Optimal reduced space for Variational Data Assimilation

Rossella Arcucci [a,b,*], Laetitia Mottet [c,d], Christopher Pain [c,b], Yi-Ke Guo [a,b]

[a] Data Science Institute, Department of Computing, Imperial College London, UK
[b] Data Assimilation Lab, Data Science Institute, Imperial College London, UK
[c] Department of Earth Science & Engineering, Imperial College London, UK
[d] Department of Architecture, University of Cambridge, UK

A R T I C L E   I N F O

A B S T R A C T

Data Assimilation (DA) is an uncertainty quantification technique used to incorporate observed data into a prediction model in order to improve numerical forecasted results. Variational DA (VarDA) is based on the minimisation of a function which estimates the discrepancy between numerical results and observations. Operational forecasting requires real-time data assimilation. This mandates the choice of opportune methods to improve the efficiency of VarDA codes without loosing accuracy. Due to the scale of the forecasting area and the number of state variables used to describe the physical model, DA is a big data problem. In this paper, the Truncated Singular Value Decomposition (TSVD) is used to reduce the space dimension, alleviate the computational cost and reduce the errors. Nevertheless, a consequence is that important information is lost if the truncation parameter is not properly chosen. We provide an algorithm to compute the optimal truncation parameter and we prove that the optimal estimation reduces the ill-conditioning and removes the statistically less significant modes which could add noise to the estimate obtained from DA. In this paper, numerical issues faced in developing VarDA algorithm include the ill-conditioning of the background covariance matrix, the choice of a preconditioning and the choice of the regularisation parameter. We also show how the choice of the regularisation parameter impacts on the efficiency of the VarDA minimisation computed by the L-BFGS (Limited – Broyden Fletcher Goldfarb Shanno). Experimental results are provided for pollutant dispersion within an urban environment.

© 2018 Elsevier Inc. All rights reserved.

1. Introduction and motivation

Numerical simulations are widely used as a predictive tool to better understand complex air flows and pollution transport on the scale of individual buildings, city blocks and entire cities. For these complex phenomena, knowledge about the state of a system and the governing physical processes is often incomplete, inaccurate, or both. The current approach in numerical modelling (which includes air pollution predictions) consists in simulating explicitly only the largest-scale phenomena, while taking into account the smaller-scale ones by means of physical parameterisations. Due to the inability to resolve the full spectrum of physical mechanisms involved as well as the fundamentally stochastic nature of the turbulent processes, all numerical models introduce uncertainty through the selection of scales and parameters that are somewhat ambiguous. Additionally, any computational methodology contributes to uncertainty due to discretisation, finite precision

---

* Corresponding author.
  E-mail address: r.arcucci@imperial.ac.uk (R. Arcucci).

and the consequent accumulation and amplification of round-off errors. Taking into account these uncertainties is essential for the acceptance of any numerical simulation.

Uncertainty quantification is now permeating the science workload. The demand for predictive science results is driving the development of improved approaches for establishing levels of confidence in computational predictions using Data Assimilation (DA) methodologies. Data Assimilation is an uncertainty quantification technique used to incorporate observed data into a prediction model in order to improve numerical forecasted results [1].

There are many DA methods [1,2] which have been mostly custom-developed on the forecasting model with which they are combined. Those which have gained acceptance as powerful methods in the last ten years are the variational DA (VarDA) approaches [3,4] based on the minimisation of a function which estimates the discrepancy between numerical results and observations assuming that the two sources of information, forecast and observations, have errors that are adequately described by error covariance matrices.

VarDA model needs to satisfy the operational forecasting requesting real-time utilisation of DA to improve predictions. As there is insufficient time to restart a run from the beginning with new data, the information provided by observations must be incorporated on the fly. Moreover, during a single run, repeated corrections to data are required to bring the code output into agreement with the latest data. The most popular DA software using a VarDA model compute the minimum of the DA function by a Conjugate Gradient (CG) algorithm [5,6]. The main computational kernel is then the solution of a linear system [1,7] which is strongly ill-conditioned [7,8] due to the background error covariance matrices. This mandates the introduction of preconditioning methods in DA software.

In summary, the necessity to run DA in real-time implies proper choices of numerical algorithms to regularise the ill-posed problem, to compute the minimum as well as to introduce preconditioning.

2. Related works and contribution of the present work

During the last 20 years, algorithms for DA have been investigated by a number of federal research institutes and universities. Up to now, the main efforts towards the development of Variational DA systems were achieved in numerical weather prediction applications, namely by the ECMWF (European Center for Medium-Range Weather Forecasts), in Reading (UK) and by the NCAR (National Center for Atmospheric Research), in Colorado (USA). Variational DA models, IS4DVAR and NEMOVAR, were developed for ocean general circulation models: the Regional Ocean Modeling System (ROMS) [6] and the Nucleus for European Modeling of the Ocean (NEMO) [5] respectively. These software products are mostly custom-developed on the forecasting model with which they are combined. The strong dependencies of the codes on the application domains, the data and the type of assimilated observations, do not allow a simple use of these codes in general cases. In this paper, the problem of assimilating data to improve prediction of air flows and pollution transport is addressed using a Variational DA model. Simulations are performed using the open-source, finite-element, fluid dynamics model Fluidity (http://fluidityproject.github.io/). The details of the equations solved and their implementation can be found in [9–11]. The state variable consists of values of pressure, velocity and pollution concentration. Observed values of the state variable are assumed from positions mainly located on the roof of the buildings. However, the algorithm and numerical methods proposed in this work can be applied to other physical problem involving other equations and/or state variables.

Over the last years, several CG methods have been developed in order to accelerate the convergence of the minimisation process. Most of them are efficient improvements of the Preconditioned Conjugate Gradient (PCG) method [12]. When the quadratic minimisation is performed directly in the space where the state variable is defined, the method is referred to as the *primal* approach. However, due the increasing number of available observed data, a popular approach consists in performing the minimisation in the space where the observed values are defined: the so-called *dual* space. In [13], the authors compute the minimum in the *dual* space defining a Restricted Preconditioned Conjugate Gradient (RPCG) algorithm. They show how the RPCG allows preconditioning and reorthogonalisation making this method suitable for large-scale applications. However, they did not implement this technique in a real-life data assimilation system where the large size of the state variable makes RPCG an interesting alternative to the PCG algorithm. In [14], the authors proposed a new *dual* algorithm: the Restricted B-preconditioned Lanczos method (RBLanczos) where **B** referred to the background-error covariance matrix. This algorithm, can be viewed as the Lanczos version of a RPCG special case in which the corresponding *primal* first-level preconditioner is **B**. Hence, they call this specific algorithm Restricted B-preconditioned Conjugate Gradient (RBCG). In [14], the authors provide a practical demonstration of the benefits of RBCG and RBLanczos using two operational variational data assimilation systems for the ocean (NEMOVAR and ROMS). For the prediction of air flow and pollution transport in urban environment, the number of available observed data defined at small scale is not significant enough to compute the solution in the *dual* space. Therefore, in this paper, the L-BFGS (Limited-Broyden Fletcher Goldfarb Shanno) method, which has been proved to be fast for large scale optimisation problems [15], is adopted. An exhaustive study of CG and L-BFGS gradient based minimisation algorithms is covered by highly cited papers such as [16–19]. L-BFGS method is a Quasi Newton method [20] that can be viewed as an extension of conjugate-gradient methods in which the addition of some modest storage serves to accelerate the convergence rate. The convergence rate of L-BFGS depends on the conditioning of the numerical problem [15] which is dominated by the condition number of the background covariance matrix [21].

In order to reduce the ill-conditioning of the background covariance matrix and remove the statistically less significant modes which could add noise to the estimate obtained from DA, we use the Truncated Singular Values Decomposition (TSVD) method. In the TSVD method, only the first largest eigenvalues of the error covariance matrix are considered, thus

reducing the dimension and alleviating the computational cost (less expensive run). Nevertheless, a consequence of using this method is that important information is lost [22]. This issue represents a severe drawback to the reliability of the TSVD if the truncation parameter is not properly chosen. The proper choice of the truncation parameter is usually related to a reference solution [23] or statistically related to the variance of the full spectrum of the error covariance matrix. However, in operational Data Assimilation, the knowledge of this reference solution represents a strong condition. Also, as the error covariance matrix has a dimension $(N \times N)$ related to the size N of the domain, the computation of the spectrum for big domains, requiring $\mathcal{O}\left((N \times N)^3\right)$, is often prohibitively expensive. In order to alleviate the computational cost, several approaches, mainly based on a domain decomposition, have been developed. In [21,24], the authors employed a domain decomposition version of the Variational Data Assimilation model described in [25]. They studied the effects of the Empirical Orthogonal Functions (EOFs) based on the TSVD and the Tikhonov regularisation approach. They highlighted that the accuracy of the solution obtained by truncating EOFs has a strong sensitivity to the truncation parameter value. Hence, using TSVD can be problematic in operative software modelling different scenarios [26]. If the truncation parameter is properly chosen, TSVD is a powerful method that can be efficiently combined with domain decomposition to strongly alleviate the computational cost. In [27], the authors answered the problem concerning the selection of an optimal truncation parameter using a quantitative analysis of the condition number and the error related to data provided by the software Fluidity. Even if the approach in [27] provided good results on a simple test case, it has limitations in its usability for operational data assimilation as it requires the computation of the whole spectrum of the background error covariance matrix which is computationally prohibitive for big data problems (such as 3D realistic urban environment). The works presented in [27] are applied to the transport of pollution within an urban environment using the software Fluidity and can be seen as the premise of this paper. Hence, it becomes crucial to develop new methods able to chose the optimal truncation parameter and thus optimise the space reduction in the case of realistic 3D scenarios. In this paper, we provide a method to estimate the truncation parameter independently of the knowledge of an exact solution. Also, the approach we propose does not require the computation of the whole spectrum of the background error covariance matrix.

Our DA model and algorithm addresses the requirement of both efficiency and accuracy as described in the following:

- *Efficiency*: In order to alleviate the computational cost, we use:
  - the Truncated Singular Value Decomposition (TSVD) method. TSVD allows us strongly reduce the dimension of the problem making the running less expensive;
  - the L-BFGS (Limited-Broyden Fletcher Goldfarb Shanno) method to compute the minimum of the VarDA function. L-BFGS is proved to be the fastest for large scale optimisation problems [15]. In order to validate our choice, a comparison of the results obtained in terms of execution time by the L-BFGS and by the PCG algorithm [12] is provided;
- *Accuracy*: The use of TSVD reduces the ill-conditioning and removes the statistically less significant modes which could add noise to the estimate obtained from DA. The method we propose to estimate the truncation parameter is independent from the knowledge of an exact solution. The optimal truncation parameter is chosen to minimise both:
  - the condition number of the problem after the preconditioning;
  - a relative Preconditioning Error defined to provide an estimate of how much the preconditioned problem differs from the starting problem.

Variational approaches essentially implement a standard Tikhonov (or L^2) regularisation. Solution of a VarDA problem requires computing the minimum of a Tikhonov function which includes the choice of the Tikhonov regularisation parameter. In general, operational DA software assumes a regularisation parameter equal to one by assuming that the observations and the background state have the same relative weight. In this paper, we assume that this weight depends on the type of the observed data and their reliability. The possibility to set-up an arbitrary value of the regularisation parameter in our DA algorithm is then provided. The impact of the choice of different regularisation parameter on the performance and the accuracy of the DA solution is evaluated. The impact of our VarDA model on Fluidity in terms of accuracy of prediction is also studied.

The paper is structured as follows. In Section 3, some preliminary definitions are introduced. In Section 4, the Data Assimilation problem is described and definition of Variational approaches to solve it is presented. Section 5 describes the VarDA algorithm as well as the algorithm that we have developed to compute the TSVD by the optimal truncation parameter. Numerical issues are also discussed. Experimental results on realistic test cases are provided in Section 6. Section 7 concludes this work and describes future works.

3. Preliminary definitions

In this section, we present some definitions that we will use in the next sections.

Definition 1 (*Variance-covariance matrix*). Let $\mathbf{X}$ be a matrix of measurements of pv physical variables at spatial locations $\mathcal{D} = \{x_j\}_{j=1,\dots,np}$, and at a correlation time window $[0, T_1] = \{\tau_k\}_{k=1,\dots,M}$:

$$\mathbf{X} = \begin{bmatrix} X_1 \\ \vdots \\ X_{NP} \end{bmatrix} \in \Re^{NP \times M} \tag{1}$$

where each of NP row is a time series for a given location and $NP = [pv] \cdot np$. Let assume that each row X_i of $\mathbf{X}$ has mean $E[X_i] = \{m_i\}_{i=1,\ldots,NP}$ and let $\mathbf{m} = (m_i)_{i=1,\ldots,NP}$. Let

$$\mathbf{V} = \mathbf{X} - \mathbf{m} \in \Re^{NP \times M}, \tag{2}$$

be the deviation matrix. The variance-covariance matrix $\mathbf{B} \in \Re^{NP \times NP}$ of $\mathbf{X}$ is defined via the expected value[1] of the outer product:

$$\mathbf{B} = \mathbf{V}\mathbf{V}^T \quad . \tag{3}$$

Definition 2 (*Singular value decomposition*). Let $\mathbf{A} \in \Re^{N \times M}$ where $M \geq N$ and let

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{W}^T \tag{4}$$

be the singular value decomposition (SVD) of $\mathbf{A}$ where $\mathbf{U} \in \Re^{N \times N}$ and $\mathbf{W} \in \Re^{M \times M}$ are orthogonal (or orthonormal) matrices and

$$\mathbf{\Sigma} = diag(\sigma_j)_{j=1,\ldots,N} \tag{5}$$

where singular values σ_j appear in decreasing order:

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_N > 0 \quad . \tag{6}$$

Definition 3 (*Determinant*). Let $\mathbf{A} \in \Re^{N \times M}$ where $M \geq N$ and let

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{W}^T \tag{7}$$

be the singular value decomposition (SVD) of $\mathbf{A}$ as in Definition 2. Then the determinant of the matrix $\mathbf{A}$ is given by the product of the N singular values:

$$det(\mathbf{A}) = \prod_{i=1}^{N} \sigma_i . \tag{8}$$

Definition 4 (*Condition number*). Let $\mathbf{A} \in \Re^{N \times M}$ where $M \geq N$ and let

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{W}^T \tag{9}$$

be the singular value decomposition (SVD) of $\mathbf{A}$ as in Definition 2. Then the condition number of $\mathbf{A}$ is such that

$$\mu(\mathbf{A}) = \frac{max\{\sigma_j\}_{j=1,\ldots,N}}{min\{\sigma_j\}_{j=1,\ldots,N}} \tag{10}$$

as singular values σ_j appear in decreasing order, from (6), we have

$$\mu(\mathbf{A}) = \frac{\sigma_1}{\sigma_N} . \tag{11}$$

If $\mathbf{A}$ is a matrix of an over-determined linear system, then the discrete problem is ill-posed and the contribution to the solution corresponding to the smallest singular values needs to be filter out [23,28]. Filtering can be introduced using Truncated Singular Value Decomposition as given in the following definitions:

---

[1] If each vector X_i has a distribution with probability density function P, then the expected value of X_i is defined by

$$E(X_i) = \frac{1}{M-1} \sum_{j=1,\ldots,M} x_{ij} P(X_j) \quad .$$

Definition 5 *(Truncated singular value decomposition).* Let $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{W}^T$ be the SVD of $\mathbf{A}$ as in Definition 2. Let $\boldsymbol{\Phi}_\tau \in \Re^{N \times N}$ be a matrix such that

$$\boldsymbol{\Phi}_\tau = diag(\sigma_1, \sigma_2 \ldots \sigma_\tau, 0 \ldots, 0) \quad , \tag{12}$$

with $1 \leq \tau \leq N$. Then the matrix

$$\mathbf{A}_\tau := \mathbf{U}\boldsymbol{\Phi}_\tau\mathbf{W}^T, \tag{13}$$

is the truncated SVD (TSVD) matrix for $\mathbf{A}$.

4. The DA problem and the VarDA formulation

The method described in this section is the most general VarDA method. It is called four-dimensional (4D VarDA) because it takes into account observations that are distributed in space and over an interval of time $[t_i, t_{i+\Delta t}]$. If $\Delta t = 0$, i.e. the time window is reduced to one instant, the method is called three-dimensional (3D VarDA) [1,22,29]. Let us give the mathematical settings describing the VarDA problem.

4.1. The DA model: set up and problem definition

Let $\Omega \subset \Re^3$ be a spatial domain and let:

$$\begin{cases} u(t_i, x) = \mathcal{M}[u(t_j, x)], & \forall x \in \Omega, \ t_i, t_j \in [0, T], (t_i > t_j > 0) \\ u(t_0, x) = u_0(x), & t_0 = 0, \quad x \in \Omega, \end{cases} \tag{14}$$

be a description of the forecasting model of interest where

$$u : (t, x) \in [0, T] \times \Omega \mapsto u(t, x) \tag{15}$$

is the state function of $\mathcal{M}$.

Let:

$$v : (t, x) \in [0, T] \times \Omega \mapsto v(t, x) \tag{16}$$

be the observation function and

$$\mathcal{H} : \quad u(t, x) \mapsto v(t, x), \qquad \forall (t, x) \in [0, T] \times \Omega \tag{17}$$

denotes the non-linear observation mapping. According to the applications of model-based assimilation of observations [5,6], we will use the following discrete formulation for the VarDA problem. Given

1. NP points of $\Omega \subset \Re^3$: $\{x_j\}_{j=1,\ldots,NP}$;
2. $nobs$ points of $\Omega \subset \Re^3$, where $nobs << NP$: $\{y_j\}_{j=1,\ldots,nobs}$;
3. N points of $[0, T]$: $\{t_k\}_{k=0,1,\ldots,N-1}$;
4. the background estimate, i.e. vector

$$\mathbf{u}_0 = \{u_0^j\}_{j=1,\ldots,NP} \equiv \{u(t_0, x_j)\}_{j=1,\ldots,NP} \in \Re^{NP} \tag{18}$$

   which is the state at time t_0;
5. the operator

$$\mathbf{M}_{k-1,k} \in \Re^{NP \times NP}, \quad k = 1, \ldots, N, \tag{19}$$

   representing a discretisation of a first order approximation of $\mathcal{M}$ from t_{k-1} to t_k;
6. the vector

$$\mathbf{v}_k \equiv \{v(t_k, y_j)\}_{j=1,\ldots,nobs} \in \Re^{N \times nobs}$$

   consisting of the observations at t_k, for $k = 0, \ldots, N - 1$;
7. the linear operator

$$\mathbf{H}_k \in \Re^{nobs \times NP}, \quad k = 0, \ldots, N - 1$$

   representing a linear approximation of the Jacobian of $\mathcal{H}$;

8. a block diagonal matrix $\mathbf{G} \in \Re^{(N \times nobs) \times (NP \times N)}$ such that

$$\mathbf{G} = \begin{cases} diag\,[\mathbf{H}_0, \mathbf{H}_1 \mathbf{M}_{0,1}, \ldots, \mathbf{H}_{N-1} \mathbf{M}_{N-2,N-1}] & N > 1; \\ \mathbf{H}_0 & N = 1. \end{cases} \tag{20}$$

9. the observation error covariance matrix $\mathbf{R} \in \Re^{(N \times nobs) \times (N \times nobs)}$ which describes the probability distribution function (pdf) of measurement errors. $\mathbf{R}$ is assumed to be defined as follows

$$\mathbf{R} = diag\,(\mathbf{R_k})_{k=0,\ldots,N-1} \quad \mathbf{R_k} := \sigma_0^2 \mathbf{I}, \tag{21}$$

with $0 \leq \sigma_0^2 \leq 1$ and $\mathbf{I} \in \Re^{nobs \times nobs}$ the identical matrix;

10. the background error covariance matrix $\mathbf{B} \in \Re^{NP \times NP}$ which describes the pdf of background errors. We assume that $\mathbf{B}$, defined as in Definition 1 where $T_1 > T$, is such that

$$\mathbf{B} = \sigma_b^2 \mathbf{C}, \tag{22}$$

where the matrix $\mathbf{C}$ denoting the correlation structure of the background error, is homogeneous, and the correlations depend only on distance between states and not position, i.e.

$$\mathbf{C}_{(NP,h,L)} = (c_{ij}), \quad c_{ij} = \exp\left(-\frac{1}{2}(j-i)^2 \cdot \|x_j - x_{j-1}\|_\infty^2\right), \tag{23}$$

with length scale $L = NP \cdot \|x_j - x_{j-1}\|_\infty$.

Given the DA problem set up, we now define the DA inverse problem.

Definition 6 *(The DA inverse problem).* Given the vectors

$$\mathbf{v} = (\mathbf{v}_k)_{k=0,\ldots,N-1} \in \Re^{N \times nobs}, \quad \mathbf{u}_0 \in \Re^{NP}$$

and the block diagonal matrix

$$\mathbf{G} \in \Re^{(N \times nobs) \times (NP \times N)}$$

a DA problem concerns the computation of

$$\mathbf{u}^{DA} = (\mathbf{u}_k^{DA})_{k=0,\ldots,N-1} \in \Re^{NP \times N},$$

such that

$$\mathbf{v} = \mathbf{G} \cdot \mathbf{u}^{DA}, \tag{24}$$

subject to the constraint that

$$\mathbf{u}_0^{DA} = \mathbf{u}_0 \quad . \tag{25}$$

Since $\mathbf{G}$ is typically rank deficient, the DA is an ill-posed problem [30,7]. In next section we define the variational formulation which leads to an unconstrained least square problem, where the term in equation (25) ensures the existence of a solution of the equation (24).

4.2. The VarDA model

In this section, description of the standard Variational Data Assimilation (VarDA) Model, the incremental VarDA model and the preconditioned VarDA formulation are provided.

Definition 7 *(The VarDA problem).* The VarDA problem can be described as follows:

$$\mathbf{u}^{DA} = argmin_{\mathbf{u} \in \Re^{NP \times N}} J(\mathbf{u}) \tag{26}$$

with

$$J(\mathbf{u}) = \alpha \|\mathbf{u} - \mathbf{u}_0\|_{\mathbf{B}^{-1}}^2 + \|\mathbf{G}\mathbf{u} - \mathbf{v}\|_{\mathbf{R}^{-1}}^2 \tag{27}$$

where, for any vector $w \in \Re^{NP}$ and $q \in \Re^{N \times nobs}$, $\|w\|_{\mathbf{B}^{-1}} = w^T \mathbf{B}^{-1} w$ and $\|w\|_{\mathbf{R}^{-1}} = w^T \mathbf{R}^{-1} w$. Parameter $\alpha > 0$ denotes the regularisation parameter. In general, operational DA software assume $\alpha = 1$. Choosing $\alpha = 1$ can be considered as giving the same relative weight to the observations in comparison to the background state.

If equation (27) is linearised around the background state [31] the VarDA problem is formulated by the following form:

Definition 8 *(The incremental VarDA problem).* The incremental VarDA problem can be described as follows:

$$\delta\mathbf{u}^{DA} = argmin_{\delta\mathbf{u}\in\Re^{NP\times N}} J(\delta\mathbf{u}) \tag{28}$$

with

$$J(\delta\mathbf{u}) = \frac{1}{2}\alpha\delta\mathbf{u}^T\mathbf{B}^{-1}\delta\mathbf{u} + \frac{1}{2}(\mathbf{G}\delta\mathbf{u} - \mathbf{d})^T\mathbf{R}^{-1}(\mathbf{G}\delta\mathbf{u} - \mathbf{d}) \tag{29}$$

where $\mathbf{d} = [\mathbf{v} - \mathbf{G}\mathbf{u}_0]$ is the misfit, $\mathbf{G}$ is here the linearised observational and model operators evaluated at $\mathbf{u} = \mathbf{u}_0$ and $\delta\mathbf{u} = \mathbf{u} - \mathbf{u}_0$ are the increments.

In equation (29) the minimisation problem is defined on the field of increments [29]. In order to avoid the inversion of $\mathbf{B}$ and to precondition the minimisation of the cost function, it is assumed that $\mathbf{B}$ can be written in the form $\mathbf{B} = \mathbf{V}\mathbf{V}^T$ (see equation (3)) and the cost function can be minimised using a new variable [31]:

Definition 9 *(The preconditioned VarDA problem).* The preconditioned VarDA problem can be described as follows:

$$\mathbf{w}^{DA} = argmin_{\mathbf{w}\in\Re^{NP\times N}} J(\mathbf{w}) \tag{30}$$

with

$$J(\mathbf{w}) = \frac{1}{2}\alpha\mathbf{w}^T\mathbf{w} + \frac{1}{2}(\mathbf{G}\mathbf{V}\mathbf{w} - \mathbf{d})^T\mathbf{R}^{-1}(\mathbf{G}\mathbf{V}\mathbf{w} - \mathbf{d}) \tag{31}$$

where $\mathbf{w} = \mathbf{V}^+\delta\mathbf{u}$ and $\mathbf{V}^+$ denotes the generalised inverse of $\mathbf{V}$.

The cost function in (31) can be expressed as follows:

$$J(\mathbf{w}) = J_b(\mathbf{w}) + J_o(\mathbf{w}) \tag{32}$$

where,

- $J_b(\mathbf{w}) = \frac{1}{2}\alpha\mathbf{w}^T\mathbf{w}$ is called first-guess cost function and
- $J_o(\mathbf{w}) = \frac{1}{2}(\mathbf{G}\mathbf{V}\mathbf{w} - \mathbf{d})^T\mathbf{R}^{-1}(\mathbf{G}\mathbf{V}\mathbf{w} - \mathbf{d})$ is called observation cost function.

5. The reduced space VarDA algorithm and numerical issues

This section describes Algorithm 1 and Algorithm 2 that we have developed to assimilate data from sensors in Fluidity by the preconditioned VarDA problem as described in equations (30)–(31). The numerical issues faced in developing these algorithms mainly include the ill-conditioning of the background covariance matrix, the preconditioning method and the choice of the truncation parameter (mainly faced in developing Algorithm 2) to determine the reduced space, and also, the method used to compute the minimum of the functional J in the reduced space in Algorithm 1.

5.1. Minimisation

As an important issue in Data Assimilation is to provide a result in real-time, the choice of an efficient method to compute the minimum of the functional J is a fundamental topic. One of the problems we address is the explosive growth in both on-line computer memory and remote storage requirements of large-scale assimilation studies. This implies a severe limitation on the size of assimilation studies, even on the largest computers [16]. By using a recursive strategy, a schedule can be constructed that enables the forward/adjoint model runs to be performed in such a way that storage requirements can be traded for longer computational times [19]. This strategy allows data assimilation to be performed on significantly larger domains than would otherwise be possible given particular hardware constraints. It was shown, that this trade-off is indeed viable and that when the schedule is optimized, the storage and computational times grow at most logarithmically [17,18]. In this paper, we compute the minimum of the linearised preconditioned functional J in equation (31) by the minimisation method proved to be the fasted for large scale optimisation problems [15], i.e. the L-BFGS (Limited-Broyden Fletcher Goldfarb Shanno) method [20]. The L-BFGS method is a Quasi-Newton method that can be viewed as extension of

---

Algorithm 1 The reduced space VarDA algorithm.

1: Input: α, $\{\mathbf{v}_k\}_{k=0,\ldots,m}$, $\mathbf{u}_0^M$, $\mathbf{G}$ and $\mathbf{G}^T$
2: Define $\mathbf{H}_k$
3: Compute $\mathbf{d}_k \leftarrow \mathbf{v}_k - \underbrace{\mathbf{G}\ldots\mathbf{G}}_{k}\mathbf{u}_0^M$ ▷ compute the misfit
4: Define $\mathbf{R}_k$ starting from the observed data $\mathbf{v}_k$
5: Compute $\mathbf{V}_\tau = PECM(\{\mathbf{u}_k^M\}_{k=0,\ldots,M})$ ▷ see Algorithm 2 for details
6: Define the initial value of $\delta\mathbf{u}^{DA}$
7: Compute $\mathbf{w} \leftarrow \mathbf{V}_\tau^+ \delta\mathbf{u}^{DA}$ ▷ from the physical to reduced space
8: repeat   ▷ start of the L-BFGS steps
9: Compute $J_b \leftarrow J_b(\mathbf{w})$ ▷ compute the first guess cost function
10: Compute $J_o \leftarrow J_o(\mathbf{w})$ ▷ compute the observation cost function
11: Compute $J \leftarrow J_b + J_o$ ▷ defined in (32)
12: Compute $grad\, J \leftarrow \nabla J(\mathbf{w})$ ▷ defined in (33)
13: Compute new values for $\mathbf{w}$
14: until (Convergence on $\mathbf{w}$ is obtained)   ▷ end of the L-BFGS steps
15: Compute $\delta\mathbf{u}^{DA} \leftarrow \mathbf{V}_\tau \mathbf{w}$ ▷ from the reduced to physical space
16: Compute $\mathbf{u}^{DA} \leftarrow \mathbf{u}_0^M + \delta\mathbf{u}^{DA}$

Algorithm 2 Preconditioning Error Covariance Matrix (PECM) algorithm.

1: Input: $\{u_{ki}^M\}_{k=0,\ldots,M\ i=1,\ldots,N}$ ▷ temporal sequence of historical data
2: Compute $\{\bar{u}_i\}_{i=1,\ldots,N} = mean_{k=0,\ldots,M}\left(\{u_{ki}^M\}_{i=1,\ldots,N}\right)$ ▷ compute the mean over the temporal sequence
3: Compute $\mathbf{V} = \{V_{ki}\}_{k=0,\ldots,M\ i=1,\ldots,N}$ such that $V_{ki} = u_{ki}^M - \bar{u}_i$ ▷ compute the matrix $\mathbf{V}$
4: Compute $\mathbf{V} = \mathbf{U}\mathbf{\Sigma}\mathbf{W}^T$.   ▷ compute the TSVD of the matrix $\mathbf{V}$
5: Set $e = \sqrt{\sigma_1}$
6: Set $\tau = 1$
7: for $i = 2,\ldots,M$ ▷ compute the truncation parameter provided in Theorem 1
8:     if ($\sigma_\tau < e$) then
9:         $\tau = \tau + 1$
10:    end if
11: end for
12: Compute $\mathbf{V}_\tau = TSVD(\mathbf{V}, \tau)$.   ▷ Truncated SVD regularised matrix $\mathbf{U}\mathbf{\Phi}_\tau\mathbf{W}^T$ in (13)

---

conjugate-gradient methods in which the addition of some modest storage[2] serves to accelerate the convergence rate. To compute the minimum by the L-BFGS method we need to compute the gradient of equation (31):

$$\nabla J(\mathbf{w}) = \alpha\mathbf{w} + \mathbf{V}^T\mathbf{G}^T\mathbf{R}^{-1}(\mathbf{G}\mathbf{V}\mathbf{w} - \mathbf{d}) \tag{33}$$

where $\mathbf{G}^T$ denotes the Adjoint Operator [32].

The convergence rate of L-BFGS depends on the conditioning of the numerical problem, i.e. it depends on the condition number of the preconditioned Hessian of $J(\mathbf{w})$:

$$\mathbf{A} = \mathbf{I} + \mathbf{V}^T\mathbf{G}^T\mathbf{R}^{-1}\mathbf{G}\mathbf{V}. \tag{34}$$

The condition number of the Hessian in equation (34) is dominated by the condition number of the matrix $\mathbf{V}$ [21] then, in the next section, a preconditioner for this matrix (see Step 5 of Algorithm 1) is introduced and described in Algorithm 2.

5.2. Reduced space and preconditioning

In this paper, we adopt the TSVD method in order to reduce the ill-conditioning and remove the statistically less significant modes which could add noise to the estimate obtained from DA. In order to improve the conditioning, only the first largest eigenvalues of the error covariance matrix are considered. Even if the employment of TSVD which strongly reduce the dimension, alleviate the computational cost, nevertheless, a consequence is that important information is lost [22]. This issue introduces a severe drawback to the reliability of the TSVD if the truncation parameter is not properly chosen.

As it is known that the numerical error which propagates into the DA solution is influenced by the condition number [21], a proper value of the truncation parameter should *minimise the condition number*. However, to be sure that the preconditioned problem does not depart too much from the original problem, the optimal truncation parameter should also

---

[2] The L-BFGS update formula generates the matrices approximating the Hessian using information from the last m Quasi-Newton iterations, where m is determined by the user (generally $3 \leq m \leq 30$). After having used the m vector storage locations for m quasi-Newton updates, the approximation of the Hessian matrix is updated by dropping the oldest information and replacing it by the newest information. Hence, time complexity of the L-BFGS increases linearly with the size of the m vectors [15].

minimise a Relative Preconditioning Error (RPE) which provides an estimate of how much the preconditioned problem differs from the starting problem as defined in Definition 10.

Definition 10 *(Relative Preconditioning Error (RPE)).* Let E_τ be the relative Preconditioning Error which provides an estimate of how much the preconditioned problem differs from the starting problem and defined as:

$$E_\tau = \frac{\|\mathbf{\Sigma} - \mathbf{\Phi}_\tau\|_\infty}{\|\mathbf{\Sigma}\|_\infty} \ . \tag{35}$$

The proper choice of the truncation parameter is usually related to a reference exact solution [23]. However, in operational Data Assimilation, the knowledge of this reference solution represents a strong constraint. To avoid this drawback, we provide, in Theorem 1, an estimation of the truncation parameter which is independent of the knowledge of an exact solution.

An optimal truncation parameter σ_{opt} should be picked to minimise both:

– the condition number of **V** after the preconditioning [28]

$$\mu(\mathbf{V}_\tau) \simeq \frac{\sigma_1}{\sigma_\tau} \ , \tag{36}$$

– the Relative Preconditioning Error (35).

The asymptotic behaviour of the condition number in (36) is

$$\lim_{\sigma_\tau \to 0} \mu(\mathbf{V}_\tau) = +\infty, \quad \lim_{\sigma_\tau \to +\infty} \mu(\mathbf{V}_\tau) = 0, \tag{37}$$

and of the Relative Preconditioning Error is

$$\lim_{\sigma_\tau \to 0} E_\tau = \frac{\|\mathbf{\Sigma} - \mathbf{I}\|_\infty}{\|\mathbf{\Sigma}\|_\infty} \simeq 1 - \frac{1}{\sigma_1}, \quad \lim_{\sigma_\tau \to +\infty} E_\tau = \frac{\|\mathbf{\Sigma}\|_\infty}{\|\mathbf{\Sigma}\|_\infty} = 1 \ . \tag{38}$$

As σ_τ is subject to the constraints $\sigma_1 \geq \sigma_\tau \geq \sigma_N$ [23], from equation (37), we find that the smallest value of the condition number is obtained for $\sigma_\tau \simeq \sigma_1$. From equation (38), however, the smallest error is obtained for $\sigma_\tau \simeq \sigma_N$.

Due to this difference in the asymptotic behaviour of the two functions E_τ and $\mu(\mathbf{V}_\tau)$, in order to find an optimal value, we consider the function

$$f(\sigma_\tau) = E_\tau + \mu(\mathbf{V}_\tau) \ . \tag{39}$$

The following result held

Theorem 1.

$$\sigma_{opt} = argmin \ f(\sigma_\tau) = \sqrt{\sigma_1} \ . \tag{40}$$

Proof. Fixed a value of $\tau \in \{1, \ldots, N\}$. From (5), (6) and (12), we obtain

$$\|\mathbf{\Sigma} - \mathbf{\Phi}_\tau\|_\infty = \|diag(0, \ldots, 0, \sigma_{\tau-1}, \ldots, \sigma_N)\|_\infty \ . \tag{41}$$

Also, as[3] $\|diag(0, \ldots, 0, \sigma_{\tau-1}, \ldots, \sigma_N)\|_\infty = \sigma_{\tau-1}$, from (35) and (41) it yields

$$E_{opt} = \frac{\sigma_{\tau-1}}{\sigma_1} \ . \tag{42}$$

Then, from (42), (36) and (39), we have

$$f(\sigma_\tau) = \frac{\sigma_{\tau-1}}{\sigma_1} + \frac{\sigma_1}{\sigma_\tau} = \frac{\sigma_\tau \sigma_{\tau-1} + \sigma_1}{\sigma_1 \sigma_\tau} \ . \tag{43}$$

From (6), for an $\epsilon \geq 0$, $\sigma_{\tau-1}$ can be written as

$$\sigma_{\tau-1} = \sigma_\tau - \epsilon \ . \tag{44}$$

---

[3] If **x** is a vector such that $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, then: $\|\mathbf{x}\|_\infty := \max(|x_1|, \ldots, |x_n|)$.

Then, from (43) and (44), $f(\sigma_\tau)$ becomes

$$f(\sigma_\tau) = \frac{\sigma_\tau^2 - \epsilon\sigma_\tau + \sigma_1}{\sigma_1\sigma_\tau} \, . \tag{45}$$

Considering the first derivative of $f(\sigma_\tau)$ in (45), we obtain

$$f'(\sigma_\tau) = \frac{\sigma_\tau^2 - \sigma_1}{\sigma_1\sigma_\tau^2} \tag{46}$$

and for $f'(\sigma_\tau) = 0$, the (40) follows. □

The equation (40), proved in Theorem 1, represents a condition to select the optimal truncation parameter as implemented in Algorithm 2.

In Section 6, we validate the results provided in this section. We also show that the employment of the TSVD method alleviates the computational cost making the running less expensive as it has an impact on the convergence of the L-BFGS method.

6. Testing

The VarDA model presented in the previous sections is applied to the pollutant dispersion within an urban environment. Hence, the VarDA model is coupled with Fluidity, an open-source, finite-element, fluid dynamic software (http://fluidityproject.github.io/). The basic Large Eddy Simulation (LES) equations describing the turbulent flows are based on the filtered incompressible Navier–Stokes equations (momentum equations and continuity of mass). The dispersion of the pollution is described by the classic advection–diffusion equation such that the concentration of the pollution is seen as a passive scalar. The equations are solved using second order schemes in time and space. Details of the equations solved and their implementations can be found in [9–11]. The state variable is such that:

$$\mathbf{u} = [p, v_x, v_y, v_z, C], \tag{47}$$

where p denotes the pressure, v_x and v_y denote the horizontal components of the velocity and v_z the vertical component of the velocity and C denotes the pollutant concentration. In this section, we validate the choice of the truncation parameter τ provided by Theorem 1. Starting from the spectrum of the background error deviance matrix $\mathbf{V}$, we evaluate the condition number $\mu(\mathbf{V}_\tau)$ provided in equation (36) and the Relative Error E_τ provided in equation (35) as function of τ to validate our hypothesis regarding the behaviour of these two functions. We use Algorithm 2 to compute the truncation parameter τ and we evaluate the choice of τ by evaluating the mean square error

$$MSE(\mathbf{u}) = \frac{\|\mathbf{u} - \mathbf{u}_C\|_{L^2}}{\|\mathbf{u}_C\|_{L^2}} \tag{48}$$

computed with respect to a control variable u_C; and the execution time to compute the solution of the DA problem by Algorithm 1.

6.1. Test cases

A 2D and 3D cases are presented in this paper. The 3D case is a realistic case, it includes 14 buildings representing a real urban area located in London South Bank University (LSBU) in Elephant and Castle, South London, UK (Fig. 1a). The mesh includes 100,040 nodes (Fig. 1b). The 2D case represents an idealised case used to test the ability of Fluidity to be coupled with the VarDA model as we know this does not affect the generality of our studies. The 2D case represents 3 buildings and the mesh includes 852 nodes (Fig. 2). Both meshes are unstructured.

The inlet boundary condition is a constant velocity equal to $v_x = 1$ m/s for the 2D case, while a log-profile velocity (representing the wind profile of the atmospheric boundary layer) is used in the 3D case. No-slip boundary conditions are applied on all building façades and the bottom surface of the domain. A free-slip boundary condition is applied at the top of the domain and on the sides of the domain for the 3D case. The outflow boundary condition is defined by a zero pressure $p = 0$. The background of pollution is set up as a sinusoidal function as expressed by equation (49):

$$C(t) = \frac{1}{2}\left(sin\left(\frac{2\pi t}{T}\right) + 1\right) \tag{49}$$

where C is the pollutant concentration, t is the time (in seconds) and T is the period (in seconds). Even if this background pollution is not based on real data, it mimics waves of pollution in an urban environment. Moreover, a point source of pollution is located in the middle of the domain of the 3D Case to mimic pollution generated by traffic in a busy intersection for example.
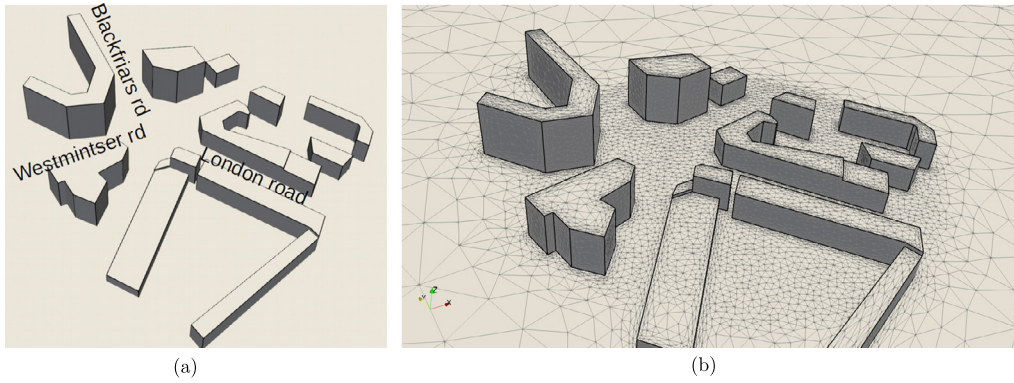
Fig. 1. (a) 3D Domain: London South Bank University (LSBU) and (b) unstructured mesh of the 3D Case.
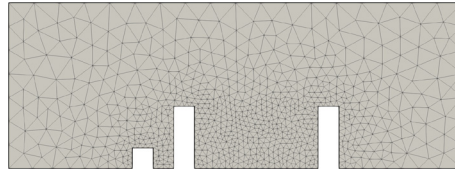

Fig. 2. Unstructured mesh of the 2D Case.
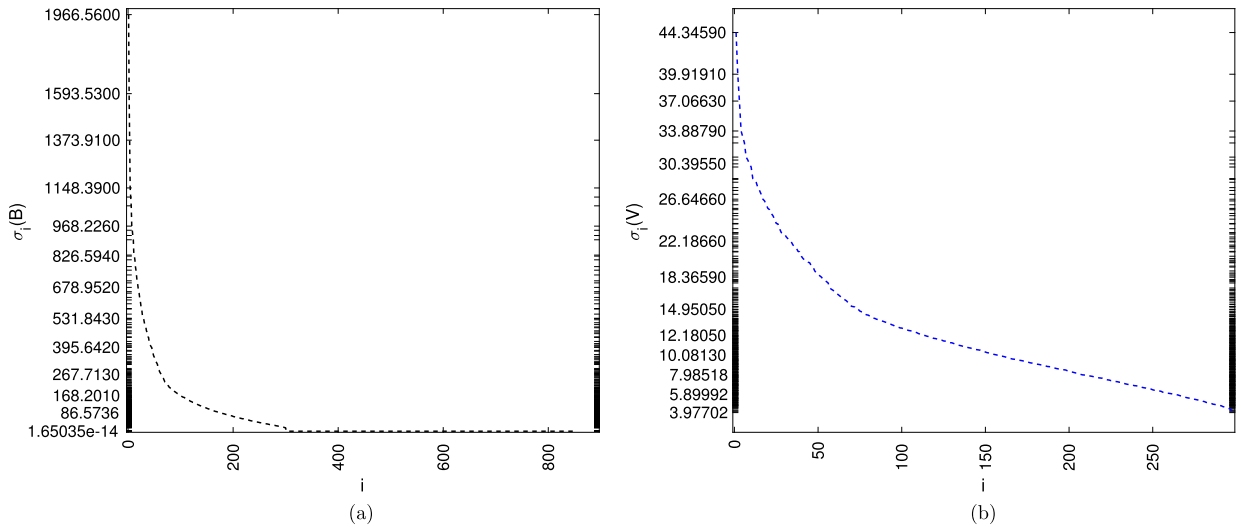
Fig. 3. (a) Spectrum of the background error covariance matrix **B** defined in equation (3) and (b) spectrum of the background error deviance matrix **V** defined in equation (2). Both computed for $M = 300$ temporal steps and $N = 852$ nodes for the 2D Case.

6.2. Conditioning of the VarDA problem and choice of the truncation parameter τ

Fig. 3b and Fig. 3a show the spectrum of the background error deviance matrix **V** defined in equation (2) and spectrum of the background error covariance matrix **B** defined in equation (3), computed for $M = 300$ temporal steps and $N = 852$ nodes for the 2D Case. Based on Fig. 3, the condition numbers, as defined in equation (11), are equal to $\mu(\mathbf{B}) = \mathcal{O}(10^{17})$ and $\mu(\mathbf{V}) = \mathcal{O}(10^1)$ respectively. As the condition number of the deviance and the covariance matrices affect the condition number of the VarDA problem [7,21], the VarDA problem is still ill-conditioned.

In the following, the TSVD method is used to improve the conditioning of the VarDA problem and Algorithm 2 is used to properly choose the truncation parameter τ. Fig. 4 shows the behaviour of the two functions E_τ and $\mu(\mathbf{V}_\tau)$ defined in equation (35) and equation (36) respectively. These figures confirm the theoretical evaluation provided in (38) and (37). For the 2D Case, using Algorithm 2, the optimal truncation parameter found is $\tau = 145$. This choice is confirmed by the post-processing study shown in Fig. 5a. The values of the MSE as defined in equation (48), computed with respect of a control variable, reduce up to $\tau = 145$ as expected. Fig. 5 highlights that the mean square error decreases with the increase of the truncation parameter, then reaches a plateau (Fig. 5a), while, as expected, the execution time rises linearly with the

(a) Condition Number TSVD

(b) Relative Error TSVD
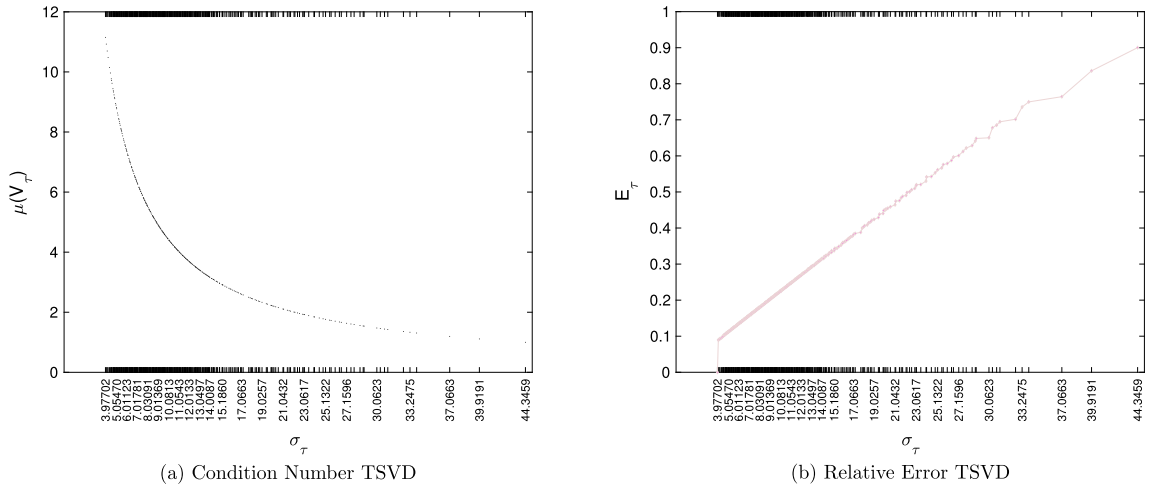
Fig. 4. Pre-processing for the choice of the truncation parameter $\tau \le M$ for the 2D Case with $M = 300$.

(a) Mean square error
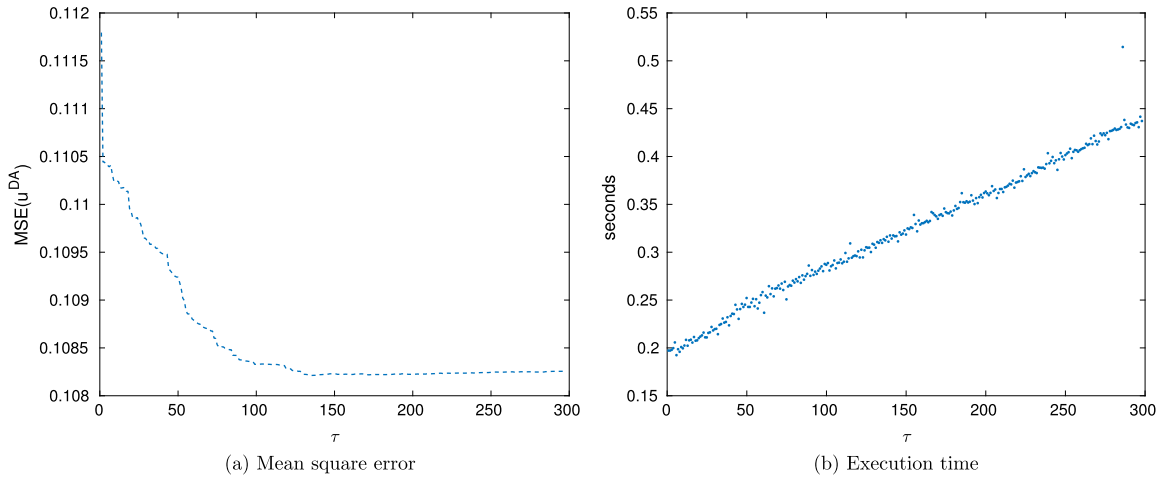
(b) Execution time

Fig. 5. Post-processing verification of the choice of the truncation parameter $1 \le \tau \le M$ for the 2D Case with $M = 300$: (a) Mean square error of the DA result as a function of the truncation parameter τ; (b) Execution time (in seconds) of Algorithm 1 as a function of the truncation parameter τ.

increase of the truncation parameter (Fig. 5b). Results show that the choice of the truncation parameter allows minimisation of the computational cost without significant loss in the solution accuracy.

Fig. 6 shows the values of $\delta \mathbf{u}^{DA}$ (see Step 15 of Algorithm 1) for different values of the truncation parameter τ. The state variables presented are C (i.e. the pollution concentration) in Fig. 6c, Fig. 6e and Fig. 6g and v (i.e. the velocity) in Fig. 6d, Fig. 6f and Fig. 6h. Values of the state variable simulating observations provided by sensors located on the roof of the three buildings were assimilated using $nobs = 6$ (Fig. 6a) observation points for C and $nobs = 45$ (Fig. 6b) observation points for v.

We have shown that the computational cost increases as the value of τ increases (Fig. 5b). On the other hand, for small values of τ, the numerical error propagates into the solution, impacting the model prediction, as it can be seen in Fig. 6c and Fig. 6d. Fig. 6e to Fig. 6h show a decreased value of the cost function close to the locations of the observed data. The optimum value of τ was used for Fig. 6e and Fig. 6f, indicating that the value of τ provided by Algorithm 2 allows the observed data to be assimilated, while also solving the problem in a reduced dimensional space, thereby reducing the computational cost as shown in Fig. 5b.

6.3. Errors evaluation

To evaluate the benefit provided by the assimilation of data using Algorithm 1 and Algorithm 2, we evaluate the error in data before and after the VarDA process computed with respect to a control variable $\mathbf{u}_C$.

In this section, for the 2D Case, τ is chosen equal to 145 using $M = 300$ and α is equal to 1. $nobs = 45$ observation points are used. Fig. 7 presents the absolute error before ($|\mathbf{u}_0 - \mathbf{u}_C|$) and after ($|\mathbf{u}^{DA} - \mathbf{u}_C|$) the VarDA process for three variables of state, i.e. the pressure p, the velocity v and the pollutant concentration C. Before the VarDA process, the error
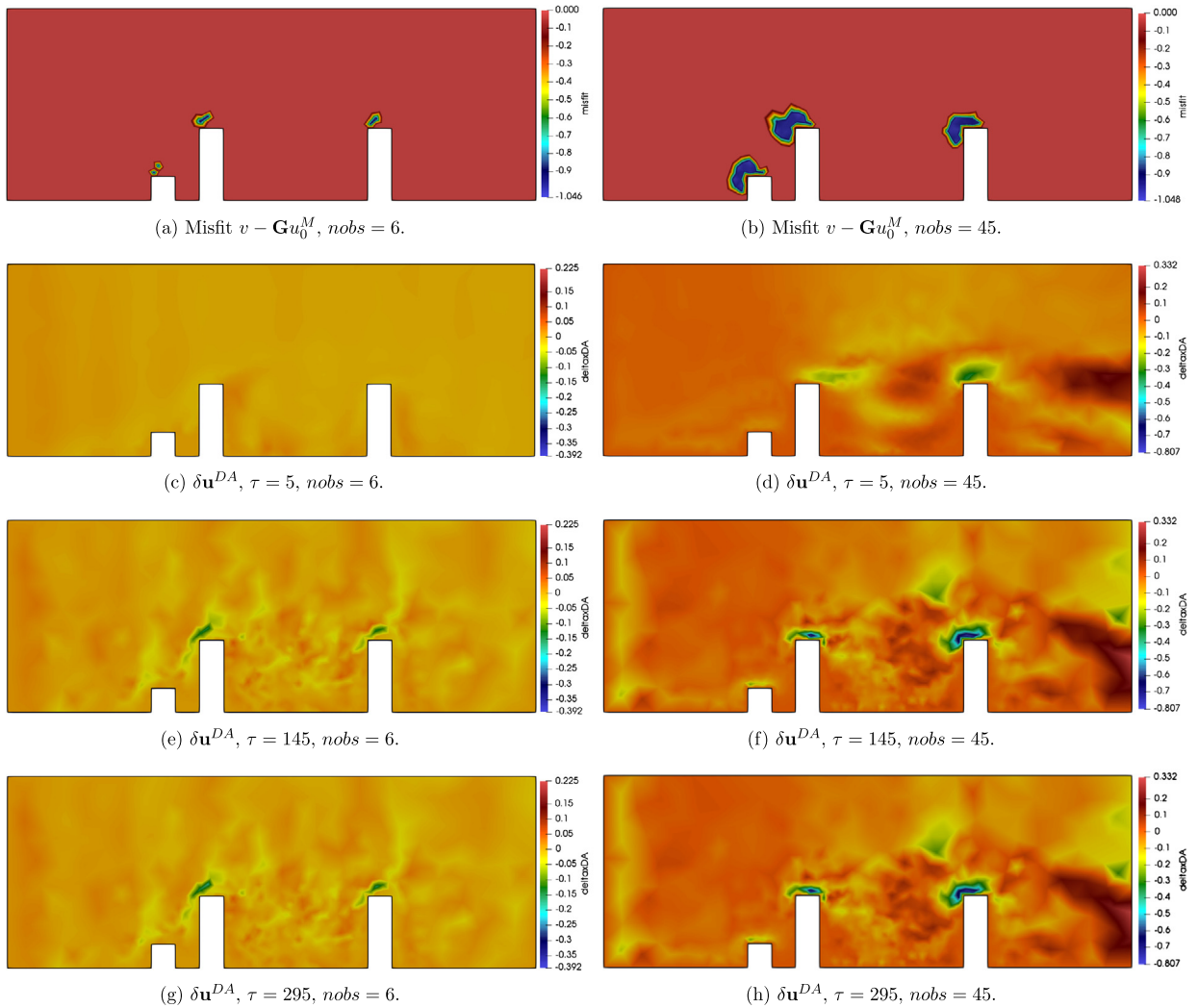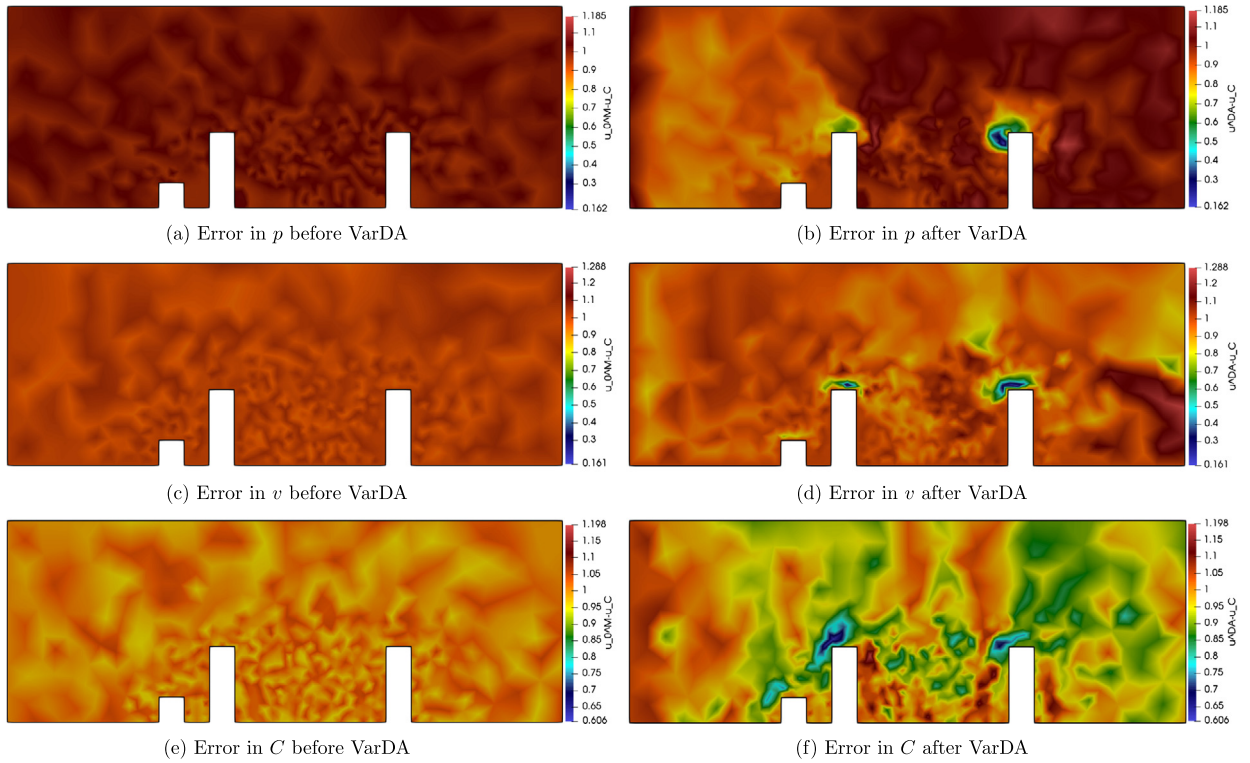
Fig. 6. Results of the misfit (see Step 3 in Algorithm 1) and $\delta \mathbf{u}^{DA}$ (see Step 15 in Algorithm 1) for different truncation numbers. Results are for the 2D Case and $\alpha = 1$. Figs. 6a, 6c, 6e and 6g are results for 6 observation points (2 observations for each roof) and the state variable is C, while Figs. 6b, 6d, 6f and 6h are results for 45 observation points (15 observations for each roof) and the state variables is v. The scale of Figs. 6c, 6e and 6g is between 0.225 (red colour) and -0.392 (blue colour), while the scale of Figs. 6d, 6f and 6h is between 0.332 (red colour) and -0.807 (blue colour). (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)

values are mostly between 0.9 and 1.185 for the pressure field (Fig. 7a), between 0.9 and 1.288 for the velocity field (Fig. 7c) and between 0.95 and 1.198 for the pollutant concentration field (Fig. 7e). The observation points are located at the top of the buildings and we can see that the error values at these locations can be reduced up to 0.162 for the pressure field (Fig. 7b), 0.161 for the velocity field (Fig. 7d) and 0.606 for the pollutant concentration (Fig. 7f). From Fig. 7, it can also be noticed that the error is, as expected, reduced at the observation point locations but also in the rest of the domain.

For the 3D case, the value of the truncation parameter provided by Algorithm 2 is $\tau = 51$ using $M = 105$ temporal steps. Moreover, α is chosen equal to 1. Fig. 8 and Fig. 9 show the absolute error before and after the VarDA process using 50,020 observation points for the pressure and the velocity fields respectively. For the pressure field, the error values around the buildings are mostly comprised between 0.15 and 0.25 before the VarDA process (Fig. 8a), while they are successfully reduced to values between 0 (not included) and 0.15 after the VarDA process (Fig. 8b). Similar observation can be made for the velocity field: the error values are between 0.3 and 0.5 before the VarDA process (Fig. 9a) and between 0 (not included) and 0.3 after the VarDA process (Fig. 9b). From Fig. 8 and Fig. 9, it can be observed that the error after the VarDA process can be divided by a factor up to 1.6.

Fig. 10 shows the pollutant concentration, along a horizontal slice at 10 m above the ground, generated by a point source of pollution located in the centre of the domain. Fig. 10a shows the results predicted by the forecasting model Fluidity, i.e. $\mathbf{u}_0$, while Fig. 10b shows the observed data, i.e. $\mathbf{v}$. Values $\mathbf{v}$ are assimilated to correct the forecasting data $\mathbf{u}_0$. The assimilated data after the VarDA process, i.e. $\mathbf{u}^{DA}$, are then obtained (Fig. 10c). From Fig. 10, it can be seen that the

(a) Error in p before VarDA

(b) Error in p after VarDA

(c) Error in v before VarDA

(d) Error in v after VarDA

(e) Error in C before VarDA

(f) Error in C after VarDA

Fig. 7. Comparison of the absolute error before VarDA (values of $|\mathbf{u}_0 - \mathbf{u}_C|$) and after VarDA (values of $|\mathbf{u}^{DA} - \mathbf{u}_C|$) computed with respect of a control variable $\mathbf{u}_C$ for the pressure p, velocity v and pollutant concentration C fields produced by Algorithm 1 and Algorithm 2 assimilating $nobs = 45$ data from sensors located on the top of the three buildings. Scale for p (Fig. 7a and Fig. 7b) is between 0.162 (blue colour) and 1.185 (red colour). Scale for v (Fig. 7c and Fig. 7d) is between 0.161 (blue colour) and 1.28 (red colour). Scale for C (Fig. 7e and Fig. 7f) is between 0.606 (blue colour) and 1.198 (red colour). Results for the 2D Case using $\tau = 145$, $M = 300$ and $\alpha = 1.0$.

(a) $|\mathbf{u}_0 - \mathbf{u}_C|$
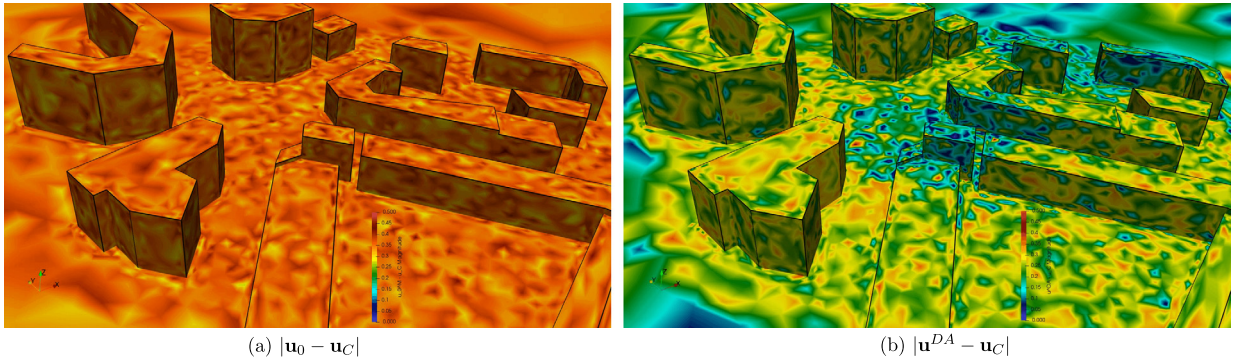
(b) $|\mathbf{u}^{DA} - \mathbf{u}_C|$
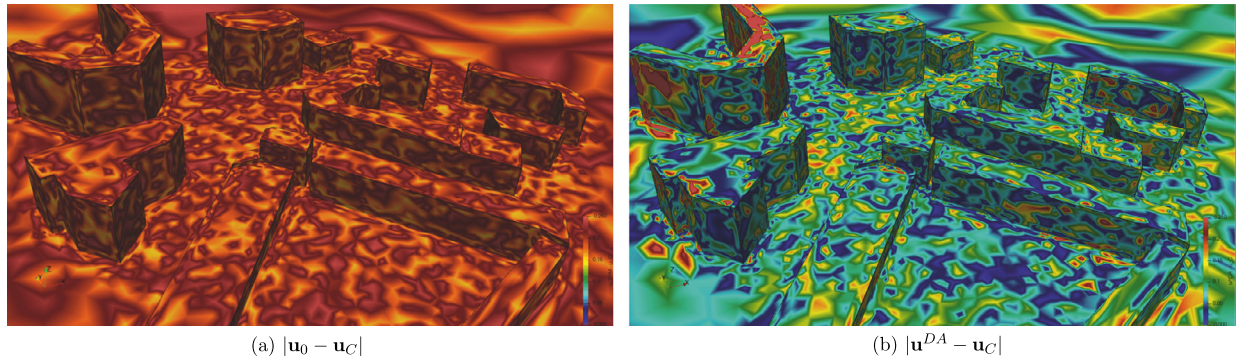
Fig. 8. Absolute error (a) before and (b) after Data Assimilation for the pressure field produced by Algorithm 1 and Algorithm 2 using $\tau = 51$, $M = 105$, $nobs = 50,020$ and $\alpha = 1$. In both figures, the scale is between 0 (blue colour) and 0.25 (red colour).
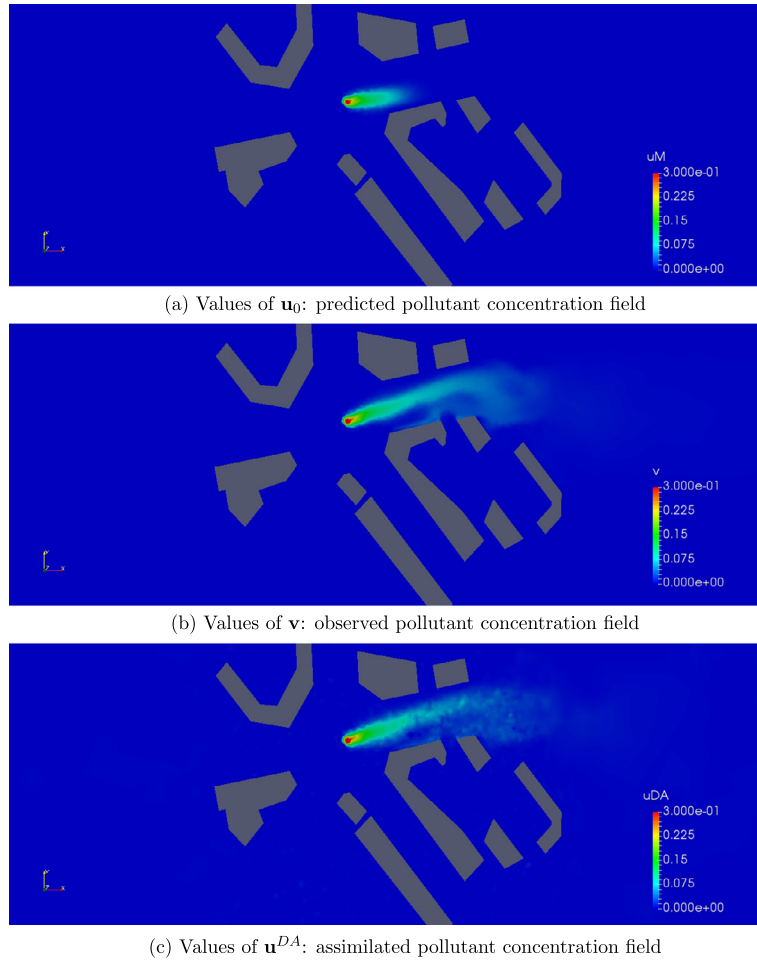
observed data are well-assimilated: the assimilated pollutant concentration after the VarDA process is much closer to the real observed data than the one predicted by Fluidity. Indeed, the spread of the pollution towards the right of the domain was under-predicted by Fluidity compared to observation data. The VarDA process proposed in this paper allows to improve the numerical forecasted results. This observation was expected and is consistent with the fact that the absolute error values are considerably reduced after the VarDA process compared to the values before.

6.4. Regularisation parameter α and comparison of execution time between L-BFGS and CG methods

Based on the analysis in Section 6.2, the truncation number τ is here chosen to be equal to 145 for the 2D Case. In this section, MSE is computed as described in equation (48): the error before the VarDA process is computed with $\mathbf{u} = \mathbf{u}_0$, while the error after the VarDA process is computed with $\mathbf{u} = \mathbf{u}^{DA}$. Fig. 11a shows the variation of MSE as a function of the number of assimilated observation points $nobs$ (with, $0 \leq nobs \leq 852$) for several values of regularisation parameter α.

(a) $|\mathbf{u}_0 - \mathbf{u}_C|$　　　　　(b) $|\mathbf{u}^{DA} - \mathbf{u}_C|$

Fig. 9. Absolute error (a) before and (b) after Data Assimilation for the velocity field produced by Algorithm 1 and Algorithm 2 using $\tau = 51$, $M = 105$, $nobs = 50{,}020$ and $\alpha = 1$. In both figures, the scale is between 0 (blue colour) and 0.5 (red colour).

(a) Values of $\mathbf{u}_0$: predicted pollutant concentration field

(b) Values of $\mathbf{v}$: observed pollutant concentration field

(c) Values of $\mathbf{u}^{DA}$: assimilated pollutant concentration field

Fig. 10. Horizontal slice of the 3D Case, at 10 m above the ground, showing the pollutant concentration field generated by a point source of pollution located in the centre of the domain. The scale is between 0 (blue colour highlighting region with no pollution) and 0.3 (red colour highlighting region with high concentration of pollutant). Results using $\tau = 51$, $M = 105$ and $\alpha = 1$.

The regularisation parameter α varies between 0.1 and 10: values lower than 1.0 means that the observation is considered more important than the background state, while values higher than 1.0 indicates that the observation has less importance than the background state. In other words, $\alpha < 1$ can be used if the accuracy of sensors are relatively precise, while $\alpha > 1$ can be seen as sensors having low reliability for example.

In Fig. 11a we can observe that the error decreases up to one order of magnitude using our VarDA algorithm for values of α in equation (31) such that $0.1 \leq \alpha \leq 10$. Moreover, as expected, the error decreases as the value of α decreases, i.e.
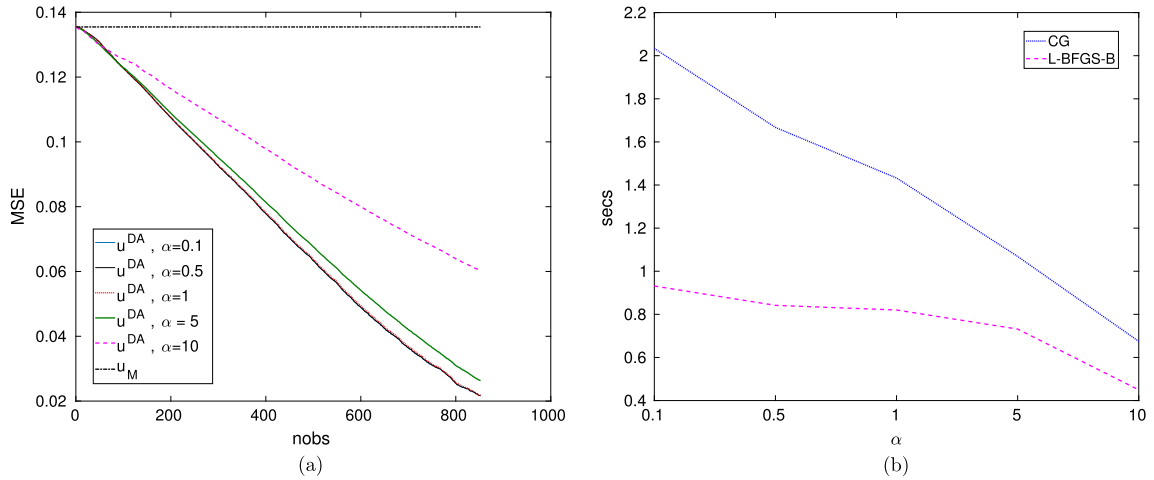
Fig. 11. Results for the 2D Case using $\tau = 145$ and $M = 300$. (a) Values of the Mean Square Error (equation (48)) for the pressure field as a function of the number of observation points *nobs* using Algorithm 1 for several values of regularisation parameter α. (b) Execution time of Algorithm 1 for minimisation achieved by the L-BFGS method (Step 8 – Step 14) or by replacing the L-BGFS with the Preconditioned Conjugate Gradient (PCG) method.

when more importance is given to the observed data obtained from reliable sensors. Indeed, for $\alpha = 10$ the error after DA is larger. Hence, the choice of α depends on the required accuracy and is a user-defined parameter.

Fig. 11b shows the execution time of Algorithm 1 as a function of the regularisation parameter α using the L-BFGS method or the PCG method [12]. Whatever the minimisation method used (L-BFGS or PCG), the execution time decreases when α increases. Indeed, given more importance to the observed data implies that the cost function required more iterations, i.e. more time, to be minimised. The execution time decreases almost linearly for the PCG method, while the L-BFGS method's execution time seems to be less impacted by the choice of α (for $\alpha < 10$) making the L-BFGS method more reliable. As shown in Fig. 11a, the error is the largest for $\alpha = 10$, however the gain provided by the Algorithm 1 with $\alpha = 10$ is in terms of execution time which decreases as shown in Fig. 11b. The execution time is divided by around 2 for $\alpha = 0.1$, by around 1.7 for $\alpha = 1$ and by around 1.5 for $\alpha = 10$ when the L-BFGS method is used instead of the PCG method, making the L-BFGS method more appropriate for real-time data assimilation utilisation.

6.5. Impact of VarDA in fluidity

Fig. 12a shows the MSE of the pollutant concentration field as defined in Section 6.4 before running Fluidity at time $t = t_0$. Fig. 12b shows the values of the MSE for the pollution concentration computed with respect of a control variable at time $t = t_1$, i.e. $\mathbf{u}_C(t_1)$, after running Fluidity. For

$$\mathbf{u}_1 = \mathbf{M}_{0,1}(\mathbf{u}_0^{DA}), \tag{50}$$

with $\mathbf{M}_{0,1}$ as defined in (19) for $k = 1$, we have

$$MSE = \frac{\|\mathbf{M}_{0,1}(\mathbf{u}_0^{DA}) - \mathbf{u}_C(t_1)\|_{L^2}}{\|\mathbf{u}_C(t_1)\|_{L^2}}. \tag{51}$$

Fig. 12b shows that the error after running Fluidity decreases as well and, in particular, the error magnitude decreases as the number of observed data increases with the same trend of the error magnitude before running Fluidity (see Fig. 12a).

As expected, results shown in Fig. 12b confirm the impact of Data Assimilation on the accuracy of the prediction produced by Fluidity. In fact, let $\mathbf{u}_1$ and $\mathbf{u}_{1(DA)}$ (defined in (50)) be the solution of the model at time $t = t_1$ with initial condition u_0 and $\mathbf{u}_0^{DA}$ respectively, we can assume

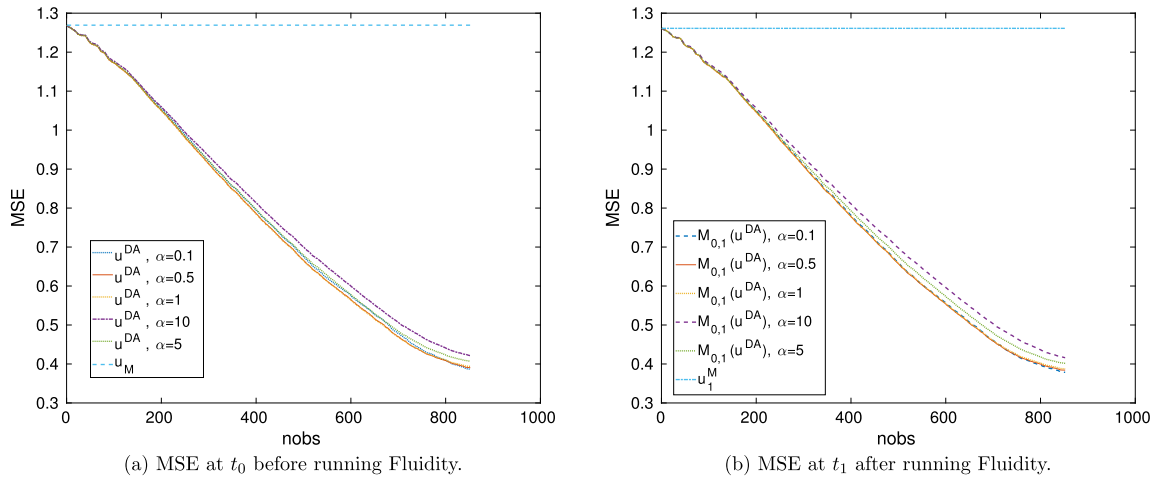$$\mathbf{u}_1 = \tilde{\mathbf{u}}_1 + \mathbf{e}_1$$

and

$$\mathbf{u}_{1(DA)} = \tilde{\mathbf{u}}_{1(DA)} + \mathbf{e}_{1(DA)}$$

where $\tilde{u}$ denotes the exact solution and $\mathbf{e}_1$ and $\mathbf{e}_{1(DA)}$ denote the numerical errors in the solution.

It is proved that [21]:

$$\mathbf{e}_1 \propto C_{\mathcal{A}_\mathbf{M}} \mathbf{e}_0 \tag{52}$$

(a) MSE at t_0 before running Fluidity.

(b) MSE at t_1 after running Fluidity.

Fig. 12. Values of MSE for the pollution concentration field as function of *nobs* assimilated observations and for values of $0.5 \leq \alpha \leq 10$ at time (a) $t = t_0$, i.e. before running Fluidity and (b) $t = t_1$, i.e. after running Fluidity. Results for the 2D Case using $\tau = 145$ and $M = 300$.

Table 1
Comparison of the CPU times of Algorithm 1 and Algorithm 2 and of Steps 8–14 of Algorithm 1 for values of the truncation parameter $\tau = 51$ and $\tau = 105$. Results for the 3D Case and $\alpha = 1$.

τ	105	51	Gain
$\mathcal{A}_1 + \mathcal{A}_2$	2.1e + 03 s	8.1e + 02 s	2.6
Steps 8–14 of $\mathcal{A}_1$	0.44 s	0.21 s	2.1

and

$$\mathbf{e}_{1(DA)} \propto C_{\mathcal{A}_{\mathbf{M}}} \mathbf{e}_0^{DA} \tag{53}$$

where $C_{\mathcal{A}_{\mathbf{M}}}$ is a constant which depends on the condition number of the numerical model and on the Algorithm implemented in Fluidity. As [21]:

$$\mathbf{e}_0^{DA} \leq \mathbf{e}_0, \tag{54}$$

from (52) and (53) and (54), we obtain

$$\mathbf{e}_{1(DA)} \leq \mathbf{e}_1 \tag{55}$$

as confirmed by results shown by results in Fig. 12.

6.6. CPU time

Fig. 13 shows the CPU time of Algorithm 1 and Algorithm 2 for the 3D Domain (shown in Fig. 1) as the function of the truncation parameter τ up to $\tau = 105$, i.e. the total number of time steps used in the simulation. Note that Fig. 13a shows the execution time of Algorithm 1 which uses Algorithm 2 to compute the preconditioned background covariance matrix. Fig. 13b shows the execution time for computing the minimum of the cost function by the L-BFGS method, i.e. Steps 8–14 in Algorithm 1. Particular results for $\tau = 51$, i.e. the optimal truncation parameter computed by Algorithm 2, and for $\tau = 105$, i.e. the number of time steps M used for the 3D simulation, are reported in Table 1 where the combination of the two algorithms is denoted by $\mathcal{A}_1 + \mathcal{A}_2$. The last column of Table 1 shows the gain computed as the ratio of the execution times for $\tau = 105$ and for $\tau = 51$.

7. Conclusions and future work

In this paper, we have presented a regularised and preconditioned Variational Data Assimilation (VarDA) method to assimilate data from sensors into the open-source, finite-element, fluid dynamics model Fluidity. The efficiency and the accuracy of our model were discussed and proved using a 2D and a 3D case of air flows and pollution transport in urban environments.

Our VarDA model is defined in a reduced optimal space obtained by the Truncated Singular Value Decomposition (TSVD) of the background error covariance matrix. The truncation parameter is chosen to minimise both the condition number of
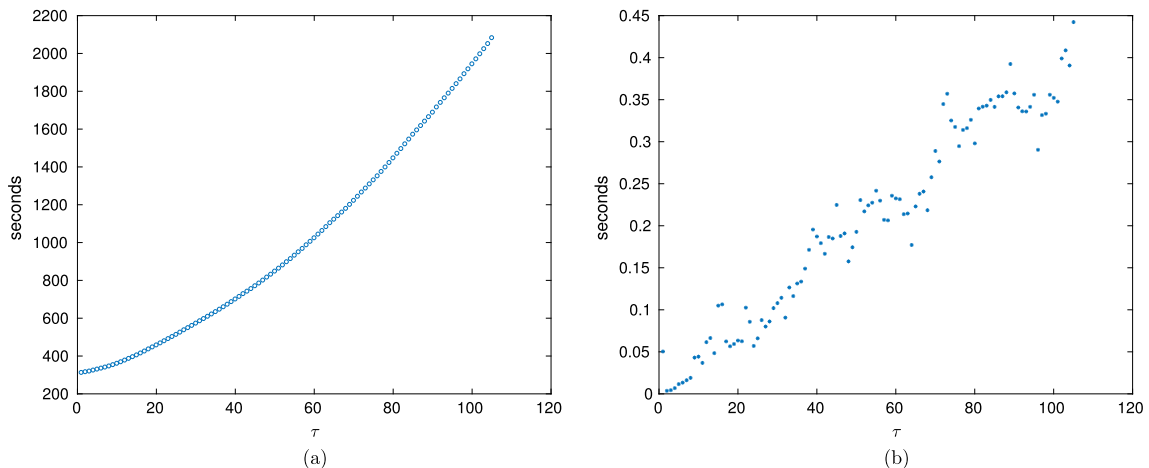
Fig. 13. CPU time of (a) Algorithm 1 and Algorithm 2 and (b) Steps 8–14 in Algorithm 1 as the function of the truncation parameter τ. Results for the 3D Case and $\alpha = 1$.

the numerical problem and the regularisation error. An algorithm, based on Theorem 1, to compute the optimal truncation parameter, independently of the knowledge of an exact solution, is provided. It has been shown that the optimal truncation parameter obtained with the proposed method is reliable and alleviates considerably the computational time.

We have also shown that our model improves the prediction of air flows and pollution transport in an urban environment. The state variable consists of values of pressure and velocities of the air flows and pollution concentration. It has been highlighted that the observed data are well-assimilated decreasing the error at the sensors location and also impacting the error in the rest of the domain. The error after the VarDA process can be reduced up to one order of magnitude compared to before the VarDA process. Finally, it has been proved that our VarDA model reduces the error in the forecasting model and this reduction in error propagates in the next time step when Fluidity is running.

A user-defined regularisation parameter was introduced in the cost function allowing flexibility to the user to give more importance, if desired, to the observed data compared to the background state. The impact of the regularisation parameter on the error and the execution time was discussed. The more the observation data are important, the longer the time execution is.

The efficiency of the L-BFGS method, compared to the PCG method, was discussed, highlighting that the L-BFGS method is appropriate for real-time utilisation.

Finally, the VarDA model proposed in this paper was used, for the first time, for air flows and pollution transport in urban environments. The algorithms and the method proposed are, however, enough generic and not intrinsically link to the software and then can be used easily for other physical problems.

Acknowledgement

References

[1] E. Kalnay, Atmospheric Modeling, Data Assimilation and Predictability, Cambridge University Press, 2003.
[2] R. Kalman, A new approach to linear filtering and prediction problems, Trans. ASME Ser. D, J. Basic Eng. 82 (1960) 35–45.
[3] E. Andersson, J. Haseler, P. Undén, P. Courtier, G. Kelly, D. Vasiljevic, C. Brancovic, C. Cardinali, C. Gaffard, A. Hollingsworth, C. Jakob, P. Janssen, E. Klinker, A. Lanzinger, M. Miller, F. Rabier, A. Simmons, B. Strauss, J.-N. Thepaut, P. Viterbo, The ecmwf implementation of three dimensional variational assimilation (3dvar). Part III: experimental results, Q. J. R. Meteorol. Soc. 124 (1998) 1831–1860.
[4] D.M. Baker, W. Huang, Y. Guo, J. Bourgeois, Q. Xiao, Three-dimensional variational data assimilation system for mm5: implementation and initial results, Mon. Weather Rev. 132 (2004) 897–914.
[5] A.W. Kristian Mogensen, Magdalena Alonso Balmaseda, The NEMOVAR Ocean Data Assimilation System as Implemented in the ECMWF Ocean Analysis for System 4, CERFACS, Toulouse, 2012.
[6] A.M. Moore, H.G. Arango, G. Broquet, B.S. Powell, A.T. Weaver, J. Zavala-Garay, The regional ocean modeling system (roms) 4-dimensional variational data assimilation systems: part I – system overview and formulation, Prog. Oceanogr. 91 (2011) 34–49.
[7] J. Nichols, Mathematical concepts in data assimilation, in: W. Lahoz, et al. (Eds.), Data Assimilation, 2010.
[8] J.A. Haben, Conditioning and Preconditioning of the Minimisation Problem in Variational Data Assimilation, PhD Thesis, 2011.
[9] R. Ford, C.C. Pain, A.J.H. Goddard, C.R.E. De Oliveira, A.P. Umpleby, A non-hydrostatic finite-element model for three-dimensional stratified oceanic flows. Part I: model formulation, Mon. Weather Rev. 132 (2004) 2816–2831.
[10] E. Aristodemou, T. Bentham, C. Pain, A. Robins, A comparison of mesh-adaptive les with wind tunnel data for flow past buildings: mean flows and velocity fluctuations, Atmos. Environ. 43 (2009) 6238–6253.
[11] Imperial College London AMCG, Fluidity manual v4.1.12, 2015.
[12] G.H. Golub, C.F.V. Loan, Matrix Computations, JHU Press, 1996.

[13] S. Grattona, J. Tshimanga, An observation-space formulation of variational assimilation using a restricted preconditioned conjugate gradient algorithm, Q. J. R. Meteorol. Soc. 135 (2009) 1573–1585.
[14] S. Gurol, A.T. Weaver, A.M. Moore, A. Piacentini, H.G. Arangod, S. Gratton, B-preconditioned minimization algorithms for variational data assimilation with the dual formulation, Q. J. R. Meteorol. Soc. 140 (2014) 539–556.
[15] D. Liu, J. Nocedal, On the limited memory bfgs method for large scale optimization, Math. Program. 45 (1989) 503–528.
[16] I.M. Navon, D.M. Legler, Conjugate-gradient methods for large-scale minimization in meteorology, Mon. Weather Rev. 115 (4) (1987) 1479–1502.
[17] D. Daescu, I.M. Navon, An analysis of a hybrid optimization method for variational data assimilation, Int. J. Comput. Fluid Dyn. 17 (4) (2003) 299–306.
[18] X. Zou, I.M. Navon, M. Berger, K. Phua, T. Schlick, F.L. Dimet, Numerical experience with limited memory quasi-Newton and truncated Newton methods, SIAM J. Optim. 3 (3) (1993) 582–608.
[19] A.K. Alekseev, I.M. Navon, J. Steward, Comparison of advanced large-scale minimization algorithms for the solution of inverse ill-posed problems, Optim. Methods Softw. 24 (1) (2009) 63–87.
[20] J. Nocedal, R. Byrd, P. Lu, C. Zhu, L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization, ACM Trans. Math. Softw. 23 (1997) 550–560.
[21] R. Arcucci, L. D'Amore, J. Pistoia, R. Toumi, A. Murli, On the variational data assimilation problem solving and sensitivity analysis, J. Comput. Phys. 335 (2017) 311–326.
[22] D. Cacuci, I.M. Navon, M. Ionescu-Bujor, Computational Methods for Data Evaluation and Assimilation, CRC Press, 2013.
[23] C. Hansen, Rank-Deficient and Discrete Ill-Posed Problems, Numerical Aspects of Linear Inversion, SIAM, 1998.
[24] R. Arcucci, L. Carracciuolo, R. Toumi, Toward a preconditioned scalable 3dvar for assimilating sea surface temperature collected into the Caspian sea, J. Numer. Anal. Ind. Appl. Math. 12 (1–2) (2018) 9–28.
[25] R. Arcucci, L. D'Amore, L. Carracciuolo, G. Scotti, G. Laccetti, A decomposition of the Tikhonov regularization functional oriented to exploit hybrid multilevel parallelism, Int. J. Parallel Program. 45 (5) (2017) 1214–1235.
[26] A. Hannachi, A Primer for EOF Analysis of Climate Data, Department of Meteorology, University of Reading, UK, 2004.
[27] R. Arcucci, C. Pain, Y. Guo, Effective variational data assimilation in air-pollution prediction, Big Data Mining Anal. 1 (4) (2018) 297–307.
[28] P.C. Hansen, J.G. Nagy, D.P. O'Leary, Deblurring Images: Matrices, Spectra, and Filtering, SIAM, 2006.
[29] J. Courtier, A strategy for operational implementation of 4d-var, using an incremental approach, Q. J. R. Meteorol. Soc. 120 (1994) 1367–1387.
[30] H.K. Engl, M. Hanke, A. Neubauer, Regularization of Inverse Problems, Kluwer, 1996.
[31] A. Lorenc, Development of an operational variational assimilation scheme, J. Meteorol. Soc. Jpn. 75 (1997) 339–346.
[32] D. Cacuci, M. Ionescu-Bujor, I.M. Navon, Sensitivity and Uncertainty Analysis, vol. II: Applications to Large Scale Systems, Chapman and Hall, Boca Raton, Florida, 2005.