Hybrid Data Assimilation: an Ensemble-Variational Approach

Edward M. Lim
Data Science Institute,
Department of Computing,
Imperial College London (UK)
eyl215@ic.ac.uk

Miguel Molina-Solana
Dept. Computer Science and AI
Universidad de Granada
Granada, Spain
miguelmolina@ugr.es

Christopher Pain
Department of Earth Science & Engineering
and Data Science Institute,
Imperial College London (UK)
c.pain@imperial.ac.uk

Yi-Ke Guo
Data Science Institute,
Department of Computing,
Imperial College London (UK)
y.guo@imperial.ac.uk

Rossella Arcucci
Data Science Institute,
Department of Computing,
Imperial College London (UK)
r.arcucci@imperial.ac.uk

***Abstract*—Data Assimilation (DA) is a technique used to quantify and manage uncertainty in numerical models by incorporating observations into the model. Variational Data Assimilation (VarDA) accomplishes this by minimising a cost function which weighs the errors in both the numerical results and the observations. However, large-scale domains pose issues with the optimisation and execution of the DA model. In this paper, ensemble methods are explored as a means of sampling the background error at a reduced rank to condition the problem. The impact of ensemble size on the error is evaluated and benchmarked against other preconditioning methods explored in previous work such as using truncated singular value decomposition (TSVD). Localisation is also investigated as a form of reducing the long-range spurious errors in the background error covariance matrix. Both the mean squared error (MSE) and execution time are used as measure of performance. Experimental results for a 3D case for pollutant dispersion within an urban environment are presented with promise for future work using dynamic ensembles and 4D state vectors.**

***Index Terms*—Data Assimilation, Ensemble, Variational approach**

I. Introduction

Air pollution is the cause of premature deaths daily, thus necessitating the development of more reliable and accurate numerical tools [1]. Furthermore, global warming and deteriorating outdoor air quality has resulted in excessive energy consumption for cooling, air-conditioning and burning of fossil fuels [2]. As a result, emissions of greenhouse gases, pollutants and heat have produced heat islands, air pollution and unhealthy air conditions particularly in urban areas. One of the methods of mitigating this is with smart urban planning such that the natural ventilation of buildings can provide a sustainable way to cool indoor environments, manage building energy consumption and control the flow of pollutants in the air. Thus, the necessity to develop an advanced computational system to forecast the airflow and air quality for an urban area, informing the urban planning process. The goal is on utilising these numerical airflow models to optimise the natural ventilation in buildings, reduce greenhouse gas emission and conserve energy usage.

To accomplish this, an efficient and practical model for predicting airflow at various scales has to be developed. Numerical simulations are widely used to predict and model the complex behaviour of fluid flows from scales of small tunnels and rooms to entire cities [3]. In these complex and chaotic systems, the initial conditions, the background state and the governing equations are often incomplete without an accurate closed form [4] resulting in the introduction of uncertainty and error in calculations. This fact, coupled with the accumulated errors introduced from discretisation and finite precision of numerical simulations, exacerbates the validity and accuracy of these models [5].

Data Assimilation (DA) techniques are widely used in this field to develop forecasting models with higher confidence, taking into account the various errors that arise in numerical computation. Data Assimilation is a powerful method of uncertainty quantification used to integrate observations into a prediction model to improve the forecast. It is in fact the de-facto method used for state-of-the-art Numerical Weather Prediction (NWP) models [6], [7]. Most of these NWP models however operate within a 2D environment.

Variational Data Assimilation (VarDA) is based on the minimisation of a cost function that accounts for the errors in the observations and forecasts that assimilates future observations [8], [9]. For our purposes, the VarDA model has to be operational at nearly real-time. Previous work utilised a truncated singular value decomposition (TSVD) method of preconditioning that yielded significant results. While TSVD is an effective method of reducing the rank of the background error covariance matrix, it comes with computational overhead and is limited by its truncation parameter τ [10]. This manuscript extends the data assimilation (DA) methods used in [11] by means of ensemble extensions of them.

Ensemble-Variational data assimilation is a hybrid method

978-1-7281-5686-6/19/$31.00 ©2019 IEEE
DOI 10.1109/SITIS.2019.00104

which avoids building this background error covariance directly, instead using an ensemble of possible forecasts to reduce the rank and propagate the background-error statistics [12]. The objective is thus to investigate and validate the use of ensemble methods with 3D variational data assimilation in a 3D environment at a large scale. We verify the performance of different ensemble sizes to determine the best number of ensemble members by comparing the mean squared errors. The impact of localisation is also investigated —localisation has been previously explored in 2D environments [13] but here we explore localisation in 3D. The execution times of each method are also weighed to determine their suitability for this application.

This work covers the formulation of the 3D Ensemble Variational Data Assimilation (3DEnVar) method and its performance on a dataset of pollutant concentration and velocity profiles collected in the Borough of Southwark, London provided by London South Bank University (LSBU) and a computational fluid dynamics model named Fluidity (available at http://fluidityproject.github.io) [2]. The results are benchmarked against previous works (using the TSVD method of preconditioning) in [11]. This is to serve as a proof of concept for future works in applying ensemble data assimilation or hybrid methods with fully dynamic background error covariance matrices in a 3D environment. The methods described here are general enough to be applied to any other data assimilation problem.

This work is organized as follows: Section II provides mathematical settings and the main steps of the Data Assimilation (DA) model and the DA algorithm. Section III introduces the Ensemble methods and the procedure to build ensemble. Experimental results on realistic test cases are provided in Section IV. The conclusions of this work and the description of future work are in Section V.

II. DA Model

A. The General DA Model

Let the description of the forecasting model be

$$\mathbf{x}_{k+1} = \boldsymbol{\mathcal{M}}_{k+1}\mathbf{x}_k \tag{1}$$

where $\mathbf{x}_k$ and $\boldsymbol{\mathcal{M}}_{\boldsymbol{k}}$ are respectively the state vector and nonlinear model operator at time k. In addition, let $\mathbf{y}_k^o$ be the vector of observations at time k and $\boldsymbol{\mathcal{H}}_k$ be the nonlinear observation operator that maps the model space to the observation space:

$$\mathbf{y}_k^x = \boldsymbol{\mathcal{H}}_k\mathbf{x}_k \tag{2}$$

The formulation of the VarDA equation requires knowledge of the Background Error covariance matrix, **B** and observation error covariance matrix, **R** [14]. The VarDA cost function is essentially a form of Tikhonov regularisation [15] and is defined as

$$\begin{aligned} J(\mathbf{x}) = &\frac{1}{2}\left(\mathbf{x}-\mathbf{x}^b\right)^T\mathbf{B}^{-1}\left(\mathbf{x}-\mathbf{x}^b\right) \\ &+\frac{1}{2}\left(\mathbf{H}\mathbf{x}-\mathbf{y}^o\right)^T\mathbf{R}^{-1}\left(\mathbf{H}\mathbf{x}-\mathbf{y}^o\right) \end{aligned} \tag{3}$$

where **H** is the linearised form of the nonlinear observation operator $\boldsymbol{\mathcal{H}}$ and $\mathbf{x}^b$ is the background state vector.

Equation (3) can be linearised about the background state vector which will result in the incremental form of the cost function given by

$$J(\delta\mathbf{x}) = \frac{1}{2}(\delta\mathbf{x})^T\mathbf{B}^{-1}(\delta\mathbf{x}) + \frac{1}{2}\sum_{k=0}^{K}(\delta\mathbf{d}_k)^T\mathbf{R}_k^{-1}(\delta\mathbf{d}_k) \tag{4}$$

where where $\delta\mathbf{d}_k$ = $\mathbf{d}_k - \mathbf{H}_k\delta\mathbf{x}$ and $\mathbf{d}_k = \mathbf{y}_k^o - \mathbf{H}_k\mathbf{x}_k^b$. This cost function is minimised to provide a solution,

$$\delta\mathbf{x}^a = argmin\, J(\delta\mathbf{x}) \tag{5}$$

and

$$\mathbf{x}^a = \mathbf{x}^b + \delta\mathbf{x}^a \tag{6}$$

with $\mathbf{x}^a$ denoting the state vector after data assimilation.

This form is called 3D First Guess at Appropriate Time or 3D-FGAT [16] and is useful applications where the model operator, $\boldsymbol{\mathcal{M}}$ is not available.

B. Reduced VarDA Model

The high dimensionality of the background error covariance matrix **B**, which is of size $n \times n$ (where n is the number of features in the state vector $\mathbf{x}_k$), is an issue. Hence, methods have been devised to effectively reduce the problem space. One of the most popular methods for reduction factorises **B** as it is often required in variational DA. It also exploits the sparse nature and symmetry of the matrix.

$$\mathbf{B} = \mathbf{V}\mathbf{V}^T \tag{7}$$

where **V** is the deviation matrix defined as

$$\mathbf{V} = \underline{\mathbf{x}} - E[\underline{\mathbf{x}}]$$

and $E[\underline{\mathbf{x}}]$ is the *expected* value or *mean* of $\underline{\mathbf{x}}$. In this case, $\underline{\mathbf{x}}$ is an $n \times m$ matrix of states where n is the number of state features and m is the size of the assimilation window. $\underline{\mathbf{x}}$ is basically all the state vectors at time k, $\mathbf{x}_k$ stacked together into a matrix. For 3D-Var, this simplifies to

$$\mathbf{V} = \underline{\mathbf{x}}^b - E[\underline{\mathbf{x}}^b]$$

as there is no model operator (taken as identity). **V** is called the *background perturbation matrix*. There are different forms of background error covariance factorisation, but this method is the most prevalent [14].

A *control variable transform* (CVT) is introduced to reformulate the cost function. Instead of dealing with (4) which is a function of $\delta\mathbf{x}$, the idea is to reformulate the cost function to a new control variable which avoids the explicit knowledge of **B**. There are several choices for the control variable [17], but the choice of control variable should replace the use of **B** with **V** in the cost function. Hence, the following transformation is made:

$$\delta\mathbf{x} = \mathbf{V}\delta\boldsymbol{\mathcal{X}} \tag{8}$$

Here, $\delta\boldsymbol{\mathcal{X}}$ is the control variable and **V** is the control variable transform. $\delta\boldsymbol{\mathcal{X}}$ is chosen such that it is of a smaller dimension

than $\delta\mathbf{x}$ which improves the conditioning of the problem. The aim of this is to have control variables with error covariance $\mathbf{I}$. This is accomplished with our choice of factorisation. This can be verified by rearranging (7) to

$$\mathbf{V}^T\mathbf{B}\mathbf{V} = \mathbf{I}$$

Reformulating the cost function around this control variable transform will give

$$J(\delta\boldsymbol{\mathcal{X}}) = \frac{1}{2}(\delta\boldsymbol{\mathcal{X}})^T\mathbf{I}(\delta\boldsymbol{\mathcal{X}}) + \frac{1}{2}\sum_{k=0}^{K}(\mathbf{H}\mathbf{V}\delta\boldsymbol{\mathcal{X}} - \mathbf{d}_k)^T\mathbf{R}_k^{-1}(\mathbf{H}\mathbf{V}\delta\boldsymbol{\mathcal{X}} - \mathbf{d}_k) \quad (9)$$

where $\mathbf{d}_k = \mathbf{y}_k^o - \mathbf{H}\mathbf{x}_k^b$

Finally, we assume the assimilation of each observation at every time step is independent of each other, giving us the final cost function

$$J(\delta\boldsymbol{\mathcal{X}}_k) = \frac{1}{2}\alpha(\delta\boldsymbol{\mathcal{X}}_k)^T\mathbf{I}(\delta\boldsymbol{\mathcal{X}}_k) + \frac{1}{2}(\mathbf{H}\mathbf{V}\delta\boldsymbol{\mathcal{X}}_{\boldsymbol{k}} - \mathbf{d}_k)^T\mathbf{R}_k^{-1}(\mathbf{H}\mathbf{V}\delta\boldsymbol{\mathcal{X}}_k - \mathbf{d}_k) \quad (10)$$

where α is a regularisation parameter analogous to the Tykhonov regularisation. k refers to the time step index, but since we are using 3DVar where the time domain is not explicitly considered, it is analogous to having k different background state vector measurements at the same timestep. Choosing $\alpha = 1$ is considered as giving the same relative weight to the observations compared to the background state. The minimisation of (10) is easier and better conditioned than a full formed version. The gradient in the reduced space is then

$$\nabla J(\delta\boldsymbol{\mathcal{X}}_k) = \mathbf{V}^T\nabla J(\delta\mathbf{x}_k) = \alpha\mathbf{V} + \mathbf{V}^T\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{H}\mathbf{V}\delta\boldsymbol{\mathcal{X}}_k - \mathbf{d}_k) \quad (11)$$

III. Ensemble Methods

A. Ensemble Formulation

If the background state vector is denoted with, $\mathbf{x}^b$, then an ensemble of state vectors is denoted with

$$\mathbf{x}^b_{(1)}, \mathbf{x}^b_{(2)}, .., \mathbf{x}^b_{(N)} \quad (12)$$

If we denote the ensemble mean to be $\overline{\mathbf{x}}^b$, then $\mathbf{V}_{ens}$, the background state perturbations are found with

$$\mathbf{V}_{ens} = \mathbf{X}^b = \frac{1}{\sqrt{N-1}}(\mathbf{x}^b_{(1)} - \overline{\mathbf{x}}^b, \mathbf{x}^b_{(2)} - \overline{\mathbf{x}}^b, ..., \mathbf{x}^b_{(N)} - \overline{\mathbf{x}}^b) \quad (13)$$

Here, $\mathbf{V}_{ens}$ and $\mathbf{X}^b$ are a n x N matrix called the ensemble background perturbation matrix. We will denote the rank-deficient version of the background error covariance matrix as $\mathbf{P}^b$ where

$$\mathbf{P}^b = {\mathbf{X}^b}^T\mathbf{X}^b \quad (14)$$

Using the ensembles in 3D Ensemble Variational DA (3DEnVar) then proceed similarly with equations (8), (10) and (11) just with $\mathbf{V}_{ens}$ instead of $\mathbf{V}$.

$$\delta\mathbf{x} = \mathbf{V}_{ens}\delta\boldsymbol{\mathcal{X}}_{ens} \quad (15)$$

Here, we will proceed with 3DEnVar with static ensembles as a proof of concept. Although the ensemble is static in the sense it does not evolve with time, it still contains the flow-dependent information at time $k = 0$ which is still beneficial for a 3D analysis that assumes no time dependence.

The success of ensemble DA depends on the choice of the ensemble. Previous work [18] suggests using a free run of the model as a source of ensemble members. However, the timeframe for our model is much smaller than the timeframes for natural weather prediction (NWP) applications meaning that the ensembles will not accurately represent the variability.

The choice of ensemble for our application needs to be able to capture the variability with time of the background error and the correlations between features via the sampling. Our devised method was to divide the collection of background states, $\underline{\mathbf{x}}^b$ based on the size of the ensemble into N groups with each group denoted $\underline{\mathbf{x}}^b_{(i)}$ meaning the ith group. The mean and standard deviation of each group is then calculated and used to sample the ensemble members from.

Algorithm 1 Build Ensemble

Inputs: $\underline{\mathbf{x}}^b$
$i = 0$, $N =$ ensemble size, $n = length(\mathbf{x}^b)$
for $\underline{\mathbf{x}}^b_{(i)}$ in $array\_split(\underline{\mathbf{x}}^b, N)$ **do**
  $\mu_{(i)} = mean(\underline{\mathbf{x}}^b_{(i)})$
  $\sigma^2_{(i)} = standard\_deviation(\underline{\mathbf{x}}^b_{(i)})$
  $ensemble[:, i] = normal\_distribution(\mu_{(i)}), \sigma^2_{(i)})$
  $i = i + 1$
end for
$ensemble\_mean = mean(ensemble)$
for $i = 0, 1, .., N$ **do**
  $\mathbf{V}_{ens}[:, i] = ensemble[:, i] - ensemble\_mean$
end for
return $\mathbf{V}_{ens}$

Algorithm 1 details the how the ensembles are formed and $\mathbf{V}_{ens}$ is built. The full background state matrix, $\underline{\mathbf{x}}^b$ is split into N groups each of size $n \times \frac{n}{N}$. The means and standard deviations of the n rows are calculated and used to sample from a normal distribution to form the ensemble. The ensemble mean is then calculated and subtracted from each ensemble member to form $\mathbf{V}_{ens}$

The low rank of the ensemble error covariance matrix, $\mathbf{P}_b$ will result in *sampling errors*. Furthermore, the rank deficiency causes spurious correlations at long distances meaning points which are located far from each other and are expected to have little correlation will exhibit some form of correlation due to the nature of sampling in ensemble methods. Therefore, methods such as *localisation* have been developed to handle these errors.

B. Localisation

$\delta\boldsymbol{\mathcal{X}}$ is of a smaller dimension than $\delta\mathbf{x}$ effectively having fewer degrees of freedom. This rank deficiency can be mitigated by *Schur localisation*. The insufficient rank will inevitable cause long-range correlations. These correlations are

considered *spurious*, meaning that they are nonphysical and not present in the true background error covariance matrix.

The idea is to smooth out the long-range correlations in $\mathbf{P}^b$ by regularisation and thus increase its rank. We start with our background error covariance, $\mathbf{P}^b$ defined in (14). $\mathbf{P}^b$ can be viewed as a symmetric matrix with rows i and columns j. $\mathbf{P}^b_{ij}$ denotes the covariance between points i and j. The goal is to reduce the magnitude of the covariance depending on the distance between i and j.

$$\mathbf{P}^C = \mathbf{C} \circ \mathbf{P}^b \tag{16}$$

where each element in $\mathbf{C}$ consists of values between 0 and 1 and $\circ$ denotes the *Schur product* or element wise product of the matrices. However, in our formulations of the cost function, (10), $\mathbf{P}^b$ does not appear in our equations, but $\mathbf{V}$ does. So instead of replacing $\mathbf{P}^b$ with $\mathbf{P}^C$, we need to replace $\mathbf{V}_{ens}$ with $\mathbf{V}^C_{ens}$ where

$$\mathbf{V}^C_{ens} = \mathbf{C'} \circ \mathbf{V}^1_{ens}, \mathbf{C'} \circ \mathbf{V}^2_{ens}, ..., \mathbf{C'} \circ \mathbf{V}^N_{ens} \tag{17}$$

where $\mathbf{V}^1_{ens}$ is an $n \times n$ matrix where every column is identical to the first column in $\mathbf{V}_{ens}$. Same goes for $\mathbf{V}^2_{ens}$ but with the second column and so on until $\mathbf{V}^N_{ens}$. $\mathbf{C'}$ is related to $\mathbf{C}$ by

$$\mathbf{C} = \mathbf{C'}^T\mathbf{C'} \tag{18}$$

To formulate $\mathbf{C'}$, eigendecomposition is carried out on $\mathbf{C}$

$$\mathbf{C} = \mathbf{E}\lambda\mathbf{E}^T \tag{19}$$

where $\mathbf{E}$ is an $n \times r$ matrix containing the eigenvectors and r is the number of dominant empirical orthogonal functions (EOF) modes we choose to retain. λ is a $r \times r$ diagonal matrix which contains the corresponding eigenvalues of the eigenvectors in $\mathbf{E}$. $\mathbf{C'}$ is then

$$\mathbf{C'} = \mathbf{E}\lambda^{1/2} \tag{20}$$

In this work, a choice of both horizontal localisation and vertical localisation is employed using different localisation functions. This choice is based off [19] and is motivated by the fact that the horizontal scale and vertical scale influence the system differently for air pollution forecasting. Problems which involve the atmosphere usually exhibit this behaviour hence this choice of localisation.

The localisation matrix, $\mathbf{C}$ is now split into horizontal and vertical matrices, $\mathbf{C}_h$ and $\mathbf{C}_v$ and (16) becomes

$$\mathbf{P}^C = \mathbf{C}_v \circ (\mathbf{C}_h \circ \mathbf{P}^b) \tag{21}$$

$$\mathbf{V}^C_{ens} = \mathbf{C}'_v \circ (\mathbf{C}'_h \circ \mathbf{V}_{ens}) \tag{22}$$

The horizontal localisation [20] function used in this paper is according to

$$\rho(s) = \begin{cases} 1 & ; s \leq L_h/2 \\ \frac{1}{2}\left\{1 + \cos\left[\frac{2\pi(s - L_h/2)}{L_h}\right]\right\} & ; L_h/2 \leq s < L_h \\ 0 & ; s \geq L_h \end{cases} \tag{23}$$

where L_h is a horizontal localisation length scale which acts as a cutoff distance and s is the horizontal distance.

The elements in the vertical localisation matrix, $\mathbf{C}_v$ is calculated based on [21] given by

$$\rho(\triangle z) = \frac{1}{1 + \left(\frac{\triangle z}{L_v}^2\right)} \tag{24}$$

where $\triangle z$ is the vertical distance and L_v is the vertical length scale which is the size of 1 grid cell unit, in our case $L_v = 10$.

The selection criteria for r_h and r_v are based on the explained variance for each of the EOF modes and to select enough modes to make up at least 90% of the variance. Given $\mathbf{E}$ is the matrix of all the eigenvectors and $\mathbf{e}_i$ is the ith column of $\mathbf{E}$, the explained variance of the ith EOF mode, σ^2_i is given by

$$\sigma^2_i = \frac{Var[\mathbf{e}_i]}{\sum_{i=0}^{n} Var[\mathbf{e}_i]} \tag{25}$$

One important thing to note is that localisation effectively increases the length of the control vector, $\delta\mathbf{X}_{ens}$ from size N to rN. Therefore carrying out localisation increases the computation time [18]. This is why the EOF approach in (19) and (20) are pivotal in reducing this additional burden. Furthermore, localisation may also remove some true long-range correlations. Therefore it is important to consider the nature of the problem and choice of localisation function and parameters when using with an ensemble DA system.

C. DA Algorithm

The cost function in (10) is minimised around the perturbation of the state $\delta\mathbf{x}$, instead of on the state vector $\mathbf{x}$ directly as it is more stable. The minimisation will be carried out iteratively via the L-BFGS method which is proven to be efficient for large scale optimisation problems. [22].

Algorithm 2 details each step of the 3DEnVar Data Assimilation. First, $\mathbf{C}'_h$ and $\mathbf{C}'_v$ are respectively constructed with (23) and (24). $\mathbf{V}_{ens}$ is the built based on Algorithm 1. Localisation is then applied to get $\mathbf{V}^C_{ens}$ and the initial values are initialised to 0 [18].

This initial guess is then projected to the reduced space using the control variable transform. The while loop initiates the beginning of the L-BFGS steps. The cost functions are then evaluated using equations (26) and (27) with V^C_{ens},

$$\begin{aligned} J(\delta\boldsymbol{\mathcal{X}}_k) = &\frac{1}{2}\alpha(\delta\boldsymbol{\mathcal{X}}_k)^T\mathbf{I}(\delta\boldsymbol{\mathcal{X}}_k) \\ &+ \frac{1}{2}(\mathbf{H}\mathbf{V}^C_{ens}\delta\boldsymbol{\mathcal{X}}_{\boldsymbol{k}} - \mathbf{d}_k)^T\mathbf{R}^{-1}_k(\mathbf{H}\mathbf{V}^C_{ens}\delta\boldsymbol{\mathcal{X}}_k - \mathbf{d}_k) \end{aligned} \tag{26}$$

$$\nabla J(\delta\boldsymbol{\mathcal{X}}_k) = \alpha\mathbf{V}^C_{ens} + (\mathbf{V}^C_{ens})^T\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{H}\mathbf{V}^C_{ens}\delta\boldsymbol{\mathcal{X}}_k - \mathbf{d}_k) \tag{27}$$

with α being set equal to 1 [11] and $\mathbf{H} = \mathbf{I}$ as the observations and model outputs exist within the same space.

After convergence the analysis vector, $\delta\boldsymbol{\mathcal{X}}^a_k$ is projected back up to the model space and the analysis state vector at time k, $\delta\mathbf{x}^a_k$ is calculated. This is repeated for all k giving $\underline{\mathbf{x}}^a$ which is a matrix of $\mathbf{x}^a_k$.

Algorithm 2 3DEnVar Algorithm

Inputs: $\underline{\mathbf{x}}^b, \underline{\mathbf{y}}^o$
Construct $\mathbf{C}'_h$ and $\mathbf{C}'_v$
Build $\mathbf{V}$ with Ensemble: $\mathbf{V}_{ens} = BuildEnsemble(\underline{\mathbf{x}}^b)$
Apply Localisation: $\mathbf{V}^C_{ens} = \mathbf{C}'_v \circ (\mathbf{C}'_h \circ \mathbf{V}_{ens})$
Define initial guess: $\delta\mathbf{x}_0$ as array of zeroes with length n
Define $\mathbf{R}$ from the observed data, $\underline{\mathbf{y}}^o$
$\delta\boldsymbol{\mathcal{X}}_0 = (\mathbf{V}^C_{ens})^{-1}\delta\mathbf{x}_0$
for $k = 0, 1, 2, ...K$ **do**
  $\mathbf{d}_k = \mathbf{y}^o_k - \mathbf{H}_k\mathbf{x}^b_k$
  while $||\nabla J|| > \epsilon$ **do**
    Compute $J(\delta\boldsymbol{\mathcal{X}}_k)$ using (26)
    Compute $\nabla J(\delta\boldsymbol{\mathcal{X}}_k)$ using (27)
  end while
  $\delta\mathbf{x}^a_k = \mathbf{V}^C_{ens}\delta\boldsymbol{\mathcal{X}}^a_k$
  $\mathbf{x}^a_k = \mathbf{x}^b_k + \delta\mathbf{x}^a_k$
  $\delta\boldsymbol{\mathcal{X}}_{k+1} = \delta\boldsymbol{\mathcal{X}}^a_k$
end for
return $\underline{\mathbf{x}}^a$

IV. Testing

Algorithm 2 has been implemented entirely in python 3.6 environment and the external libraries used are numpy, scipy and vtk. The performance of the DA is evaluated using the mean squared error (MSE) based on the equation

$$MSE(\mathbf{x}) = \frac{||\mathbf{x} - \mathbf{x}_C||_{L^2}}{||\mathbf{x}_C||_{L^2}} \tag{28}$$

where $\mathbf{x}_C$ is a control variable. For our application, the control variable will be taken as the observations $\mathbf{y}^o$. Since the observations and model output is within the same space, $\boldsymbol{\mathcal{H}}$ resolves to identity. In this paper, a realistic 3D case which includes 14 buildings represented the urban area around London South Bank University (LSBU) in Elephant and Castle, London, UK is investigated. The unstructured 3D mesh of the area used in Fluidity comprises of 100,040 nodes (Fig. 1). $K = 494$ timesteps of the model and observations are assimilated for analysis.

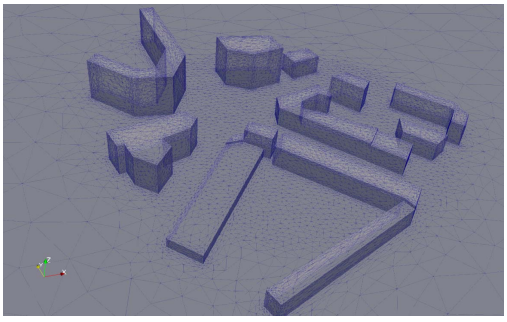
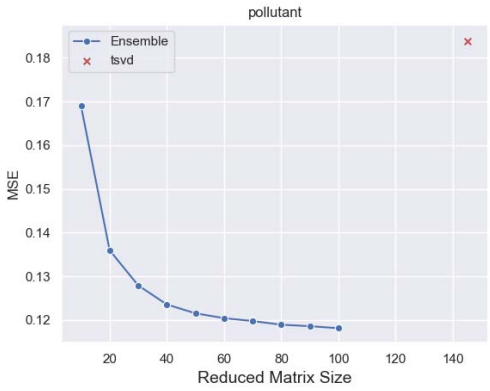


Fig. 1. Unstructured Mesh of Problem Domain

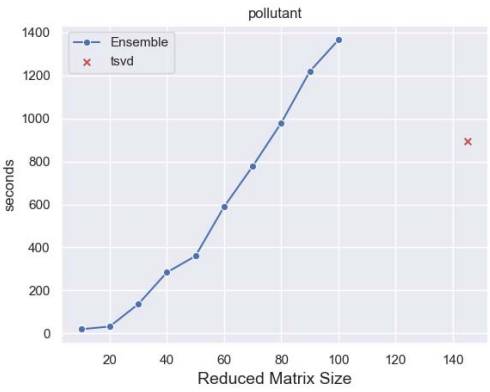
V. Results

A. Ensemble Size Evaluation

In Fig. 2, the mean squared error (MSE) and execution times for data assimilation with ensembles of different sizes and TSVD are plotted. Ensembles of size $10, 20, ..., 100$ are constructed based on Algorithm 1 to represent the background error covariance in 3DEnVar. For the TSVD method, a truncation parameter $\tau = 145$ is selected based on previous work in [11]. The values of the MSE are computed using (28). The MSE decreases with an increase in the ensemble size until it reaches a plateau. Note that the MSE for TSVD (which has a reduced matrix size of 145) is also plotted on the graph denoted by the red cross. For pollutant concentration, the ensemble method outperforms the TSVD method achieving a lower error even with an ensemble size of 10. This is a good indicator that the chosen ensemble members covered the correlation of the background error well even though at a reduced rank.

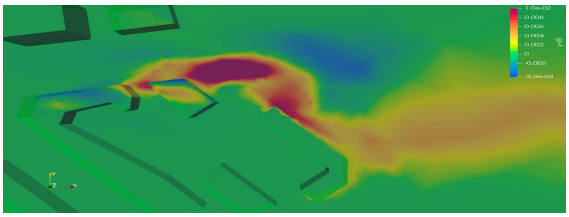


(a) Mean Squared Error of Pollutant

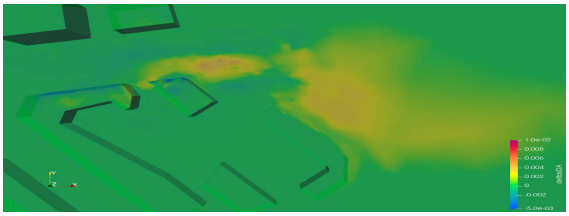


(b) Execution Time of Pollutant

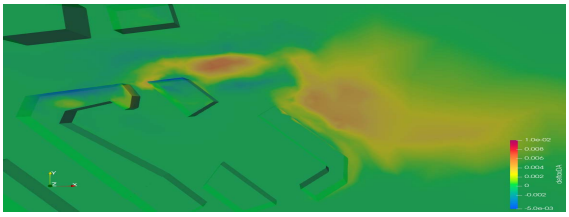
Fig. 2. (a) Mean square error of DA result of pollutant and velocity; (b) Execution time. MSE and execution time are plotted for 3DEnVar carried out with ensemble sizes of 10,20,...,100. The comparison is with the TSVD method (for $\tau = 145$) plotted as benchmark.



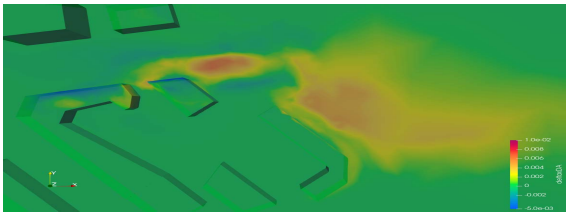
(a) $\mathbf{y} - \mathbf{x}^b$



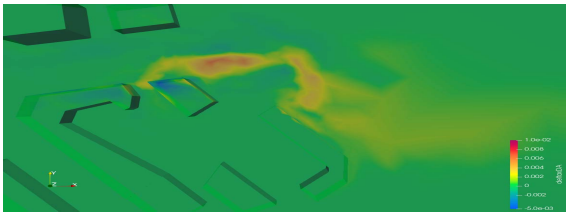
(b) $\delta\mathbf{x}^a$, ensemble size=10



(c) $\delta\mathbf{x}^a$, ensemble size=40



(d) $\delta\mathbf{x}^a$, ensemble size=70

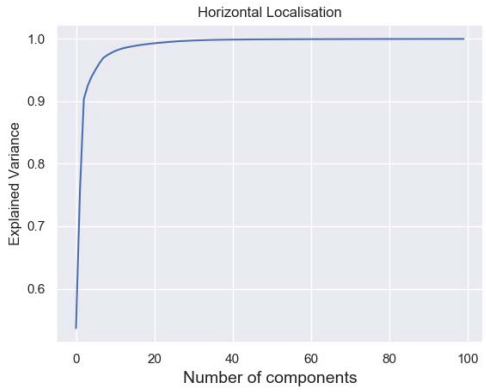


(e) $\delta\mathbf{x}^a$, TSVD $\tau = 145$

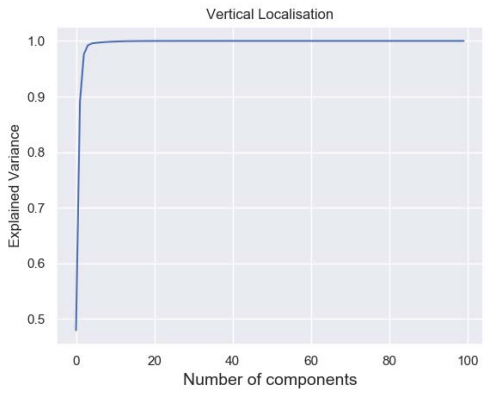
Fig. 3. (a) The innovation, $\mathbf{d}$ of C. (b),(c),(d),(e) Perturbation of the C after DA, $\delta\mathbf{x}^a$ for ensemble sizes 10,40,70. (f) Perturbation of the C after DA, $\delta\mathbf{x}^a$ for TSVD preconditioned matrix with $\tau = 145$. Scale is from -0.005 (blue colour) to 0.01 (red colour). Results show values averaged across $K = 494$ timesteps.

Execution times (Fig. 2(b)) increase linearly with the ensemble size, as expected. The TSVD method converges faster than ensemble sizes larger than 80. This is due to the fact that TSVD is a better form of preconditioning than with ensembles [23]. Hence, the selection of a good ensemble size is a trade off between accuracy and execution speed. From Fig. 2, a good choice would be an ensemble size of 40, which is the point where the accuracy starts to plateau while also being two times faster than the TSVD method. Fig. 3 shows that for small ensemble sizes, it is unable to properly assimilate the observations. It manages to capture the profile in the misfit $(\mathbf{y} - \mathbf{x}^b)$ but does not weight the observations heavily resulting in the disparity. At larger ensemble sizes it is able to assimilate the observations in more clearly and weighs against the background state appropriately. Most importantly, the DA for ensemble methods is able to assimilate the observations further from surfaces better than TSVD, indicating that ensemble methods outperform TSVD for data assimilation at sparse regions.

B. Localisation Evaluation



(a) Explained variance for horizontal localisation EOF, r_h



(b) Explained variance for vertical localisation EOF, r_v

Fig. 4. Graphs of explained variance for localisation EOFs

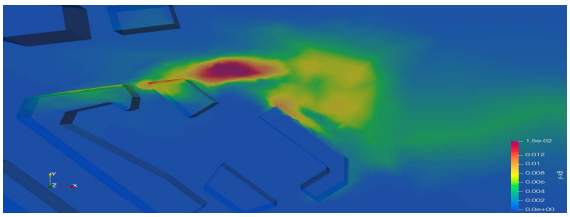
The values of r_h and r_v were found by plotting the the cumulative sum for the explained variances for the number of components of the eigendecomposition. From Fig. 4, the first 3 dominant EOFs are able to capture 90% of the explained variance for both horizontal and vertical localisation. Horizontal and vertical localisation is carried out on the

background error covariance matrix with these parameters. Table I displays the mean squared errors and execution times for the cases of TSVD with $\tau = 145$, ensemble method with ensemble size of 40, and ensemble methods with horizontal localisation only and both horizontal and vertical localisation. The results indicate that applying both horizontal and vertical localisation to the ensembles result in lower error. The MSE for DA without localisation was shown to plateau at 0.12 but localisation results have reached 0.1 and lower. This means localisation has improved the potential of the 3DEnVar model used.

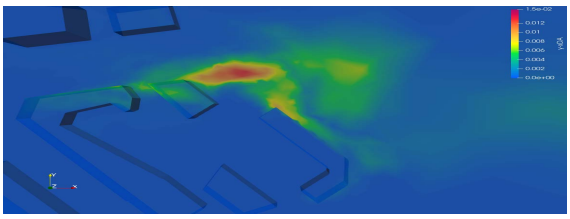
However, localisation increases the execution time of DA. This is to be expected as localisation increases the size of the control variable and the rank of the background error covariance matrix. Applying both horizontal and vertical localisation reduces the error to 0.087 but increases the running time to roughly 167 seconds per timestep. In comparison, applying only horizontal localisation has a running time of 1.94 seconds per timestep which is comparable with TSVD but has significantly lower error (by a factor of 1.8). From the results, there is a clear tradeoff for choice of localisation between performance and speed.

TABLE I
TABLE OF MSE AND AVERAGE EXECUTION TIMES FOR A SET OF OBSERVATIONS PER TIMESTEP FOR TSVD, ENSEMBLE METHODS AND ENSEMBLE METHODS WITH LOCALISATION

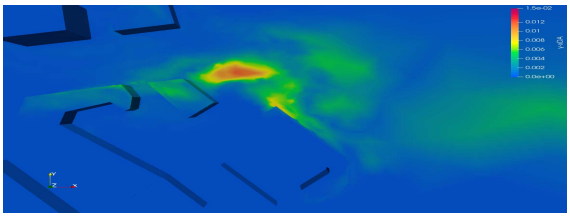
	MSE	Runtime per timestep (s)
No DA	0.261	-
TSVD	0.184	1.8
Ensemble Method (ens. size=40)	0.123	0.6
Horizontal Localisation	0.106	1.94
Horizontal & Vertical Localisation	0.087	167.5



(a) Error in C before 3DEnVar



(b) Error in C after 3DEnVar



(c) Error in C after 3DEnVar with localisation

Fig. 5. Comparison of absolute error of C before ($|\mathbf{x}^b - \mathbf{x}_C|$) and after ($|\mathbf{x}^a - \mathbf{x}_C|$) assimilation. Scale is from 0 (blue colour) to 0.015 (red colour). Ensemble size of 40 used for data assimilation. Results show values averaged across $K = 494$ timesteps

Fig. 5 shows the mean absolute error of the forecasts before DA, after DA and with localisation. The ensemble method is able to assimilate the profile of the observations at regions to the right which are further away from the surfaces of the buildings correctly forecasting that the Fluidity model had underestimated the pollutant concentration.

For the case with localisation, the areas surrounding the edges of the building have lower absolute errors and are more concentrated. This showcases the effect of localisation restricting the error correlation around these areas to provide a better forecast. However, the reverse is observed at the sparse region to the right where the error is higher than the case without localisation. At this region, localising the background errors have the opposite effect as it is trying to derive information from a limited sparse region and in this case depends more on the long range correlations.

VI. Conclusion and Future Works

A form of 3DEnVar has successfully been implemented for use with a computational fluid dynamics model named Fluidity and air pollution data in a large scale 3D environment. The DA system uses a static ensemble with its members sampled from groups of the background state. Localisation was carried out on the ensembles to reduce spurious long-range correlations. Three dominant EOFs were used for both horizontal and vertical localisation as they were calculated to contain 90% of the explained variance.

The results from the investigation show that the mean squared error (MSE) of the ensemble methods outperformed TSVD for pollutant concentration and velocity. Furthermore, the execution time for ensemble methods is much quicker than for TSVD for ensemble sizes less than 80, achieving convergence faster. This is indicative of ensembles having a good spread for the background error statistics for pollutant concentration. The error decreases as the ensemble size increases signifying that more ensemble members improves the rank of the background error covariance matrix and thus better represents it. However, the 3DEnVar method provides a better analysis at sparse regions further from buildings. An ensemble size of 40 is shown to exhibit a good tradeoff between performance and execution time.

Localisation was shown to greatly improve the error at regions near edges and surfaces. At those regions, local correlations dominate but at regions further from the surface, localisation is seen to not affect and even worsen the error. At these regions, the analysis is reliant on long-range correlations

which localisation removes hence the increase in error. Overall however, localisation was able to achieve better performance than without.

Furthermore, localisation greatly increases the execution time for DA due to its nature of increasing the rank of the background error and the size of the control vector. From the results, applying both horizontal and vertical localisation would require 167 seconds per timestep of assimilation which is far from real time.

To summarise, this work has proven the performance of using ensemble methods with 3D Variational Data Assimilation on air pollution data. It is also the first attempt at applying ensemble methods to a 3D case at this scale. Localisation of the background error show a tradeoff in performance with speed and the problem domain needs to be considered carefully.

The results are adequate to justify future work in using semi-static or fully dynamic ensembles for the background error covariance (both in 3D and 4D) [24] for this application. Furthermore, hybrid approaches of mixing a static and ensemble background error covariances [25] can also be explored to further refine the DA system. A coupling of the system with Gaussian Recursive Filter [26] can also be explored in a parallel computing environment [27]. The developed 3DEnVar model was used for the first time for air pollution flow in a 3D urban environment. The methods used are generic enough and not limited by the software making it usable for other DA problems.

ACKNOWLEDGMENTS

This work is supported by the EPSRC Grand Challenge grant Managing Air for Green Inner Cities (MAGIC) EP/N010221/1 and by the EPSRC Centre for Mathematics of Precision Healthcare EP/N0145291/1

REFERENCES

[1] J. Lelieveld, J. S. Evans, M. Fnais, D. Giannadaki, and A. Pozzer, "The contribution of outdoor air pollution sources to premature mortality on a global scale," *Nature*, vol. 525, no. 7569, p. 367, 2015.

[2] J. Song, S. Fan, W. Lin, L. Mottet, H. Woodward, M. D. Wykes, R. Arcucci, D. Xiao, J.-E. Debay, H. ApSimon, E. Aristodemou, D. Birch, M. Carpentieri, F. Fang, M. Herzog, G. R. Hunt, R. L. Jones, C. Pain, D. Pavlidis, A. G. Robins, C. A. Short, and P. F. Linden, "Natural ventilation in cities: the implications of fluid mechanics," *Building Research & Information*, vol. 46, no. 8, pp. 809–828, 2018.

[3] H. B. Awbi and G. Gan, "Predicting air flow and thermal comfort in offices," *ASHRAE Journal*, vol. 36, no. 2, pp. 17–21, 02 1994.

[4] P. Murad, "Closed-form solutions to the transient/steady-state navier-stokes fluid dynamic equations," *AIP Conference Proceedings*, vol. 813, p. 1264, 01 2006.

[5] S.-C. Mou, Y.-X. Luan, W.-T. Ji, J.-F. Zhang, and W.-Q. Tao, "An example for the effect of round-off errors on numerical heat transfer," *Numerical Heat Transfer, Part B: Fundamentals*, vol. 72, no. 1, pp. 21–32, 2017.

[6] A. C. Lorenc, N. E. Bowler, A. M. Clayton, S. R. Pring, and D. Fairbairn, "Comparison of hybrid-4denvar and hybrid-4dvar data assimilation methods for global nwp," *Monthly Weather Review*, vol. 143, no. 1, pp. 212–229, 2015.

[7] M. Buehner, P. L. Houtekamer, C. Charette, H. L. Mitchell, and B. He, "Intercomparison of variational data assimilation and the ensemble kalman filter for global deterministic nwp. part i: Description and single-observation experiments," *Monthly Weather Review*, vol. 138, no. 5, pp. 1550–1566, 2010.

[8] O. Talagrand and P. Courtier, "Variational assimilation of meteorological observations with the adjoint vorticity equation. i: Theory," *Quarterly Journal of the Royal Meteorological Society*, vol. 113, no. 478, pp. 1311–1328, 1987.

[9] P. Courtier, E. Andersson, W. Heckley, D. Vasiljevic, M. Hamrud, A. Hollingsworth, F. Rabier, M. Fisher, and J. Pailleux, "The ECMWF implementation of three-dimensional variational assimilation (3D-Var). I: Formulation," *Quarterly Journal of the Royal Meteorological Society*, vol. 124, no. 550, pp. 1783–1807, 1998.

[10] S. R. Ibrahim and A. Fregolent, "Matrix decomposition techniques: Use and limitations in modal analysis," in *Proceedings-SPIE The International Society for Optical Engineering*, vol. 1, 1998, pp. 91–96.

[11] R. Arcucci, L. Mottet, C. Pain, and Y.-K. Guo, "Optimal reduced space for variational data assimilation," *Journal of Computational Physics*, vol. 379, pp. 51–69, 2019.

[12] R. N. Bannister, "A review of operational methods of variational and ensemble-variational data assimilation," *Quarterly Journal of the Royal Meteorological Society*, vol. 143, no. 703, pp. 607–633, 2017.

[13] J. L. Anderson, "Localization and sampling error correction in ensemble kalman filter data assimilation," *Monthly Weather Review*, vol. 140, no. 7, pp. 2359–2371, 2012.

[14] R. N. Bannister, "A review of forecast error covariance statistics in atmospheric variational data assimilation. ii: Modelling the forecast error covariance statistics," *Quarterly Journal of the Royal Meteorological Society*, vol. 134, no. 637, pp. 1971–1996, 2008.

[15] G. Dong, B. Jüttler, O. Scherzer, and T. Takacs, "Convergence of tikhonov regularization for solving ill-posed operator equations with solutions defined on surfaces," *Inverse Problems & Imaging*, vol. 11, no. 2, pp. 221–246, 2017.

[16] A. C. Lorenc, S. P. Ballard, R. S. Bell, N. B. Ingleby, P. L. F. Andrews, D. M. Barker, J. R. Bray, A. M. Clayton, T. Dalby, D. Li, T. J. Payne, and F. W. Saunders, "The met. office global three-dimensional variational data assimilation scheme," *Quarterly Journal of the Royal Meteorological Society*, vol. 126, no. 570, pp. 2991–3012, 2000.

[17] A. T. Weaver, C. Deltel, E. Machu, S. Ricci, and N. Daget, "A multivariate balance operator for variational ocean data assimilation," *Quarterly Journal of the Royal Meteorological Society*, vol. 131, no. 613, pp. 3605–3625, 2005.

[18] L. Axell and Y. Liu, "Application of 3-d ensemble variational data assimilation to a baltic sea reanalysis 19892013," *Tellus A: Dynamic Meteorology and Oceanography*, vol. 68, no. 1, p. 24220, 2016.

[19] C. Liu, Q. Xiao, and B. Wang, "An ensemble-based four-dimensional variational data assimilation scheme. part ii: Observing system simulation experiments with advanced research wrf (arw)," *Monthly Weather Review*, vol. 137, no. 5, pp. 1687–1704, 2009.

[20] F. Counillon and L. Bertino, "Ensemble optimal interpolation: multivariate properties in the gulf of mexico," *Tellus A*, vol. 61, no. 2, pp. 296–308, 2009.

[21] H. Zhang, J. Xue, S. Zhuang, G. Zhu, and Z. Zhu, "Grapes 3d-var data assimilation system ideal experiments," *Acta Meteor. Sin*, vol. 62, pp. 31–41, 2004.

[22] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Math. Program.*, vol. 45, no. 1-3, pp. 503–528, Aug. 1989.

[23] R. Arcucci, L. DAmore, J. Pistoia, R. Toumi, and A. Murli, "On the variational data assimilation problem solving and sensitivity analysis," *Journal of Computational Physics*, vol. 335, pp. 311–326, 2017.

[24] G. Desroziers, J.-T. Camino, and L. Berre, "4DEnVar: link with 4D state formulation of variational assimilation and different possible implementations," *Quarterly Journal of the Royal Meteorological Society*, vol. 140, no. 684, pp. 2097–2110, 2014.

[25] J. Gao, C. Fu, D. J. Stensrud, and J. S. Kain, "Osses for an ensemble 3dvar data assimilation system with radar observations of convective storms," *Journal of the Atmospheric Sciences*, vol. 73, no. 6, pp. 2403–2426, 2016.

[26] S. Cuomo, A. Galletti, G. Giunta, and L. Marcellino, "Numerical effects of the gaussian recursive filters in solving linear systems in the 3dvar case study," *Numerical Mathematics: Theory, Methods and Applications*, vol. 10, no. 3, pp. 520–540, 2017.

[27] A. Galletti, G. Giunta, L. Marcellino, and D. Parlato, "An algorithm for gaussian recursive filters in a multicore architecture," in *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2017, pp. 507–511.