



# Space–time adaptive approach to variational data assimilation using wavelets



Innocent Souopgui<sup>a</sup>, Scott A. Wieland<sup>b</sup>, M. Yousuff Hussaini<sup>a</sup>,  
Oleg V. Vasilyev<sup>b,\*</sup>

<sup>a</sup> Department of Mathematics, Florida State University, Tallahassee, FL 32310, USA

<sup>b</sup> Department of Mechanical Engineering, University of Colorado Boulder, 427 UCB, Boulder, CO 80309, USA

## ARTICLE INFO

### Article history:

Received 12 November 2014

Received in revised form 18 August 2015

Accepted 16 November 2015

Available online 2 December 2015

### Keywords:

Data assimilation

Space–time discretization

Wavelet

Wavelet collocation

Adaptive mesh

Numerical method

Multilevel method

Multigrid method

## ABSTRACT

This paper focuses on one of the main challenges of 4-dimensional variational data assimilation, namely the requirement to have a forward solution available when solving the adjoint problem. The issue is addressed by considering the time in the same fashion as the space variables, reformulating the mathematical model in the entire space–time domain, and solving the problem on a near optimal computational mesh that automatically adapts to spatio-temporal structures of the solution. The compressed form of the solution eliminates the need to save or recompute data for every time slice as it is typically done in traditional time marching approaches to 4-dimensional variational data assimilation. The reduction of the required computational degrees of freedom is achieved using the compression properties of multi-dimensional second generation wavelets. The simultaneous space–time discretization of both the forward and the adjoint models makes it possible to solve both models either concurrently or sequentially. In addition, the grid adaptation reduces the amount of saved data to the strict minimum for a given a priori controlled accuracy of the solution. The proposed approach is demonstrated for the advection diffusion problem in two space–time dimensions.

© 2015 Elsevier Inc. All rights reserved.

## 1. Introduction

The ability to accurately predict the evolution of physical systems is of crucial importance in many fields of science and engineering. Due to the complexity of most physical systems, direct approaches resolving all the physical phenomena are computationally unfeasible, necessitating the use of reduced physics models. These models often have many empirical parameters that need to be fine tuned. In addition, even if these parameters are known exactly or high fidelity approaches are used, both boundary and initial conditions are rarely available, not to mention the issues associated with aleatoric and epistemic uncertainties. Forecast and prediction algorithms should account for all the available information on the studied system. This information appear in several forms: numerical simulations of the system dynamics on a wide range of time and space scales; observations that provide measurements of the actual system state; statistical information on the errors inherent in the numerical models and observations; and qualitative information given by the knowledge of experts in the fields. Taken separately, none of these sources of information can be used to uniquely reconstruct the state of the physical system. Variational data assimilation is a mathematical technique that consistently combines all the available information in

\* Corresponding author.

E-mail addresses: [yousuff@fsu.edu](mailto:yousuff@fsu.edu) (M. Yousuff Hussaini), [Oleg.Vasilyev@Colorado.EDU](mailto:Oleg.Vasilyev@Colorado.EDU) (O.V. Vasilyev).

an optimality system to reconstruct the inputs of the model. Applications of data assimilation include, but are not limited to, meteorological forecasting, estimation and monitoring of climate change, estimation and monitoring of pollution, etc. More generally, data assimilation can be used whenever a system is described by a model with unknown parameters and supplemented by observations of the system state. In the variational data assimilation, the problem of determining model inputs (physical parameters and/or boundary/initial conditions) is formulated as a least squares problem and solved using optimization algorithms. The objective function (also referred to as the objective function or the cost function in the context of data assimilation) is expressed as the weighted norm of the difference between the solution of the *forward (direct)* problem and the observations as well as the term measuring deviation of the model's initial state and its prior estimate. Due to the nonlinearity of many problems of interest, variational data assimilation makes use of non-linear optimization algorithms (e.g., non-linear conjugate gradient methods [1,2], Newton and quasi-Newton methods [3,4]). Those methods require the gradient of the cost function. Adjoint methods provide a mathematically elegant and efficient algorithmic framework for the computation of the gradient [5]. In practice, the numerical model is a discretization of a continuous model combined with a time marching method to compute the system state at each time step. The solution of the adjoint problem is a function of the solution of the forward problem. The adjoint problem being solved using backward time marching requires the full solution (in space and time) of the forward problem to be available. That is impracticable for large-scale problems and represents one of the big challenges of adjoint methods. Two extreme strategies to address this challenge consist in saving all the intermediate results of the direct problem or recomputing them when they are required. The former requires the storage of very large data sets (at least the solution of the direct problem at all time steps), while the latter requires redundant computations. A compromise consists in saving a reduced set of intermediate results with subsequent re-computation of the missing data [6]. However, the need of memory storage and redundant computations persists.

An elegant attempt to reduce the computational cost of 4D-Var was proposed by Courtier et al. [7], where an incremental strategy (4D-Inc) to solve the 4D-Var problem at progressively increasing resolutions is introduced. The 4D-Inc could be viewed as a first attempt to solve 4D-Var problem using multi-resolution approximation. In this paper, a novel adaptive multi-resolution framework is proposed where the challenges associated with 4D-Var are addressed by formulating the 4-dimensional variational data assimilation (4D-Var) problem in the entire space–time domain and solving it on a computational mesh that adapts simultaneously in space and time to the intermittent structures of the solution, thus drastically reducing the number of required degrees of freedom. The space–time grid adaptation is achieved by using the Space–Time Adaptive Wavelet Collocation Method [8], a novel approach based on wavelet compression that was recently demonstrated to be capable of solving a problem on a near optimal space–time adaptive grid with a priori user controlled global accuracy of the solution. It should be mentioned that the use of space–time adaptation allows mesh compression that takes advantage of spatial as well as temporal intermittency, thus, resulting in considerable computational saving, both in terms of simulation time and storage space [9], compared to time marching methods. It should be mentioned that a number of attempts to use wavelet analysis to improve the efficiency of variational data assimilation has been made. However the use of wavelet analysis was limited either for analysis and extraction of spatial structures [10], regularization of ill-posed problem of adjoint parameter estimation [11], better modeling/representation/compression of error covariance matrices [12–15]. However, this work is the first attempt to develop formal wavelet-based adaptive multi-resolution methodology for 4D-Var. The distinguishing feature of the proposed approach is that it is based on space–time adaptive methods to solve both forward and adjoint problems in  $d + 1$  dimensional space, where  $d$  is the dimensionality of the physical space, and the additional dimension is the time.

The paper is organized as follows. Section 2 gives a brief introduction to the conventional formulation of the 4D-Var. The space–time formulation of 4D-Var is presented in Section 3. The space–time adaptive wavelet collocation method for solving time dependent partial differential equations is introduced in Section 4. The mathematical formulation of 4D-Var model test problem is given in Section 5. Finally, the results of application of the proposed space–time adaptive 4D-Var approach to model test problem are discussed in Section 6. The main results are summarized, and future research directions are outlined in Section 7.

## 2. Conventional formulation of 4D-Var

Let us consider a dynamic system that is described by the  $n$ -dimensional “state variable”  $\mathbf{u} = (u_1, u_2, \dots, u_n)^T$ , where the superscript T indicates transpose and  $\mathbf{u}$  is a function of the physical space and the time:  $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$ ;  $u_i = u_i(\mathbf{x}, t)$  for  $1 \leq i \leq n$ , where  $\mathbf{x}$  is the physical space variable and  $t$  is the time variable. The domain of the physical space variable is defined as  $\Omega_s \subset \mathbb{R}^d$ ,  $1 \leq d \leq 3$  so that  $\mathbf{x} = (x_1, x_2, \dots, x_d)^T$ . The domain of the time variable is defined as the interval  $[0, T]$  with  $T > 0$ . For simplicity it is assumed that  $\mathbf{u}(\mathbf{x}, t) \in \mathbb{R}^n$  for all  $(\mathbf{x}, t) \in \Omega_s \times [0, T]$ . The main components of 4D-Var are the numerical model (also referred to as direct model) and the observations.

### 2.1. Direct model

The state of the system evolves according to the following generic differential system:

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t}(\mathbf{x}, t) = \mathbf{F}(\mathbf{u}(\mathbf{x}, t)), & (\mathbf{x}, t) \in \Omega_s \times [0, T], \\ \mathbf{u}(\mathbf{x}, 0) = \mathbf{c}(\mathbf{x}), \end{cases} \quad (1)$$

with  $\mathbf{u}(\mathbf{x}, 0) = \mathbf{c}(\mathbf{x})$ , being the initial condition, which hereafter is called the control variable.  $\mathbf{F}: \mathbb{R}^n \rightarrow \mathbb{R}^n$  is a vector valued function such that  $\mathbf{F} = (F_1, F_2, \dots, F_n)^T$  with  $F_i = F_i(\mathbf{u})$  for  $1 \leq i \leq n$ . Each component  $F_i$  of  $\mathbf{F}$  is assumed to be “sufficiently smooth” in  $\mathbf{u}$  so that the solution of (1) exists and is unique for a given initial condition. All the other parameters (physical parameters, empirical parameters, boundary conditions, model errors, etc.) of the dynamic system are included in the definition of the function  $\mathbf{F}$ . The boundary conditions are discussed in detail in Section 3. The model (1) hereafter is referred to as the direct model.

2.2. Observations

Let  $\mathbf{h}: \mathbb{R}^n \rightarrow \mathbb{R}^m$  and  $\mathbf{e}^o: \Omega_s \times [0, T] \rightarrow \mathbb{R}^m$ . Let

$$\begin{aligned} \mathbf{y}^o: \Omega_s \times [0, T] &\rightarrow \mathbb{R}^m \\ (\mathbf{x}, t) &\mapsto \mathbf{y}^o(\mathbf{x}, t) = \mathbf{h}(\mathbf{u}(\mathbf{x}, t)) + \mathbf{e}^o(\mathbf{x}, t) \end{aligned} \tag{2}$$

be the observations of the state variable, where  $\mathbf{h} = (h_1, h_2, \dots, h_m)^T$  with  $h_i = h_i(\mathbf{u})$  for  $1 \leq i \leq m$  is the observation operator and  $\mathbf{e}^o = (\varepsilon_1^o, \varepsilon_2^o, \dots, \varepsilon_m^o)^T$  with  $\varepsilon_i^o = \varepsilon_i^o(\mathbf{x}, t)$  for  $1 \leq i \leq m$  is an additive noise  $\mathbf{e}^o$ . Note that the observation operator does not need to be continuous. In practice, the observations can be obtained at a finite number of space–time locations. However, in order to have consistent continuous formulation, it is assumed that  $\mathbf{h}$  and its Jacobian are defined in  $\Omega_s \times [0, T]$  space–time domain, which can be achieved by simple interpolation of the observation operator between observation points. In other words, in most general terms,  $\mathbf{h}$  is a nonlinear space–time operator.

2.3. Continuous formulation of 4D-Var

Assuming that all other parameters of the model function  $\mathbf{F}$  are known, given the observation  $\mathbf{y}^o$ , the problem is to estimate the control  $\mathbf{c}$  when the state variable  $\mathbf{u}$  is constrained by the dynamics of the direct model (1). It is important to point out that the estimation of other parameters of the model function  $\mathbf{F}$  is mathematically similar to the estimation of the initial condition. Given a background estimation  $\mathbf{c}^b$  of the control, the 4D-Var defines the control variable as the minimum of the following cost function:

$$J(\mathbf{c}) = \frac{1}{2} \int_0^T \|\mathbf{h}(\mathbf{u}(\mathbf{x}, t)) - \mathbf{y}^o(\mathbf{x}, t)\|_{\mathcal{O}}^2 dt + \frac{1}{2} \|\mathbf{c} - \mathbf{c}^b\|_{\mathcal{B}}^2 \tag{3}$$

where the norms  $\|\cdot\|_{\mathcal{O}}$  and  $\|\cdot\|_{\mathcal{B}}$  come from appropriate inner products and can take into account the statistical information in the form of error covariance matrices, where the second term measures the difference between the model’s initial state and its prior estimate. In addition, the second term of the cost function (3) serves as a regularization term in the sense of Tikhonov [16], which remedies the ill-posedness of the inverse problem.

A necessary condition for  $\mathbf{c}^a$  to be the minimum of the cost function (3) is that

$$\nabla J(\mathbf{c}^a) = 0, \tag{4}$$

where  $\nabla J$  is the gradient of the cost function with respect to the control variable. The condition (4) becomes a sufficient one if the cost function  $J$  is strictly convex and the control space is also convex. The control space is usually chosen to be convex and the regularization enforces the convexity of the cost function. In practice,  $\mathbf{c}^a$  is estimated by a descent-type algorithm. Given an initial approximation  $\mathbf{c}_0$ , such an algorithm defines  $\mathbf{c}^a$  as the limit of the sequence

$$\mathbf{c}_{k+1} = \mathbf{c}_k + \lambda_k \mathbf{d}_k, \tag{5}$$

where  $\mathbf{d}_k$  is the direction of descent and  $\lambda_k$  is the step size that realizes the minimum of the cost function along  $\mathbf{d}_k$ . The descent direction  $\mathbf{d}_k$  is deduced from the gradient  $\nabla J$ . In the general case, the gradient  $\nabla J$  of the cost function with respect to the control cannot be expressed directly. In 4D-Var,  $\nabla J$  is computed by introducing the *adjoint variable*  $\mathbf{p}$  (of the same dimension as the state variable) defined as the solution of the adjoint model:

$$\begin{cases} \frac{\partial \mathbf{p}}{\partial t}(\mathbf{x}, t) + \left[ \frac{\partial \mathbf{F}}{\partial \mathbf{u}} \right]^* \cdot \mathbf{p}(\mathbf{x}, t) = \left[ \frac{\partial \mathbf{h}}{\partial \mathbf{u}} \right]^* \cdot [\mathbf{h}(\mathbf{u}(\mathbf{x}, t)) - \mathbf{y}^o(\mathbf{x}, t)], & (\mathbf{x}, t) \in \Omega_s \times [0, T], \\ \mathbf{p}(\mathbf{x}, T) = 0, \end{cases} \tag{6}$$

where  $\left[ \frac{\partial \mathbf{F}}{\partial \mathbf{u}} \right]$  and  $\left[ \frac{\partial \mathbf{h}}{\partial \mathbf{u}} \right]$  are the Jacobian matrix of  $\mathbf{F}$  and  $\mathbf{h}$ , respectively, with respect to the state variable  $\mathbf{u}$ . The asterisk  $(\cdot)^*$  denotes the adjoint operator associated with the inner product used to define the norm in the cost function. The gradient of the cost function is then given by:

$$\nabla J(\mathbf{c}) = -\mathbf{p}(\cdot, 0) + \mathcal{B}(\mathbf{u} - \mathbf{u}^b), \tag{7}$$

where  $\mathcal{B}(\mathbf{u})$  is the transformation operator associated with the norm  $\|\cdot\|_{\mathcal{B}}$ . An example of such an operator will be given in the discrete formulation.

The derivation of the adjoint model (6) is based on the calculus of variations. Its formulation can be found in [17] or in the traditional books on data assimilation such as Ref. [5].

## 2.4. Discretization of the data assimilation model

For a time-dependent problem like the one introduced above, typical approaches of solution are based on “semidiscretization”, meaning that one discretizes first in space and then in time, or vice versa. In data assimilation, it is common to first discretize in space. The equations (1) and (6) are then solved using a forward and a backward integration in time, respectively. For this reason, the equations (1) and (6) are sometimes called the forward model and the backward model, respectively. Equation (1) can be discretized using a variety of schemes. Assuming that a uniform time step  $\Delta_t$  is used, the resulting discrete version of the direct model can be represented as

$$\begin{cases} \mathbf{u}_{k+1} = \mathbf{M}(\mathbf{u}_k), & 1 \leq k < n_t, \\ \mathbf{u}_0 = \mathbf{c}, \end{cases} \quad (8)$$

where  $\mathbf{u}_k \in \mathbb{R}^{n_s}$  is the discrete system state at the  $k$ th time step;  $n_t$  is the total number of time steps ( $T = n_t \Delta_t$ ) and  $\mathbf{M}: \mathbb{R}^{n_s} \rightarrow \mathbb{R}^{n_s}$  is a vector-valued function such that  $\mathbf{M} = (M_1, M_2, \dots, M_{n_s})^T$  with  $M_i = M_i(\mathbf{u})$  for  $1 \leq i \leq n_s$ .  $n_s$  is the size of the discrete system state at each time step. It should be noted that the same generic symbol  $\mathbf{u}$  is used for the state variable in the continuous and in the discrete formulations, which should not be a source of confusion. Similarly, the generic symbols  $\mathbf{c}$ ,  $\mathbf{y}^o$  and  $\mathbf{h}$  are used for the control variable, the observation function and for the observation operator, respectively, both in the continuous and in the discrete form. With such a discretization, the cost function is rewritten as

$$J(\mathbf{c}) = \frac{1}{2} \sum_{k=0}^{n_t} \|\mathbf{h}(\mathbf{u}_k) - \mathbf{y}_k^o\|_{\mathbf{R}_k}^2 + \frac{1}{2} \|\mathbf{c} - \mathbf{c}^b\|_{\mathbf{B}^{-1}}^2, \quad (9)$$

where  $\mathbf{y}_k^o$  is the discrete observation at the  $k$ th time step,  $\mathbf{R}_k$  is the error covariance matrix associated with  $\mathbf{y}_k^o$  and  $\mathbf{B}$  is the error covariance matrix associated with  $\mathbf{c}^b$ .  $\mathbf{R}_k^{-1}$  and  $\mathbf{B}^{-1}$  can be replaced by other symmetric, positive, definite matrices that are more appropriate. The size of the discrete state variable at each time step is equal to the number of components of the continuous state variable multiplied by the number of grid points in the physical space.

For the discrete adjoint model, there are two possible strategies: the discretization of the continuous adjoint model or the derivation of an adjoint model associated with the discrete version of the direct model. The later is preferred for two main reasons: it gives the exact gradient of the discrete cost function and the availability of automatic differentiation softwares makes it more attractive as it does not require any effort to build or rebuild the adjoint model.

## 3. Space–time formulation of the 4D-Var problem

The main difference in the space–time formulation of the 4D-Var problem compared to conventional formulations lies in the discretization. In the space–time formulation, the time variable is considered as an extra space variable. An extended space–time domain  $\Omega$  is defined by combining the physical and the time domains:  $\Omega = \Omega_s \times [0, T]$ , where  $\Omega_s$  and  $[0, T]$  are the physical space and the time domains, respectively. See Section 2 for their definition. The “semidiscretization” is not considered. The space–time domain is considered as a whole and the discretization is carried out on it. In mathematical terms, nothing changes for the continuous formulation of the data assimilation problem.

### 3.1. Direct model

The generic symbol  $\mathbf{x}$  is used for the space–time variable. This should not be a source of confusion as it will always be clear from the context.  $\mathbf{x} = (x_1, x_2, \dots, x_{d+1})^T$  where  $d$  is the dimensionality of the physical space domain. The first  $d$  components  $(x_1, x_2, \dots, x_d)^T$  of  $\mathbf{x}$  are the components of the physical space domain and the component  $d + 1$  is the time variable ( $x_{d+1} = t$ ). The state variable becomes  $\mathbf{u} = \mathbf{u}(\mathbf{x})$ . The direct model can be rewritten as follows:

$$\begin{cases} \bar{\mathbf{F}}(\mathbf{u}(\mathbf{x})) = 0 & \mathbf{x} \in \Omega, \\ \mathbf{u}(\mathbf{x}|_{x_{d+1}=0}) = \mathbf{c}, \end{cases} \quad (10)$$

where  $\bar{\mathbf{F}}(\mathbf{u}(\mathbf{x})) = \frac{\partial \mathbf{u}}{\partial t}(\mathbf{x}, t) - \mathbf{F}(\mathbf{u}(\mathbf{x}, t))$ ,  $\mathbf{F}$  being the operator defined in Eq. (1) and  $\{\mathbf{x} \in \Omega | x_{d+1} = 0\}$  is the boundary of the space–time domain with  $x_{d+1} = 0$ . Thus the second equation of Eq. (10) is the equivalent to the initial condition in the conventional formulation.

### 3.2. Cost function

With the space–time formulation, the observations  $\mathbf{y}^o$  have to be written in terms of the space–time variable

$$\begin{aligned} \mathbf{y}^o : \Omega &\rightarrow \mathbb{R}^m \\ \mathbf{x} &\mapsto \mathbf{y}^o(\mathbf{x}) = \mathbf{h}(\mathbf{u}(\mathbf{x})) + \boldsymbol{\varepsilon}^o(\mathbf{x}). \end{aligned} \quad (11)$$

With this formulation of the observations, the cost function can be written as:

$$J(\mathbf{c}) = \frac{1}{2} \|\mathbf{h}(\mathbf{u}) - \mathbf{y}^o\|_{\mathcal{O}}^2 + \frac{1}{2} \|\mathbf{c} - \mathbf{c}^b\|_{\mathcal{B}}^2 \tag{12}$$

where

$$\|\mathbf{h}(\mathbf{u}) - \mathbf{y}^o\|_{\mathcal{O}} = \int_0^T \|\mathbf{h}(\mathbf{u}(\mathbf{x})) - \mathbf{y}^o(\mathbf{x})\|_{\mathcal{O}}^2 dx_{d+1} \tag{13}$$

The adjoint model becomes:

$$\begin{cases} \bar{\mathbf{F}}^*(\mathbf{p}(\mathbf{x})) = \left[ \frac{\partial \mathbf{h}}{\partial \mathbf{u}} \right]^* \cdot [\mathbf{h}(\mathbf{u}(\mathbf{x})) - \mathbf{y}^o(\mathbf{x})], & \mathbf{x} \in \Omega, \\ \mathbf{p}(\mathbf{x}|_{x_{d+1}=T}) = \mathbf{0}, \end{cases} \tag{14}$$

where

$$\bar{\mathbf{F}}^*(\mathbf{p}(\mathbf{x})) = \frac{\partial \mathbf{p}}{\partial x_{d+1}}(\mathbf{x}) + \left[ \frac{\partial \mathbf{F}}{\partial \mathbf{u}}(\mathbf{u}(\mathbf{x})) \right]^* \cdot \mathbf{p}(\mathbf{x}) \tag{15}$$

It is important to point out that the derivation of the adjoint model from the formulation of Section 2 or from the space-time formulation, using the calculus of variations, leads to the same result. In addition, the only change in equations (11), (12) and (14) compared to (2), (3) and (6), respectively, is that the physical space variable and the time variable are grouped together. From a mathematical point of view, the formulations are equivalent. What distinguishes the space-time 4D-Var formulation from its conventional counterpart is an option to solve the problem on a space-time adaptive computational mesh, thus, fully utilizing the spatio-temporal intermittency of the solution.

### 3.3. Discretization of simultaneous space-time data assimilation

#### 3.3.1. Direct model

Equation (10) can be discretized in the space-time domain using a variety of schemes. Section 4 presents the approach used in this paper. The resulting discrete version of the direct model can be represented as

$$\mathbf{u} = \bar{\mathbf{M}}(\mathbf{c}), \tag{16}$$

where the generic symbol  $\mathbf{u}$  represents the vector of the discrete system state in the space-time domain and  $\bar{\mathbf{M}}: \mathbb{R}^{n_s} \rightarrow \mathbb{R}^n$  is a vector-valued function such that  $\bar{\mathbf{M}} = (\bar{M}_1, \bar{M}_2, \dots, \bar{M}_n)^T$  with  $\bar{M}_i = \bar{M}_i(\mathbf{c})$  for  $1 \leq i \leq n$ .  $n_s$  is the size of the discrete control variable  $\mathbf{c}$  and  $n$  is the size of the discrete system state variable  $\mathbf{c}$  in the full space-time domain. As for the system state variable, the same generic symbol  $n$  is used for the size of the state variable, which should not be a source of confusion. The technical details on the discretization and the solution of the direct model are given in Section 4.

#### 3.3.2. Cost function

With the discretization of the direct model above, the cost function becomes:

$$J(\mathbf{c}) = \frac{1}{2} \|\mathbf{h}(\mathbf{u}) - \mathbf{y}^o\|_{\mathbf{R}^{-1}}^2 + \frac{1}{2} \|\mathbf{c} - \mathbf{c}^b\|_{\mathbf{B}^{-1}}^2, \tag{17}$$

where  $\mathbf{R}$  is the matrix, denoting the error covariance associated with  $\mathbf{y}^o$ . The size of the discrete state variable is equal to the number of components of the continuous state variable multiplied by the number of grid points in the discrete space-time domain. Similar considerations are valid for the observation variable as long as the assumption of continuity is valid.

#### 3.3.3. Adjoint model

As in the conventional approach, there are two possible strategies for the discrete adjoint model: the discretization of the continuous adjoint model or the derivation of the adjoint model associated with the discrete version of the direct model. With the space-time formulation of the 4D-Var problem, the second strategy becomes less attractive, since iterative approaches are used to solve the equation (16) associated with the direct model.

The discretization of the continuous adjoint model offers three main advantages in the space-time formulation. The first advantage is that, it is possible to solve the direct and the adjoint models both concurrently and sequentially. When problems are solved sequentially, first the direct problem is solved on a space-time adaptive computational mesh, similar to the traditional time marching approach. The adjoint problem is solved on the same or subsampled mesh to speed-up the calculations. When problems are solved concurrently, both forward (16) and adjoint (14) problems are solved iteratively as one system of equations with incremental increase of resolution, similar in spirit to the 4D-Var approach of Courtier et al. [7]. The second advantage is that the computational mesh is adapted simultaneously both in space and time. Note that the mesh compression can be used as a measure of computational efficiency of the space-time adaptive 4D-Var approach, since the cost of the algorithm is linearly proportional to the total number of grid points used on the adaptive space-time mesh. The third advantage is that the mesh can locally be adapted in any subset of the space-time domain without extra

work for the adjoint model. Furthermore, the accuracy of the adjoint problem can be imposed independently of the direct model, resulting in an optimal space–time mesh for the adjoint model, which ultimately improves the efficiency of the data assimilation.

**4. Space–time adaptive wavelet collocation method**

The Space–Time Adaptive Wavelet Collocation Method (STAWCM) is a variable order numerical method for time dependent partial differential equations, which solves the problem on a computational mesh that adapts simultaneously in space and time to the intermittent structures of the solution. In this section, the main features of the method are briefly outlined. For details the reader is referred to [8,18].

The space–time grid adaptation is based on wavelet compression and exploits the one-to-one correspondence between grid-points and wavelet coefficients. More specifically, the grid adaptation process for the solution of a partial differential equation consists of the following steps:

- (i) Knowing the solution  $u(\mathbf{x})$  on the current space–time computational grid, say,  $\mathcal{G}^k$ , the solution is represented in terms of wavelet basis functions as

$$u(\mathbf{x}) = \sum_{\mathbf{l} \in \mathcal{L}^0} c_{\mathbf{l}}^0 \phi_{\mathbf{l}}^0(\mathbf{x}) + \sum_{j=0}^{+\infty} \sum_{\mu=1}^{2^n-1} \sum_{\mathbf{k} \in \mathcal{K}^{\mu,j}} d_{\mathbf{k}}^{\mu,j} \psi_{\mathbf{k}}^{\mu,j}(\mathbf{x}), \tag{18}$$

where bold subscripts denote  $n$ -dimensional indexes ( $n = d + 1$ ,  $d$  is the spatial dimension of the problem), while  $\mathcal{L}^0$  and  $\mathcal{K}^{\mu,j}$  are  $n$ -dimensional index sets associated with scaling functions at the zero level of resolution ( $\phi_{\mathbf{l}}^0$ ) and wavelets of family  $\mu$  and level  $j$  ( $\psi_{\mathbf{k}}^{\mu,j}$ ), respectively. Each level of resolution  $j$  consists of a family of wavelets  $\psi_{\mathbf{k}}^{\mu,j}$  having the same scale but located at different grid positions.

- (ii) A mask is created for the collocation points corresponding to wavelets with coefficients, whose moduli are above the prescribed threshold, resulting in the compressed representation of the solution given by

$$u_{\geq}(\mathbf{x}) = \sum_{\mathbf{l} \in \mathcal{L}^0} c_{\mathbf{l}}^0 \phi_{\mathbf{l}}^0(\mathbf{x}) + \sum_{j=0}^{+\infty} \sum_{\mu=1}^{2^n-1} \sum_{\substack{\mathbf{k} \in \mathcal{K}^{\mu,j} \\ |d_{\mathbf{k}}^{\mu,j}| \geq \epsilon \|u\|}} d_{\mathbf{k}}^{\mu,j} \psi_{\mathbf{k}}^{\mu,j}(\mathbf{x}), \tag{19}$$

where  $\epsilon > 0$  stands for the non-dimensional (relative) threshold value and  $\|u\|$  being the (absolute) dimensional scale.

- (iii) The mask is extended by adding grid points corresponding to adjacent wavelets, whose coefficients can potentially become significant at the next grid iteration  $\mathcal{G}^{k+1}$ .
- (iv) A recursive reconstruction check procedure on the mask is carried out to ensure that all ancestry grid points necessary to perform the forward wavelet transform on the modified mesh  $\mathcal{G}^{k+1}$  are present [18].

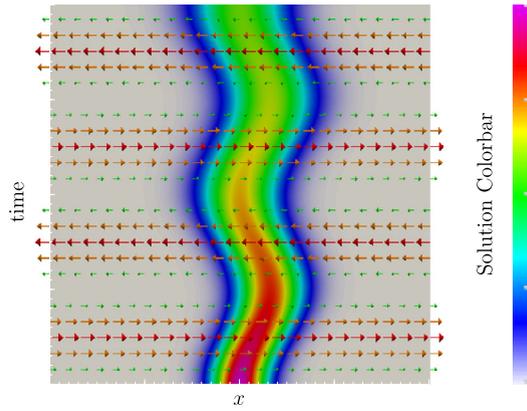
The numerical method is formally derived by solving the governing partial differential equations at collocation points corresponding to the extended mask  $\mathcal{G}^{k+1}$ . The procedure is repeated until the solution and the grid converge. With such an algorithm, the grid of collocation points is continuously refined in the space–time computational domain to resolve the local structures that appear in the solution. Note that by omitting wavelets with coefficients below a threshold parameter  $\epsilon$ , the error of approximation is automatically controlled, since it can be shown that the relative difference between unfiltered,  $u(\mathbf{x})$ , and compressed,  $u_{\geq}(\mathbf{x})$ , signals is bounded by the threshold level  $\epsilon$  [19]. Thus, in addition to obtaining the solution on a near optimal space–time adaptive grid, the STAWCM has another important feature: active control of the global accuracy of the solution. The smaller  $\epsilon$  is chosen, the smaller the error of the solution is. Also note that for practical reasons, the highest level of resolution in Eqs. (18) and (19) is set to  $j_{\max}$ . The STAWCM is used to solve the simultaneous space–time adaptive 4D-Var problem discussed in Section 3.

**5. Data assimilation model problem**

As a proof of concept, the space–time adaptive approach is applied to a model data assimilation problem in two space–time dimensions. The space–time domain  $\Omega$  is defined as  $\Omega = \Omega_s \times [0, T]$ , where  $\Omega_s = [-1, 1]$  and  $[0, T]$  are the physical space and the time domains, respectively.

**Model problem.** A linear advection–diffusion problem in one spatial dimension ( $d = 1$ ) is considered:

$$\left\{ \begin{array}{l} \frac{\partial u}{\partial x_{d+1}} + v_d \frac{\partial u}{\partial x_d} - v \frac{\partial^2 u}{\partial x_d^2} = 0, \\ u|_{x_{d+1}=0} = c(x_d), \\ u|_{x_d=-1} = u|_{x_d=+1} = 0, \end{array} \right. \tag{20}$$



**Fig. 1.** Advection velocity superimposed on the exact solution for the model problem (20). The velocity vectors are colored by the magnitude. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

where  $u$  is the scalar system state,  $c(x_d)$  is a function that defines the initial condition, and  $v_d(\mathbf{x})$  is the advection velocity. For clarity of the notation the subscripts  $d$  and  $d + 1$  are used to denote spatial and time coordinates, respectively. Equation (20) defines the direct model.

The associated adjoint model is given by:

$$\begin{cases} \frac{\partial p}{\partial x_{d+1}} + \frac{\partial(v_d p)}{\partial x_d} + v \frac{\partial^2 p}{\partial x_d^2} = \left[ \frac{\partial \mathbf{h}}{\partial \mathbf{u}} \right]^* (h(u) - y^o), & \mathbf{x} \in \Omega, \\ p|_{x_{d+1}=T} = 0, \\ p|_{x_d=-1} = p|_{x_d=+1} = 0, \end{cases} \tag{21}$$

where  $p$  is the adjoint variable. Both the direct and the adjoint model are solved using the Space–Time Adaptive Wavelet Collocation Method (see Section 4 for details).

**Observations.** The numerical results presented in the subsequent sections are based on twin experiments: the direct model is solved with a known control variable  $c^t$ . Observations are sampled on a coarse, fixed, and uniformly spaced grid associated with the locations of wavelets at level  $j = j_{\text{obs}}$  and below. The solution induces an observation operator defined by the projection onto the subspace generated by the wavelet atoms from levels 0 to  $j_{\text{obs}}$ . For simplicity, the observation points are forced to always be present in the computational mesh. In order to have a consistent representation of the observation operator, it needs to be defined at all points present on the adaptive space–time mesh. This can be achieved either by interpolation both observation operator and its Jacobian to the computational mesh. A more practical approach is to calculate the forcing function (the right-hand-side of Eq. (21)) at the observation points and interpolate it to other computational meshes. In the context of STAWCM, this interpolation is simply achieved by performing the inverse wavelet transform of the forcing function from the level  $j_{\text{obs}}$ . In practical applications and irregular space–time observation points a non-wavelet-based interpolation of the forcing function can be utilized.

**Optimization.** The cost function is minimized using the M1QN3 algorithm of the MODULOPT library [20]. M1QN3 is a quasi-Newton algorithm.

5.1. Numerical implementation

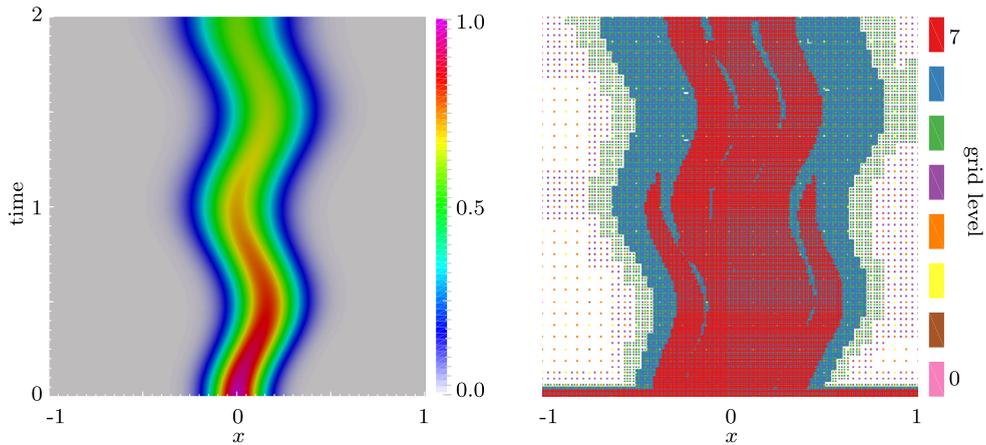
The space–time domain is  $\Omega = [-1, 1] \times [0, 2]$  and the advection velocity is given by

$$v(x_d, x_{d+1}) = 0.5 \sin(2\pi x_d + 1) \tag{22}$$

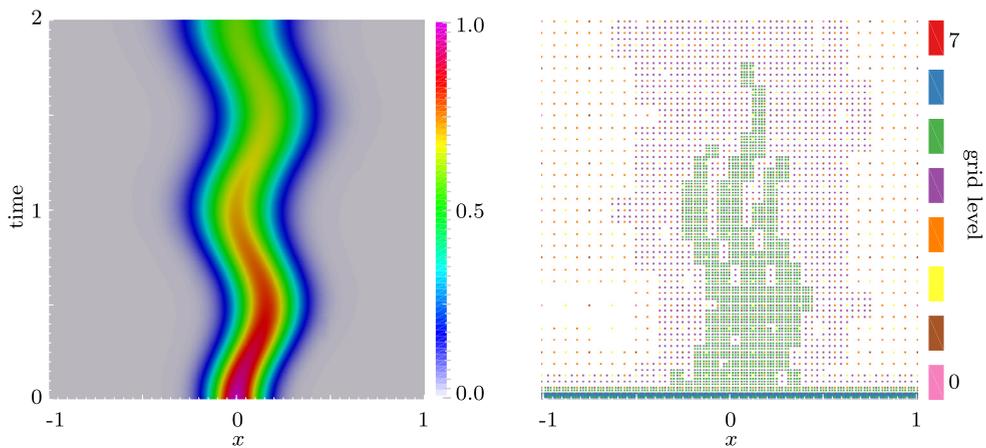
as shown in Fig. 1. The advection velocity defines a constant wind along the physical dimension (associated with the coordinate variable  $x_d$ ), which varies along the time dimension (associated with the coordinate variable  $x_{d+1}$ ). The exact control variable is given by:

$$c^t(x_d) = \exp\left(-\frac{x_d^2}{2\sigma^2}\right) \text{ with } \sigma = 0.1. \tag{23}$$

The control variable is the boundary condition at  $x_{d+1} = 0$  in the space–time formulation, or the initial condition in the conventional formulation. The Dirichlet boundary conditions on the physical space boundaries are assumed:



**Fig. 2.** True solution (left) and corresponding grid (right) with the **exact** control variable,  $j_{\max} = 7$ , and  $\epsilon = 10^{-6}$ . (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)



**Fig. 3.** Analysis: the solution (left) and the corresponding grid (right) with the **analyzed** control variable,  $j_{\max} = 6$ , and  $\epsilon = 10^{-3}$ . (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

$$u(-1, t) = u(1, x_{d+1}) = 0, \quad \forall x_{d+1} \in [0, 1] \quad (24)$$

To avoid interpolation of the control vector, the grid adaptation on the boundary  $x_{d+1} = 0$  is deactivated. As a result, all the points at the maximum level of resolution are always present on the boundary  $x_{d+1} = 0$ .

To sample observations, the direct model is simulated on the adaptive computational mesh corresponding to effective grid resolution of  $513 \times 513$  and  $j_{\max} = 7$ . For the direct problem the adaptation parameter  $\epsilon$  is set to  $10^{-6}$ . These settings ensure that both the solution and the grid converge. The associated solution of the direct problem is considered as the true solution. Observations are sampled on  $15 \times 16$  regularly spaced grid points corresponding to  $j_{\text{obs}} = 2$  level of resolution. There is no observation at time  $t = 0$  nor at the physical space boundaries. For the data assimilation problem, different settings are used:  $j_{\max} = 6$  and  $\epsilon$  ranging from  $5 \times 10^{-1}$  to  $10^{-5}$ . The lowering of the resolution  $j_{\max}$  and increasing  $\epsilon$  introduces model error that makes the problem more realistic. The variation of  $\epsilon$  is used to study the effect of the adaptation in data assimilation. With  $j_{\max} = 6$  and  $4 \times 4$  points at the 0-th level of resolution, the space-time data assimilation problem is solved at effective grid resolution of  $257 \times 257$ , which is coarser than the resolution of the true solution. Furthermore, taking a larger  $\epsilon$  has the effect of eliminating more features in the solution, reducing the number of degrees of freedom and increasing the model error.

## 5.2. Nonlinear extension

The data assimilation model problem can be easily extended for a nonlinear case by simply adding an additional nonlinear advection term to the equation. The notation and domain are the same as in the linear case, and the linear advection velocity is also taken to be identical.

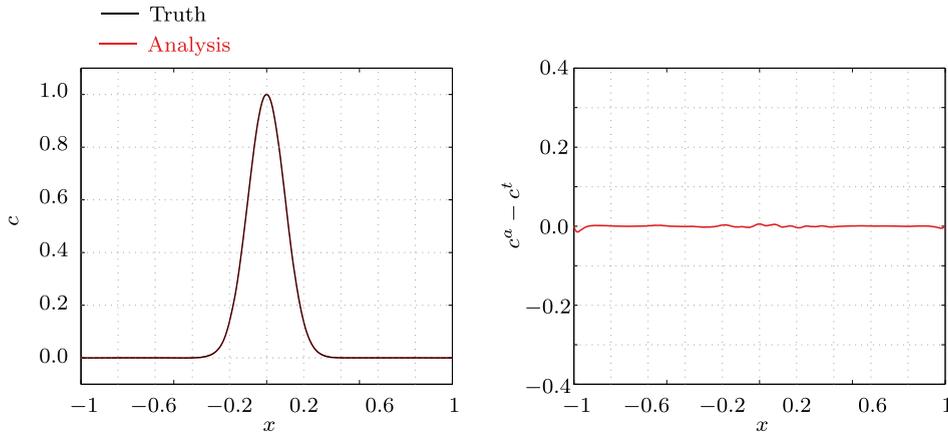


Fig. 4. Analyzed control variable: true solution and analysis (left); analysis error (right).

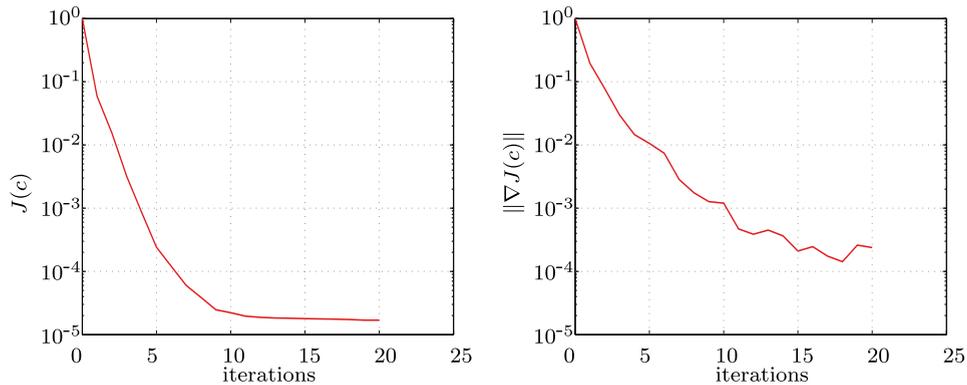


Fig. 5. Evolution of the cost function and its gradient during optimization.

**Nonlinear problem.** the linear data assimilation model problem (20) is changed by modifying the advection term:

$$\begin{cases} \frac{\partial u}{\partial x_{d+1}} + (v_d + u) \frac{\partial u}{\partial x_d} - v \frac{\partial^2 u}{\partial x_d^2} = 0, \\ u|_{x_{d+1}=0} = c(x_d), \\ u|_{x_d=-1} = u|_{x_d=+1} = 0. \end{cases} \quad (25)$$

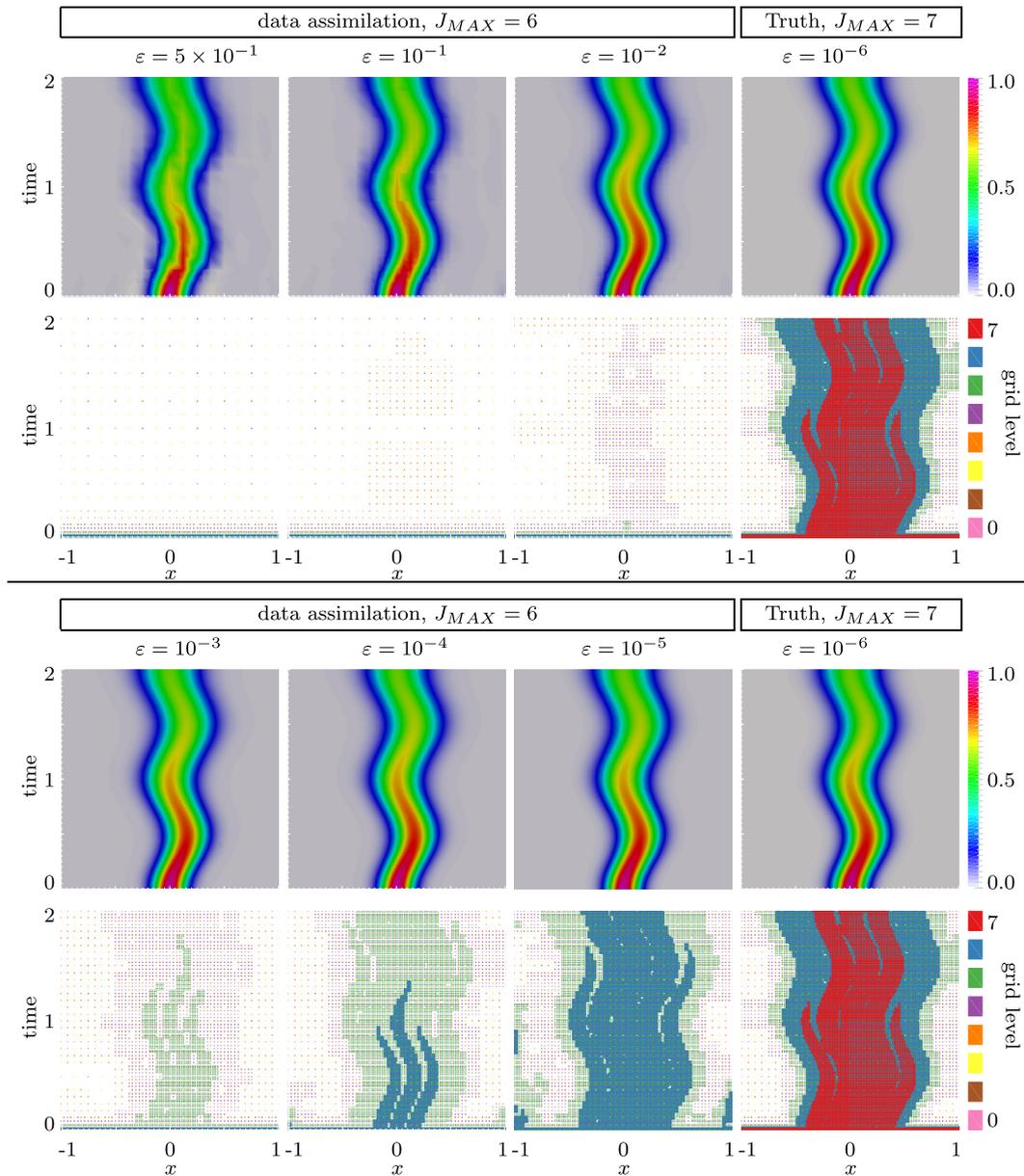
The associated adjoint model problem then becomes

$$\begin{cases} \frac{\partial p}{\partial x_{d+1}} + (v_d + u) \frac{\partial p}{\partial x_d} + v \frac{\partial^2 p}{\partial x_d^2} = \left[ \frac{\partial \mathbf{h}}{\partial \mathbf{u}} \right]^* (h(u) - y^o), \quad \mathbf{x} \in \Omega, \\ p|_{x_{d+1}=T} = 0, \\ p|_{x_d=-1} = p|_{x_d=+1} = 0. \end{cases} \quad (26)$$

Note that in contrast to the linear case, the solution of nonlinear space–time adaptive 4D-Var problem necessitates the use of nonlinear solver. In this paper the nonlinear 4D-Var problem is solved using STAWCM with Wavelet Full Approximate Solution (WFAS) nonlinear elliptic solver [8]. Also note, that for nonlinear 4D-Var model problem the solution is steepening on the right side of the structure resulting in appearance smaller scales. For that purpose the effective resolution is increased to  $j_{\max} = 9$  correspondingly. The threshold parameter is kept the same as for the linear case, namely  $\epsilon$  is set to  $10^{-6}$ . For the actual nonlinear data assimilation problem, the same trend was followed as in the linear case. The level of  $j_{\max}$  was lowered to 8, while the same coarsest level resolution  $32 \times 4$  is used.

## 6. Results and discussion

The results of this section demonstrate the feasibility of the simultaneous space–time adaptive approach to 4D-Var. The parameters described in Section 5.1 are used to generate observations. The advection velocity is given by equation (22) and is illustrated in Fig. 1. The exact control variable is given by equation (23). Fig. 2 shows the true solution and corresponding



**Fig. 6.** The analyzed solution (top row) and the corresponding computational grid (bottom) for different values of the threshold parameter. The column 4 is the true solution with the optimal value of the threshold parameter. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

grid for  $j_{\max} = 7$  and  $\epsilon = 10^{-6}$ . Each level of the wavelet decomposition is plotted using a different color in the grid. It is important to recall that in the collocation method, each wavelet is uniquely associated with a collocation point [8,18]. Note that the high spatio-temporal resolution is only present in the immediate vicinity of the localized structures and around time 0. The latter is because all the points associated with the control vector (solution at time 0) are forced to be present on the grid. For the data assimilation problem, the adaptation parameter is set to  $10^{-3}$  introducing a truncation error in the model. In addition, the maximum level of wavelet decomposition is set to 6 compared to 7 for the true solution (see Section 5.1). It is verified that with  $\epsilon = 10^{-3}$  and  $j_{\max} = 6$ , the model error is dominated by the truncation error due to threshold parameter. The background control vector (initial guess) is chosen to be  $\frac{3}{4}$  of the true solution. Fig. 3 shows the solution and the corresponding computational grid obtained with space-time data assimilation approach. The analyzed solution carries all the visible features of the true solution (Fig. 2). The grid is considerably coarser mainly due to larger value of the threshold parameter used. The convergence of the analysis to the true solution is demonstrated in Fig. 4, where the true control variable, the analyzed control variable and the analysis error are shown. The analysis error is of the order

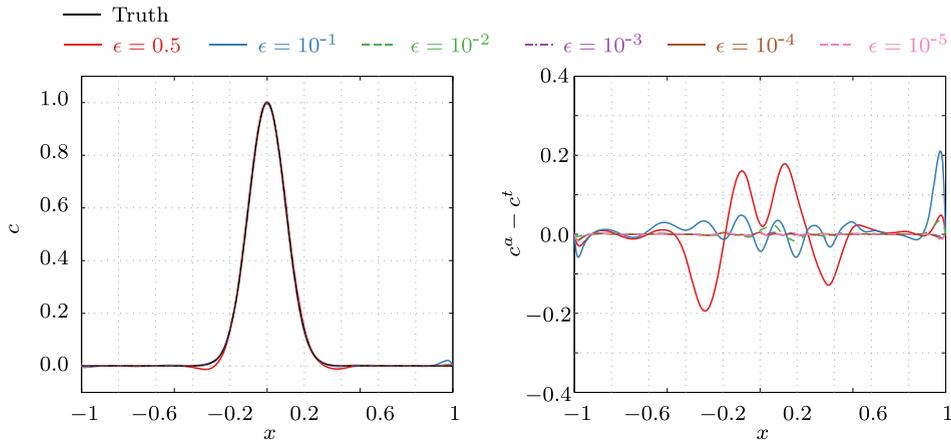


Fig. 7. Analyzed control variable: true solution and analysis (left); analysis error (right). (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

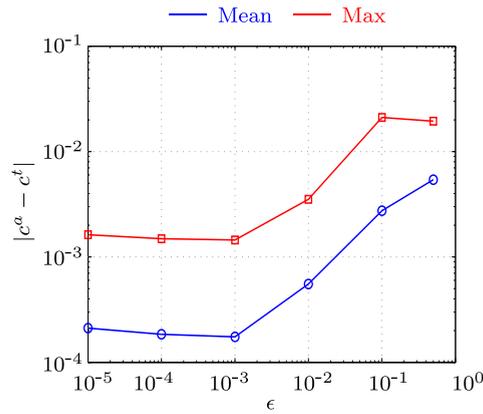


Fig. 8. Evolution of the norm of the error in the analyzed control variable with  $\epsilon$ . (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

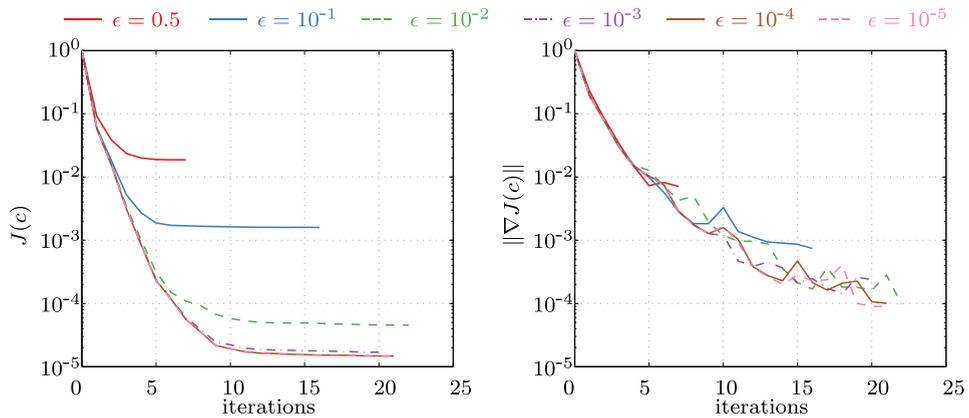
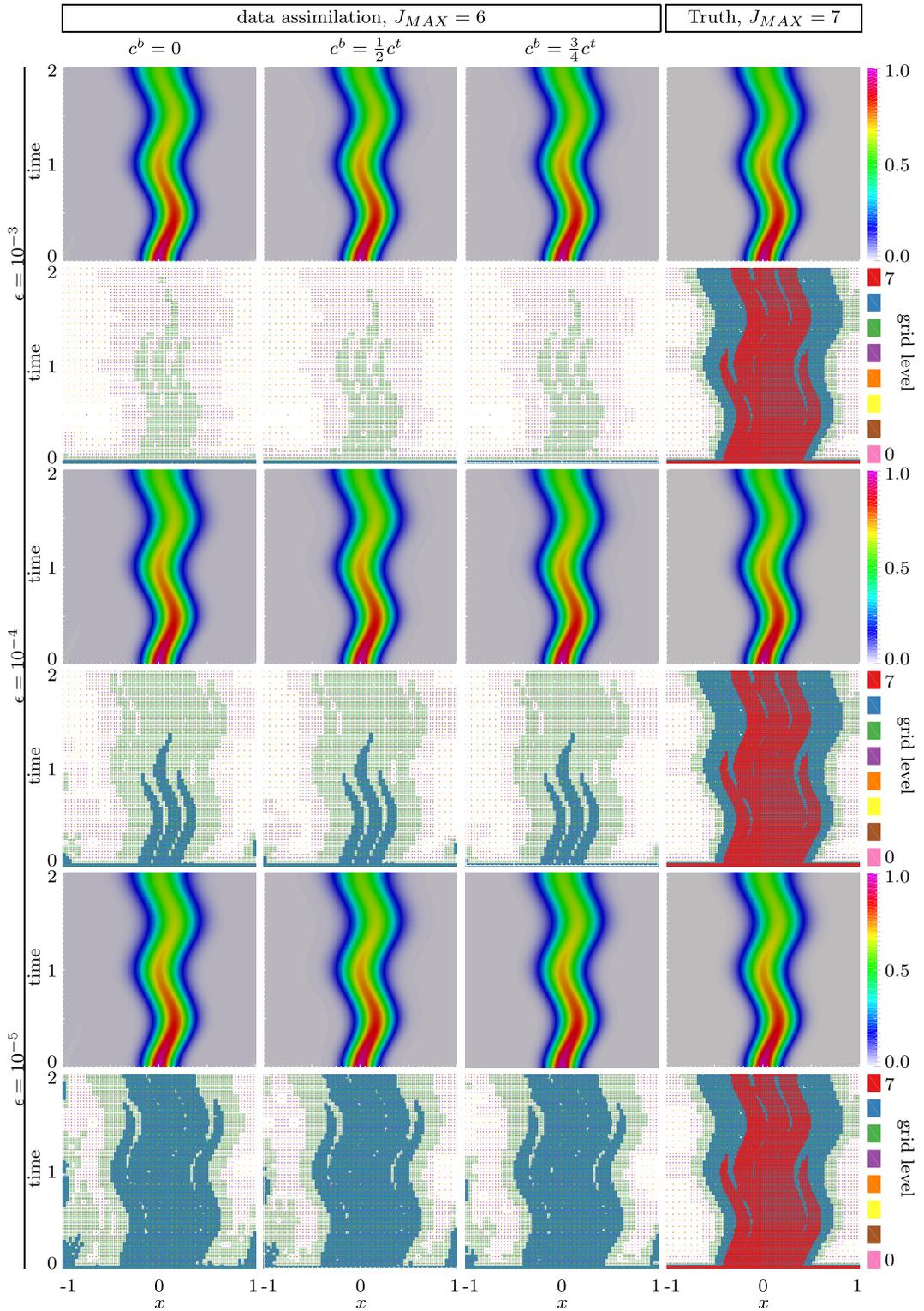


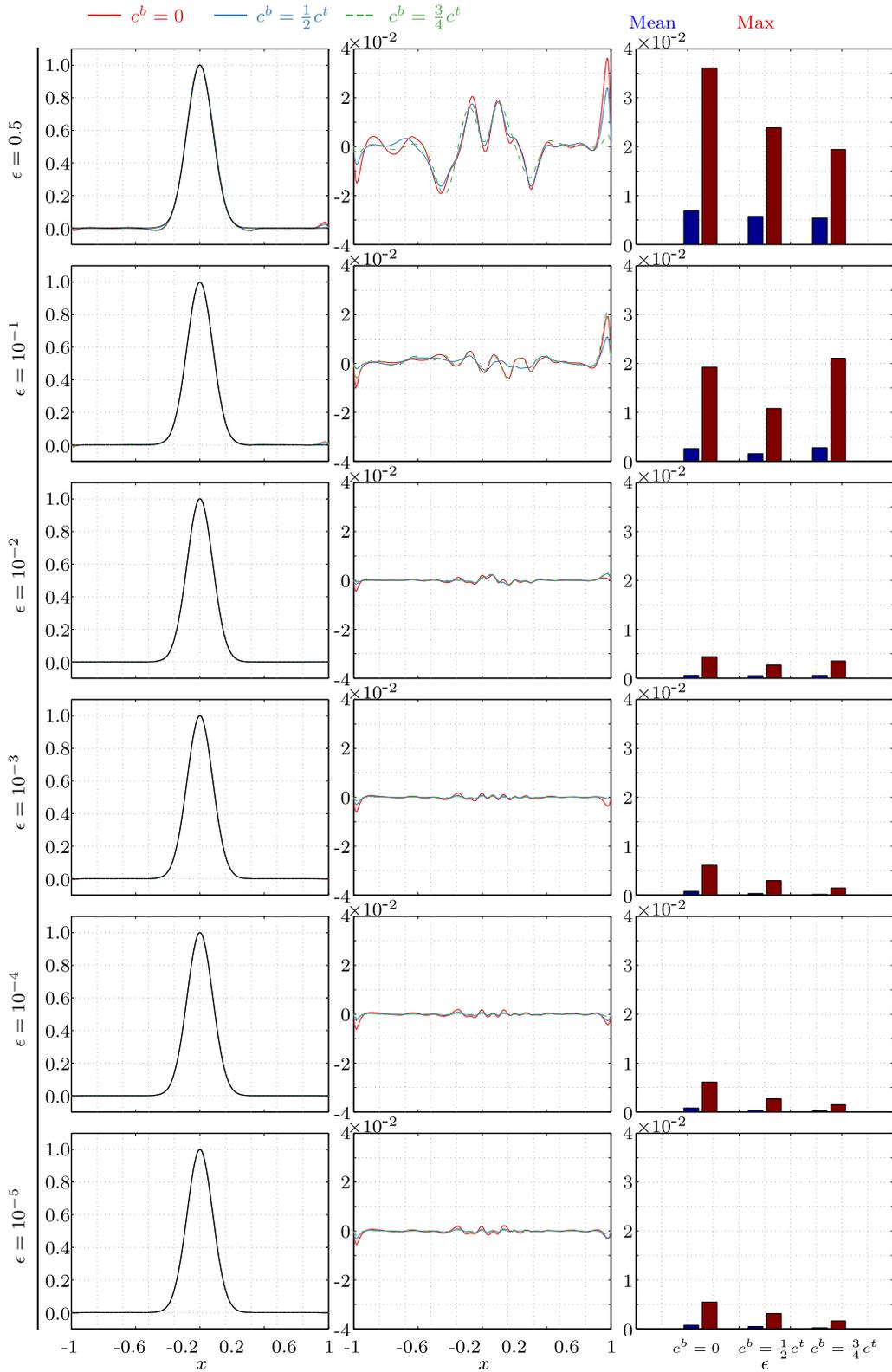
Fig. 9. Evolution of the cost function and its gradient during optimization. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

$O(10^{-3})$ , which corresponds to the truncation error associated with the threshold parameter  $\epsilon = 10^{-3}$ , thus, it is in perfect agreement with the global control of error embedded in the STAWCM.

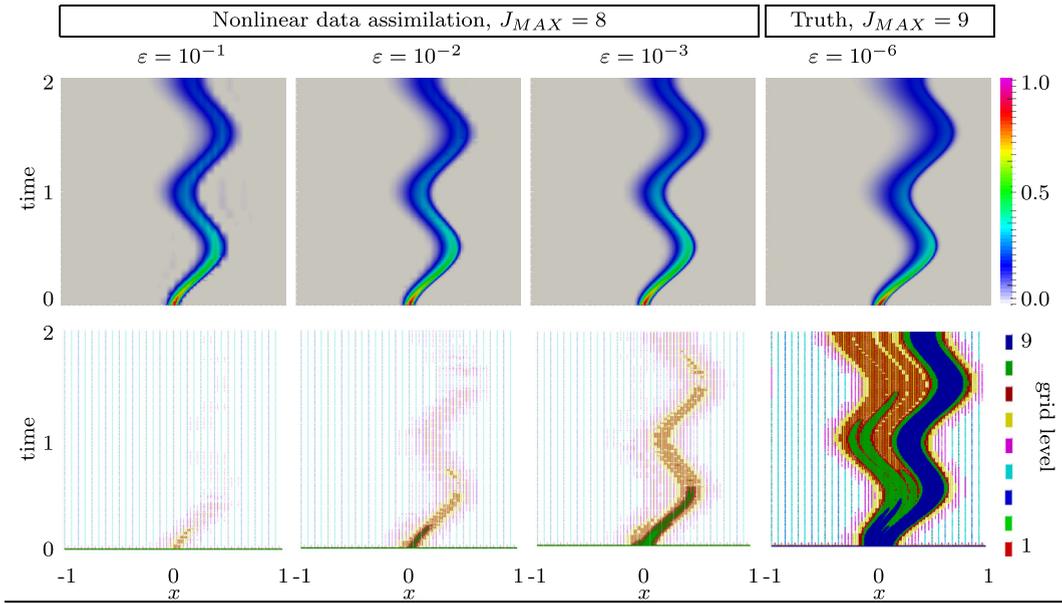
Fig. 5 shows the evolution of the cost function  $J$  and the norm of its gradient  $\|\nabla J\|$  during the minimization process. The values plotted in the figure are divided by the initial iteration values of  $J$  and  $\|\nabla J\|$ , respectively. The cost function saturates after 10 iterations, while the gradient continues to decrease almost monotonically. When converged, the cost function is reduced by the order of  $10^5$  and its gradient is reduced by the order of  $10^4$ .



**Fig. 10.** Grid and solution for different backgrounds and different values of  $\epsilon$ . (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)



**Fig. 11.** Control variable (left column) and the corresponding error (middle column) for different backgrounds and different values of  $\epsilon$ . The right column shows the mean and the maximum  $l_2$ -norm of the error. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)



**Fig. 12.** The analyzed solution of nonlinear 4D-Var (top row) and the corresponding computational grid (bottom) for different values of the threshold parameter. The column 4 is the true solution with  $\epsilon = 10^{-6}$ . (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

6.1. Influence of the adaptation parameter in the data assimilation

One of the main differences of the proposed space–time adaptive 4D-Var approach compared to the traditional time marching approaches is the availability of the complete space–time solution in a compressed form. The space–time representation of the data shifts the focus onto the efficient representation of the compressed solution. To study the influence of the grid adaptation in the data assimilation process, twin experiments with a variation in  $\epsilon$  are setup. The following values were used:  $5 \times 10^{-1}$ ;  $10^{-1}$ ;  $10^{-2}$ ;  $10^{-3}$ ;  $10^{-4}$  and  $10^{-5}$ . Apart from the variation in  $\epsilon$ , all other parameters were the same as in Section 5.1. The variation in  $\epsilon$  has a direct influence on the model error: larger values of  $\epsilon$  reduce the accuracy of the model and increase the compression rate of the solution by eliminating more wavelets in its expansion. Thus, there is a trade-off between the compression and the accuracy of the solution.

Fig. 6 shows the solution and the computational grid for different values of  $\epsilon$ . The true solution and the corresponding computational grid are also plotted for visual comparison. From larger to smaller  $\epsilon$ , the solution and the grid are refined and get close to the true solution. For  $\epsilon = 5 \times 10^{-1}$ , only wavelets of levels 0 to 2 are used to reconstruct the analyzed solution that already captures the main features of the true solution. Recall that the adaptation is deactivated in the vicinity of time  $t = 0$ . For  $\epsilon = 10^{-1}$ , the grid includes wavelets up to level 3. For  $\epsilon = 10^{-2}$ , the grid includes points up to level 5 and the analyzed solution looks like the true solution. For  $\epsilon = 10^{-3}$ , no wavelets at the higher level of resolution are added, but the grid includes more points and the solution is visually indistinguishable from the true solution. For  $\epsilon = 10^{-4}$  and  $10^{-5}$ , the grid includes the sixth level of resolution and the grid starts developing features that are not present in the true solution.

For quantitative analysis, Fig. 7 shows the true solution and the analyzed control variable for different values of  $\epsilon$  as well as the error (difference between the analyzed and the true control variable). Note that the spurious grid points for  $\epsilon = 10^{-4}$  and  $10^{-5}$  come from the analysis error at the boundary. These points are not present for the larger threshold values. The dependence of the error of the solution on the threshold parameter  $\epsilon$  is shown Fig. 8, where the saturation of the error for threshold values  $\epsilon \lesssim 10^{-3}$  is observed.

The evolution of the cost function  $J$  and the norm of its gradient  $\|\nabla J\|$  during the minimization process is shown in Fig. 9. The values plotted in the figure are divided by the initial values of  $J$  and  $\|\nabla J\|$ , respectively. The behavior is similar to Fig. 5. The smaller  $\epsilon$  is, the smaller the minimum of the cost function. However, there is almost no difference between the converged cost function for  $\epsilon \lesssim 10^{-3}$ . The cost function converges very rapidly for large  $\epsilon$ . It is important to recall that the computation time is proportional to the number of grip points used in the simulation. Thus, it is beneficial to use larger values of the threshold parameters at the early iterations, when the error is large, and progressively decrease  $\epsilon$  to the desired accuracy.

6.2. Influence of the background

To study the influence of the background in the proposed approach of data assimilation, the following numerical experiments with different values of the background (initial guess) are considered:  $c^b = 0$ ,  $c^b = \frac{1}{2}c^t$ , and  $c^b = \frac{3}{4}c^t$ , where  $c^b$  is

the initial guess and  $c^f$  is the exact control variable. The only change compared to the previous result is the variation of the background. In addition, different values of  $\epsilon$  :  $5 \times 10^{-1}$ ;  $10^{-1}$ ;  $10^{-2}$ ;  $10^{-3}$ ;  $10^{-4}$  and  $10^{-5}$  are also considered. Fig. 10 shows the analyzed solution and the corresponding grid for different values of  $c^b$  and  $\epsilon$ . The true solution is also presented for visual comparison. The results for  $\epsilon$  :  $5 \times 10^{-1}$ ;  $10^{-1}$  and  $10^{-2}$  are not presented because for each value of  $\epsilon$ , there is no significant difference between the results corresponding to different  $c^b$ . Figure shows that for a larger difference between the background and the true solution, the spurious grid points appear early and their number is considerable, while there is no visible difference between solutions. This can be further seen in Fig. 11 where the analyzed control variables, the associated errors, and their norms for different values of  $\epsilon$  and  $c^b$  are shown. For larger values of the threshold parameter, ( $\epsilon \gtrsim 10^{-2}$ ), the model error is dominant, while for smaller  $\epsilon$  ( $\epsilon \lesssim 10^{-3}$ ), the background error becomes dominant, thus, confirming the observation of Fig. 10.

### 6.3. Results of the nonlinear data assimilation problem

The results from the nonlinear 4D-Var problem are displayed in Fig. 12. As clearly observed, the nonlinear problem follows a very similar trend to that of the linear problem discussed above. Even with the largest adaptation parameter  $\epsilon$ , the shape of the solution can already be seen. With the decrease of  $\epsilon$  the solution approaches the true solution on a progressively refined space–time adaptive computational mesh.

## 7. Conclusion

The simultaneous space–time formulation of 4D-Var has been presented. In combination with the automatic adaptivity and the control of the accuracy of both the direct and the adjoint models, this formulation is more attractive than the conventional formulation with time–marching discretization. The simultaneous space–time discretization provides a natural space–time adaptation of the spatial and temporal scales present in the solution. The actual time step is smaller in regions of rapid changes (small local time scale) and larger in slowly varying regions (large local time scale). Numerical results in two space–time dimensions demonstrate the feasibility of the approach. The space–time formulation is particularly attractive for problems for which the solution can be compressed in the wavelet space. This is the case of most problems with highly localized intermittent structures. For extremely large problems, when the number of space–time degrees of freedom required to solve the problem using simultaneous space–time discretization cannot be afforded due to the limitation of available computer memory, one can still take advantage of the grid adaptation in combination with the time marching method. The time step can then be adapted to the resolution of the mesh at each point of the spatial grid. Such a strategy would be similar to the hierarchical method of adaptive mesh refinement [21,22].

## Acknowledgements

OVV was partially supported by NSF under grant No. CBET-1236505. This support is gratefully acknowledged.

## References

- [1] Magnus Rudolph Hestenes, Eduard Stiefel, Methods of conjugate gradients for solving linear systems, *J. Res. Natl. Bur. Stand.* 49 (6) (1952).
- [2] M.R. Hestenes, *Conjugate Direction Methods in Optimization*, Appl. Math., Springer-Verlag, 1980.
- [3] J.E. Dennis Jr., Robert B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, 1996, pp. 86–110, Chapter 5.
- [4] J. Ortega, W. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Class. Appl. Math., Society for Industrial and Applied Mathematics, 1987.
- [5] J.M. Lewis, S. Lakshminarayanan, S.K. Dhall, *Dynamic Data Assimilation: A Least Squares Approach*, Cambridge University Press, 2006.
- [6] A. Griewank, Achieving logarithmic growth of temporal and spatial complexity in reverse automatic differentiation, *Optim. Methods Softw.* 1 (1992) 35–54.
- [7] P. Courtier, J.N. Thepaut, A. Hollingsworth, A strategy for operational implementation of 4D-var, using an incremental approach, *Q. J. R. Meteorol. Soc.* 120 (519, B) (1994) 1367–1387.
- [8] J. Alam, N.K.R. Kevlahan, O.V. Vasilyev, Simultaneous space–time adaptive wavelet solution of nonlinear partial differential equations, *J. Comput. Phys.* 214 (2) (2006) 829–857.
- [9] N.K.R. Kevlahan, J.M. Alam, O.V. Vasilyev, Scaling of space–time modes with Reynolds number in two-dimensional turbulence, *J. Fluid Mech.* 570 (2007) 217–226.
- [10] A. Fournier, Introduction to orthonormal wavelet analysis with shift invariance: application to observed atmospheric blocking spatial structure, *J. Atmos. Sci.* 57 (23) (2000) 3856–3880.
- [11] A.K. Alekseev, I.M. Navon, The analysis of an ill-posed problem using multi-scale resolution and second-order adjoint techniques, *Comput. Methods Appl. Mech. Eng.* 190 (15–17) (2001) 1937–1953.
- [12] A. Tangborn, Wavelet approximation of error covariance propagation in data assimilation, *Tellus, Ser. A Dyn. Meteorol. Oceanogr.* 56 (1) (2004) 16–28.
- [13] M. Belo Pereira, L. Berre, The use of an ensemble approach to study the background error covariances in a global NWP model, *Mon. Weather Rev.* 134 (9) (2006) 2466–2489.
- [14] R.N. Bannister, Can wavelets improve the representation of forecast error covariances in variational data assimilation?, *Mon. Weather Rev.* 135 (2) (2007) 387–408.
- [15] H. Varela, L. Berre, G. Desroziers, Diagnostic and impact studies of a wavelet formulation of background-error correlations in a global model, *Q. J. R. Meteorol. Soc.* 137 (658, A, SI) (2011) 1369–1379.
- [16] A.N. Tikhonov, Regularization of incorrectly posed problems, *Sov. Math.* 4 (1963) 1624–1627.

- [17] F.-X. Le Dimet, O. Talagrand, Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects, *Tellus* 38A (1986) 97–110.
- [18] O.V. Vasilyev, N.K.R. Kevlahan, An adaptive multilevel wavelet collocation method for elliptic problems, *J. Comput. Phys.* 206 (2) (2005) 412–431.
- [19] D.L. Donoho, Interpolating wavelet transforms, Tech. rep. 408, Department of Statistics, Stanford University, 1992.
- [20] J.C. Gilbert, X. Jonsson, LIBOPT – an environment for testing solvers on heterogeneous collections of problems – version 1.0, Tech. rep. RT-0331, INRIA, March 2007.
- [21] H.J. Neeman, Autonomous hierarchical adaptive mesh refinement for multiscale simulations, Tech. rep., Champaign, IL, USA, 1996.
- [22] X. Li, M. Parashar, Hierarchical partitioning techniques for structured adaptive mesh refinement applications, *J. Supercomput.* 28 (3) (2004) 265–278.