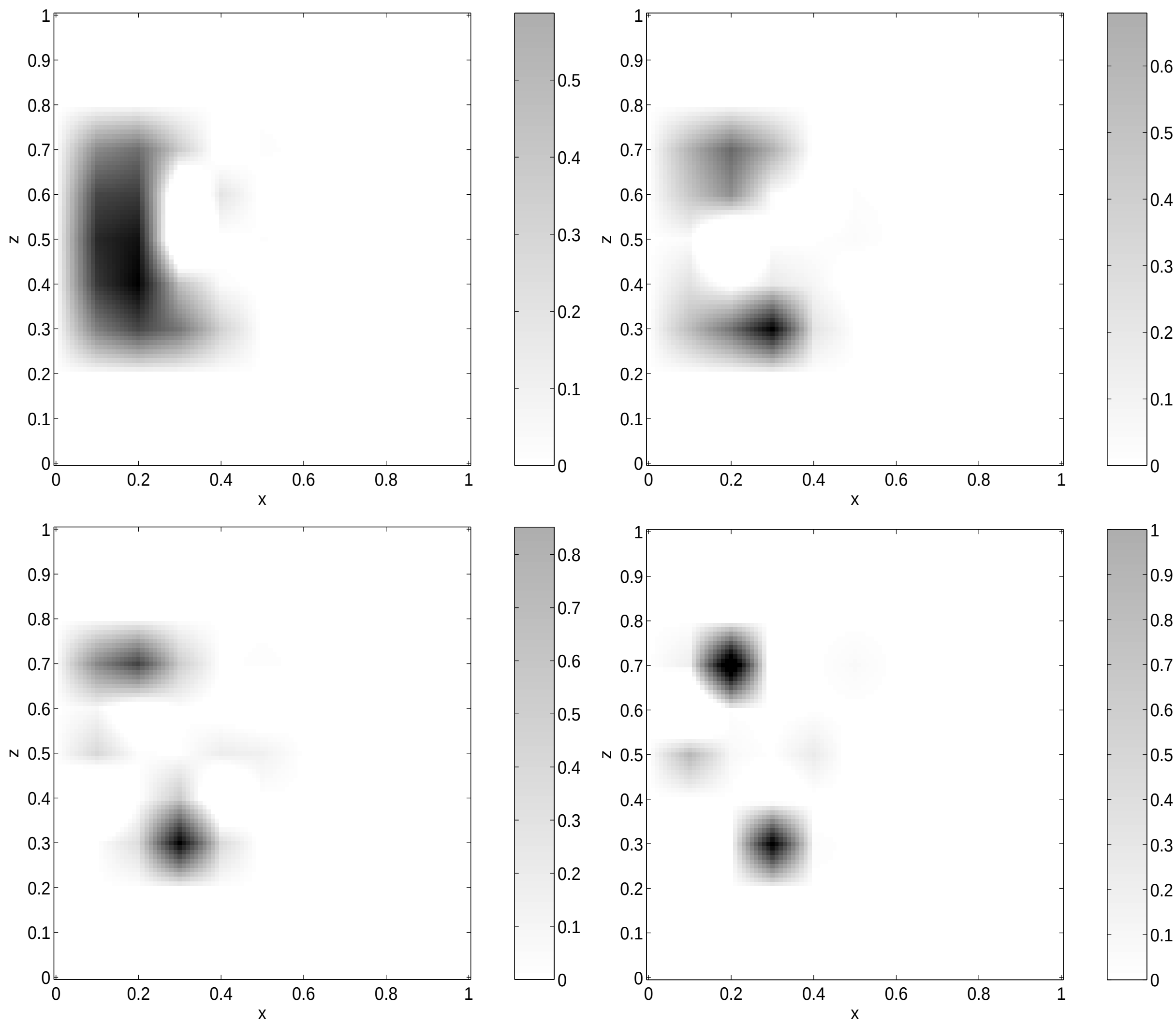


**Fig. 12.** Second realization of updated initial data: $t = 0.1$ (top left), $t = 0.2$ (top right), $t = 0.3$ (bottom left), $t = 0.4$ (bottom right).