Dear Anonymous Referee #2,

On behalf of all co-authors, first of all I would like to thank you for your efforts on reviewing our manuscript.

In the following we will give our response to the comments. To make the changes easier to identify, we have numbered them.

Best regards,
Guangliang Fu
on behalf all co-authors

(The revised manuscript is in the latter part of this pdf.)

Reply to comments:

1. *The most costly step identified by the authors is a simple matrix multiplication, $A^a = A^f X$. Such a simple step should be very easily distributed over the available CPU's and should be very fast, so in fact I do not even understand the "problem" that the authors want to solve.*

   Response:
   The problem is "we aim to accelerate the runtime time to less than the simulation time", which is important for the assimilation in an operational sense. It is described in line(s) 5.18–5.21:
   " The evaluation result of the conventional EnKF is shown in Table 1 (the middle column). It can be seen that the total computational time (4.36 h) is relatively large compared to the simulation window (3.0 h, i.e., from 9:00–12:00 UTC, 18 May, 2010), which is too much in an operational sense. Therefore, in this study, we aim to accelerate the computation to within an acceptable runtime (i.e., requires less runtime than the time period of the data assimilation application). "

   We agree that the time-consuming part $\mathbf{A^a = A^f X}$ is not a complicated matrix multiplication. However, we have employed the fastest available CPU on Cartesius (Netherlands' supercomputer. Each node is configured with 2 × 12-core 2.6 GHz Intel Xeon E5-2690 v3 (Haswell) CPUs and with memory 64 GB), the total computational time (4.36 h) is still not acceptable, compared to a simulation time of 3 h. This is mainly because $\mathbf{A^a = A^f X}$ takes large amounts of time for the analysis step of 3.14 h. Thus, that motivated us to develop the Mask-State algorithm and the result shows it is efficient for our case. Therefore, our problem can be briefly understood as "a further acceleration of $\mathbf{A^a = A^f X}$" based on model dynamics.

   In order to speed up the computation, we can do two things: (1) reduce the amount of operations in $\mathbf{A^f X}$; (2) using parallel computation. For the application at hand, the Mask-state algorithm strongly reduces the total amount of computation; parallel processing can be applied after that.

2. *This topic by itself is very technical, and arguably does not belong in the "geoscientific" journal GMD. The results may justify publication if the approach would be applicable to a large class of assimilation problems, but this is not the case as mentioned by the authors themselves. Using masks is only applicable for a very limited set of problems, typically single point source releases of short-lived species.*

Response:
Using Mask-State (MS) is applicable for many problems, where the domain is not fully polluted by the species. It is certainly not limited to "typically single point source releases of short-lived species". It doesn't matter what the emission looks like and whether the releases are "short-lived species" or "long-lived species". Given an assimilation problem, the only restriction for MS to gain an acceleration is whether the whole domain is fully polluted or partly polluted.

MS is therefore suitable for many assimilation problems, such as all the volcanic-related ash/gas assimilation, e.g., assimilation of satellite data/LIDAR data/in situ data; (sand/desert) dust storm related assimilation; tornado-related assimilation; assimilation of exploding nuclear plants or factories; chemicals or oils leaking on seas; global (forecast) fire assimilation; assimilation of environmental pollutant transport, e.g., severe smog.

For all the above applications, MS can accelerate $\mathbf{A^a} = \mathbf{A^f X}$. As a clarification, we have clearly mentioned this in the new version in line(s) 11.15–11.22:
" Using MS is also applicable for many other assimilation problems, where the domain is not fully polluted by the species. It does not matter what the emission looks like and whether the releases are short- or long-lived species. Given an assimilation problem, the only restriction for MS to gain an acceleration is whether the whole domain is fully polluted or partly polluted. The assimilation problems where MS can can achieve the acceleration effect on the computations of $\mathbf{A^a} = \mathbf{A^f X}$ are such as all the volcanic-related ash/gas assimilation, e.g., assimilation of satellite data/LIDAR data/in situ data; (sand/desert) dust storm related assimilation; tornado-related assimilation; assimilation of exploding nuclear plants or factories; chemicals or oils leaking on seas; global (forecest) fire assimilation; assimilation of environmental pollutant transport, e.g., severe smog. "

This study is submited as a "Development and technical paper" to GMD, focusing on the development of MS to accelerate a data assimilation application not only for the volcanic ash forecast problem which is used as a case study, but also for many other assimilation problems. Therefore, we believe it fits the scope of GMD.

3. *The problem is very idealised, with only a few observations (m=2). In this case the whole inversion problem in "observation space" is computationally very fast, which normally is not the case.*

Response:
Fu et al. (2016) described the methodology for an aircraft-based volcanic ash data assimilation problem, where (for one aircraft) at each assimilation time, two in situ measurements of PM_{10} and $PM_{2.5}$ are assimilated (thus m=2 here). The problem should not be considered as an idealised case, but an example of stationary data assimilation (although aircraft is not exactly stationary, but the measurement number at each assimilation time is steady and stationary).

We admit for m=2, the whole inversion problem in "observation space" is computationally fast. This is shown in Fig. 2**b**, where at one analysis step, the computational cost of the inversion problem ($\mathbf{X_4} = \mathbf{X_3}^{-1}$) is O($m^3$). However, for stationary data assimilation problems where mostly $m \ll n$ (n is the state number, in this case $n \sim 10^6$), the inversion problem is also fast compared to the part $\mathbf{A^a} = \mathbf{A^f X}$ (O(nN^2), N=100 is the ensemble size). Thus, the case m=2 can be taken as an example of stationary assimilation. Actually, this type of assimilation is a common assimilation type. For example, Barbu et al. (2009) assimilated measurements of SO_2 and SO_4 from the EMEP database. The assimilation set contains 17 sites for SO_2 and 27 for SO_4, thus m=44 ($m \ll n$).

Recently, a large number of national weather services have implemented ceilometer networks, mainly for monitoring the dispersion of volcanic ash clouds (Wiegner et al., 2014). These data set will be (and in part are already) available in near real time and will provide information about the (horizontal and) vertical distribution (with some restrictions due to cloud cover). Thus, they could be very promising candidates for stationary data assimilation for volcanic ash plumes. Under these stationary measurement setup ($m \sim 150$), based on Fig. 2**b**, the inversion problem is still not expensive (O(m^3)) compared to $\mathbf{A^a} = \mathbf{A^f X}$ (O(nN^2)) at one analysis step.

Therefore, for stationary/near-stationary data assimilation, it is usually the case that the inversion problem in "observation space" is computationally fast.

We also admit the problem discussed in the study is not the case where O(m)=O(n), e.g., for passive satellite data assimilation, where the inversion problem in the analysis step is the most time-consuming part (O(m^3)=O(n^3)), but this is not the case considered in this study for stationary data assimilation. We have added some related discussion in line(s) 13.29–13.33:

" However, in other applications when many measurements are assimilated (e.g., satellite-based or seismic-based data), and the number of measurements is of the same order as the number of state variables, the most time-consuming part will be the SVD. In these cases, the contributions of the mask-state algorithm will be limited. The reduction of the total computing time using the mask-state algorithm therefore is less significant, an effective acceleration algorithm for the analysis step must be used and should consider the computationally-expensive SVD in the first place. "

The revised manuscript starts from next page.

A mask-state algorithm to accelerate volcanic ash data assimilation: a case study with the LOTOS-EUROS model (version 1.10)

Guangliang Fu[1], Hai Xiang Lin[1], Arnold Heemink[1], Arjo Segers[2], Nils van Velzen[1,3], Tongchao Lu[4], Shiming Xu[5], and Sha Lu[1]

[1]Delft University of Technology, Delft Institute of Applied Mathematics, Mekelweg 4, 2628 CD Delft, The Netherlands.
[2]TNO, Department of Climate, Air and Sustainability, P.O. Box 80015, 3508 TA Utrecht, The Netherlands.
[3]VORtech, P.O. Box 260, 2600 AG Delft, The Netherlands.
[4]School of Mathematics, Shandong University, Jinan, Shandong 250100, China.
[5]Ministry of Education Key Laboratory for Earth System Modeling, Center for Earth System Science (CESS), Tsinghua University, Beijing, China

Correspondence to: Guangliang Fu (G.Fu@tudelft.nl)

Abstract. In this study, we investigate a strategy to accelerate the data assimilation algorithm. Based on evaluations of the computational time, the analysis step of the assimilation turns out to be the most expensive part. After a study on the characteristics of the ensemble ash state, we propose a mask-state algorithm which records the sparsity information of the full ensemble state matrix and transforms the full matrix into a relatively small one. This will reduce the computational cost in the analysis step. Experimental results show the mask-state algorithm significantly speeds up the analysis step. Subsequently, the total amount of computing time for volcanic ash data assimilation is reduced to an acceptable level. The mask-state algorithm is generic and thus can be embedded in any ensemble-based data assimilation framework. Moreover, ensemble-based data assimilation with the mask-state algorithm is promising and flexible, because it implements exactly the standard data assimilation without any approximation and it realizes the satisfying performance without any change of the full model.

1 Introduction

Volcanic ash erupted into atmospheres can lead to severe influences on aviation society (Gudmundsson et al., 2012). Turbine engines of airplanes are extremely threatened by the ash's ingestion (Casadevall, 1994). Thus, accurate real-time aviation advices are highly required during an explosive volcanic ash eruption (Eliasson et al., 2011). Recently, ensemble-based data assimilation (Evensen, 2003) has been evaluated very useful to improve volcanic ash forecasts and regional aviation advices (Fu et al., 2016). It corrects volcanic ash concentrations by continuously assimilating observations. In (Fu et al., 2016), real aircraft in situ measurements were assimilated using the ensemble Kalman filter (EnKF), which is the most known and popular ensemble-based assimilation method. Based on the validation with independent data, ensemble-based data assimilation was concluded powerful for improving the forecast accuracy.

However, to make the methodology efficient also in an operational (real-time) sense, the computational efforts must be acceptable. For volcanic ash assimilation problems, so far, no studies on the computational aspects have been reported in the literature. Actually, when large amounts of volcanic ash erupted into atmospheres, the computational speed of volcanic ash

forecasts is just as important as the forecast accuracy (Zehner, 2010). For example, due to the lack of a fast and accurate forecast system, the sudden eruption of the Eyjafjallajökull volcano in Iceland from 14 April to 23 May 2010, had caused an unprecedented closure of the European and North Atlantic airspace resulting in a huge global economic loss of 5 billion US dollars (Oxford-Economics, 2010). Since then, research on fast and accurate volcanic ash forecasts have gained much attention, because it is needed to provide timely and accurate aviation advices for frequently operated commercial airplanes. It was shown the accuracy of volcanic ash transport can be significantly improved by the assimilation system in (Fu et al., 2016). Therefore, it is urgent to also consider the computational aspect, i.e., improving the computational speed of the volcanic ash assimilation system as fast as possible. This is the main focus of this study.

Due to the computational complexity of ensemble-based algorithms and the large scale of dynamical applications, applying these methods usually introduces a large computational cost. This has been reported from literature on different applications. For example, for operational weather forecasting with ensemble-based data assimilation, Houtekamer et al. (2014) reported computational challenges at Canadian Meteorological Center with an operational EnKF featuring 192 ensemble members, using a large 600×300 global horizontal grid and 74 vertical levels. That an initialization requirement of over 7×10^{10} values to specify each ensemble, results in large computational efforts on the initialization and forecast steps in weather forecasting. For oil reservoir history-matching (Tavakoli et al., 2013), the reservoir simulation model usually has a large number of state variables, thus the forecasts of an ensemble of simulation models are often time-consuming. Besides, when time-lapse seismic or dense reservoir data is available, the analysis step of assimilating these large observations becomes very time-consuming (Khairullah et al., 2013). Large computational requirements of ensemble-based data assimilation have also been reported in ocean circulation models (Keppenne, 2000; Keppenne and Rienecker, 2002), tropospheric chemistry assimilation (Miyazaki et al., 2015), and many other applications.

To accelerate an ensemble-based data assimilation system, the ensemble forecast step can be first parallelized because the propagation of different ensemble members is independent. Thus if a computer with a sufficiently large number of parallel processors is available, all the ensemble members can be simultaneously integrated. In the analysis stage, to calculate the Kalman gain and the ensemble error covariance matrix, all ensemble states must be combined together. In weather forecasting and oceanography sciences, Keppenne (2000); Keppenne and Rienecker (2002); Houtekamer and Mitchell (2001) have reported using parallelization approaches to accelerate the expensive analysis stage. In reservoir history matching, a three-level parallelization has been proposed by Tavakoli et al. (2013); Khairullah et al. (2013) in recent years, to significantly reduce computational efforts of both forecast and analysis steps due to massive dense observations and large simulation models. The first parallelization level is to separately perform the ensemble simulations on different processors during the forecast step. This approach is usually quite efficient when a large ensemble size is used. However, the scale or model size of one reservoir simulation is constrained by the memory of a single processor. Thus, the second parallelization level is to perform one ensemble member simulation using a parallel reservoir model. These two levels do not deal with the analysis step, which collects all ensemble members to do computations usually on a single processor. Therefore, a third level of parallelization was implemented by Tavakoli et al. (2013); Khairullah et al. (2013) through parallelizing matrix-vector multiplications in the analysis steps. Furthermore, some other approaches on accelerating ensemble-based assimilation systems, have also been reported, such

as GPU-based acceleration (Quinn and Abarbanel, 2011) in Numerical Weather Prediction (NWP), domain decomposition in atmospheric chemistry assimilation (Segers, 2002; Miyazaki et al., 2015). The observations used in an assimilation system can be also optimized with some preprocessing procedures, as reported by Houtekamer et al. (2014).

Although for other applications, there were many efforts in dealing with large computational requirements in an ensemble-based data assimilation system, most of them cannot be directly used to accelerate volcanic ash data assimilation. This is because the acceleration algorithms are strongly dependent on specific problems, such as model complexity (high or low resolution), observation type (dense or sparse), primary requirement (accuracy or speed). These factors determine, for a specific application, which part is the most time-consuming, and which part is intrinsically sequential. Thus, no unified approach for efficient acceleration of all the applications can be found. Although the successful approaches in other applications cannot be directly employed in volcanic ash forecasts, their success do stress the importance of designing a proper approach based on the computational analysis of a specific assimilation system. Therefore, the computational cost of our volcanic ash assimilation system will be first analyzed. Then, based on the computational analysis, we will investigate a strategy to accelerate the ensemble-based data assimilation system for volcanic ash forecasts.

This paper is organized as follows. Section 2 introduces the methodology of volcanic ash data assimilation. Section 3 analyzes the computational cost of the conventional volcanic ash data assimilation system. In Section 4, the mask-state algorithm is developed for acceleration. The discussions on the mask-state algorithm is in Section 5. Finally, the last section summarizes the concluding remarks of our research.

2 Methodology of the volcanic ash data assimilation system

In this study, the ensemble Kalman filter (EnKF) (Evensen, 1994) is employed to perform ensemble-based data assimilation. EnKF is typically a sequential Monte Carlo method (Evensen, 2003), according to the uncertain state estimate with N ensemble members, $\xi_1, \xi_2, \cdots, \xi_N$. Each member is assumed as one sample in the distribution of the true state. It has been proposed that for operational applications, the ensemble size can be limited to 10 – 100 for cost effectiveness (Nerger and Hiller, 2013; Barbu et al., 2009). Thus, in this study, an ensemble size of 100 is used due to the high-accuracy requirement of the volcanic ash forecasts to aviation advices as mentioned in Section 1.

To simulate a volcanic ash plume, an atmospheric transport model is needed. In this paper, the LOTOS-EUROS (abbreviation of LOng Term Ozone Simulation – EURopean Operational Smog) model is used (Schaap et al., 2008) with model version 1.10 (*http://www.lotos-euros.nl/*). The LOTOS-EUROS model (Schaap et al., 2008) is an operational model focusing on nitrogen oxides, ozone, particular matter, volcanic ash. The model configurations for volcanic ash were discussed in details by Fu et al. (2016). For volcanic ash simulation, the model is configured with a state vector of size $180\times200\times18\times6$ (the dimensions correspond to longitude, latitude, vertical level, ash species), the size of model state is thus calculated as $\sim 10^6$.

The experiment in this study starts at t_0 (09:00 UTC, 18 May, 2010 for this study) by considering an initial condition from previous LOTOS-EUROS conventional model run (see Fig. 1**a**). In the second step (the forecast step) the model propagates the

ensemble members from the time t_{k-1} to t_k ($k > 0$, the time step is 10 minutes):

$$\xi_j^f(k) \quad = \quad M_{k-1}(\xi_j^a(k-1)). \tag{1}$$

The operator M_{k-1} describes the time evolution of the state which contains the ash concentrations in all model grid-boxes. The state at the time t_k has a distribution with the mean $\mathbf{x}^f$ and the forecast error covariance matrix $\mathbf{P}^f$ given by:

$$\mathbf{x}^f(k) = [\sum_{j=1}^{N} \xi_j^f(k)]/N, \tag{2}$$

$$\mathbf{L}^f(k) = [\xi_1^f(k) - \mathbf{x}^f(k), \cdots, \xi_q^f(k) - \mathbf{x}^f(k)], \tag{3}$$

$$\mathbf{P}^f(k) = [\mathbf{L}^f(k)\mathbf{L}^f(k)^T]/(N-1), \tag{4}$$

where $\mathbf{L}^f$ represents the ensemble perturbation matrix. In this study, the forecast step is performed in parallel because of the natural/common parallelism of the independent ensemble propagation, which is a trivial approach when employing ensemble-based data assimilation (Liang et al., 2009; Tavakoli et al., 2013; Khairullah et al., 2013).

When the model propagates to 09:40 UTC, 18 May, 2010, the volcanic ash state gets sequentially analyzed by the data assimilation process through combining real aircraft in situ measurements of PM_{10} and $PM_{2.5}$ concentrations until 11:10 UTC. The measurement route and values are demonstrated in Fig. 1**b**– 1**c** and the details are described in (Weber et al., 2012; Fu et al., 2016). The observational network at the time t_k is defined by the operator H_k which maps the state vector $\mathbf{x}$ to the observational space $\mathbf{y}$ by

$$\mathbf{y}(k) = H_k(\mathbf{x}(k)) + \mathbf{v}(k), \tag{5}$$

where $\mathbf{y}$ contains the aircraft measurements and $\mathbf{v}$ represents the observational error. H_k selects the grid cell in $\mathbf{x}(k)$ that corresponds to the locations of the observation. When measurements are available, the ensemble members are updated in the analysis step using

$$\mathbf{K}(k) = \mathbf{P}^f(k)\mathbf{H}(k)^T[\mathbf{H}(k)\mathbf{P}^f(k)\mathbf{H}(k)^T + \mathbf{R}]^{-1}, \tag{6}$$

$$\xi_j^a(k) = \xi_j^f(k) + \mathbf{K}(k)[\mathbf{y}(k) - \mathbf{H}(k)\xi_j^f(k) + \mathbf{v}_j(k)], \tag{7}$$

where $\mathbf{K}$ represents the Kalman gain, $\mathbf{R}$ represents the measurement error covariance matrix and $\mathbf{v}_j$ represents the realization out of the observation error distribution $\mathbf{v}$. After the continuous assimilation ending at 11:10 UTC, the forecast at 12:00 UTC is illustrated in Fig. 1**d**, for which the forecast accuracy has been carefully evaluated as significantly improved compared to the case without assimilation (Fu et al., 2016).

The EnKF with above setups is abbreviated as "conventional EnKF" and used in this study for the computational evaluation. Note that in the study we don't use covariance localization as proposed by Hamill et al. (2001) for reducing spurious covariances. This is because although localization is possible, the ideal case is not to use it in order to have the correct covariances in a large (converged) ensemble. It is crucial for localization that when unphysical (spurious) covariances are eliminated, physical (correct) covariances can be well maintained (Petrie and Dance, 2010). If the "filtering length scale" for localization is too

long (i.e., all the dynamical covariances are allowed), many of the spurious covariances may not be eliminated. If the length is too short, important physical dynamical covariances then may be lost together with the spurious ones. Therefore, essentially deciding an accurate localization is a challenging subject (Riishojgaard, 1998; Kalnay et al., 2012) especially for accuracy-demanding applications. Therefore, in this study we choose the ensemble size of 100 to guarantee the accuracy and avoid large spurious covariances.

3 Computational analysis for volcanic ash data assimilation

3.1 Computational analysis of the total runtime

Ensemble-based data assimilation is a useful approach to improve the forecast accuracy of volcanic ash transport. However, if it is time-consuming, it cannot be taken as efficient due to the high requirement on speed for volcanic ash assimilation (see Section 1). Based on this consideration, we need to analyze the computational cost of a conventional volcanic ash assimilation system.

As introduced in Section 2, the total execution time of conventional EnKF comprises four parts, i.e., initialization, forecast, analysis and other computational cost. The initialization time includes reading meteorological data, initializing model geographical and grid configurations, reading emission information, initializing stochastic observer for reading and transforming observations to the model grid, initializing all the ensemble states and ensemble mean, and so on. The forecast time is obtained from Eq. (1), while the analysis time corresponds to the computational sum from Eq. (2) to (7). The other computational time includes script compiling, setting environment variables, starting and finalizing data assimilation algorithms, etc.

The evaluation result of the conventional EnKF is shown in Table 1 (the middle column). It can be seen that the total computational time (4.36 h) is relatively large compared to the simulation window (3.0 h, i.e., from 9:00–12:00 UTC, 18 May, 2010), which is too much in an operational sense. Therefore, in this study, we aim to accelerate the computation to within an acceptable runtime (i.e., requires less runtime than the time period of the data assimilation application).

It can be also observed from Table 1 that the main contribution to the total execution time is the analysis step. Compared to the initialization and forecast time, the analysis stage takes 72% of the total runtime. Due to the expensive analysis step, although some approaches (such as MPI-parallel I/O (Filgueira et al., 2014), domain decomposition (Segers, 2002)) can potentially accelerate the initialization and forecast step, the effect to the final acceleration of the total computational cost is little. Therefore, to get acceptable computational time, the cost reduction in the analysis step is the target. One may wonder that since the amount of observations is small, why does analysis takes so much time? The large state vector seems to be left responsible for the problem. To know the exact reason, the detailed computational cost of the analysis step must be evaluated.

3.2 Cost estimation of all analysis procedures

We start with the formulations of the analysis step. The analysis step is represented by Eq. (7), which can be written in a full matrix format with Eq. (8),

$$\mathbf{A}^a_{n\times N} = \mathbf{A}^f_{n\times N} + \mathbf{K}_{n\times m}(\mathbf{Y}_{m\times N} - \mathbf{H}_{m\times n}\mathbf{A}^f_{n\times N}), \tag{8}$$

where the subscripts represent the matrix's dimensions. $\mathbf{A}^f$ and $\mathbf{A}^a$ represent the forecasted and analyzed ensemble state matrix, and are respectively built up from ξ^f and ξ^a with N ensembles. The measurement ensemble matrix $\mathbf{Y}$ is formed by an ensemble of $\mathbf{y}+\mathbf{v}$ (see Eq. (7)). $\mathbf{H}$ is the observational matrix, which is used to select state variables (at measurement locations) in the full ensemble state matrix corresponding to the measurement ensemble matrix $\mathbf{Y}$. n is the number of model state variables in a three-dimensional (3D) domain, i.e., $\sim 10^6$ in this study (see Section 2). m is the amount of measurements at one assimilation time, which depends on the measurement type. For aircraft in situ measurements used in this study (see Fig. 1**c**), two measurements are made at each time by one research flight, so that m is 2 here. N is the ensemble size and is taken as 100 in this study. As described in Eq. (3), the ensemble perturbation matrix $\mathbf{L}^f$ in EnKF can be re-written as

$$\mathbf{L}^f_{n\times N} = \mathbf{A}^f_{n\times N} - \bar{\mathbf{A}}^f_{n\times N} = \mathbf{A}^f_{n\times N}(\mathbf{I}_{N\times N} - \frac{1}{N}\mathbf{1}_{N\times N}) = \mathbf{A}^f_{n\times N}\mathbf{B}_{N\times N}, \tag{9}$$

where $\mathbf{I}$ is an $N\times N$ unit matrix and $\mathbf{1}$ is an $N\times N$ matrix with all elements equal to 1. Thus, $\mathbf{L}^f = \mathbf{A}^f\mathbf{B}$ where $\mathbf{B}_{N\times N}$ is introduced to represent $(\mathbf{I}_{N\times N} - \frac{1}{N}\mathbf{1}_{N\times N})$. So that, $\mathbf{HL}^f = \mathbf{O}^f\mathbf{B}$, where $\mathbf{O}^f_{m\times N}$ is used to represent $(\mathbf{HA}^f)$. Here we explicitly express $\mathbf{L}^f$ and $\mathbf{HL}^f$ in the form of $\mathbf{A}^f$ and $\mathbf{O}^f$, respectively. This is because in our volcanic ash assimilation system, $\mathbf{A}^f$ and $\mathbf{O}^f$ are two of the three inputs (another one is the measurement ensemble matrix $\mathbf{Y}$ for the analysis step. These are the three inputs used for actual computations in the analysis step. As shown in Fig. 2a, $\mathbf{A}^f$ is obtained from the forecast step, $\mathbf{O}^f$ and $\mathbf{Y}$ are acquired from our stochastic observer module (see Fig. 2a) which is used for a volcanic ash transport model to integrate geophysical measurements. With the input $\mathbf{Y}$, the measurement error covariance $\mathbf{R}$, as introduced in Eq. (6), can be then computed with

$$\mathbf{R}_{m\times m} = \frac{1}{N-1}(\mathbf{Y}_{m\times N} - \bar{\mathbf{Y}}_{m\times N})(\mathbf{Y}_{m\times N} - \bar{\mathbf{Y}}_{m\times N})' = \frac{1}{N-1}(\mathbf{YB})(\mathbf{YB})'. \tag{10}$$

Based on previous definitions and Eq. (2) to (7), the analysis step can be reformulated as followings,

$$\begin{aligned}
\mathbf{A}^a_{n\times N} &= \mathbf{A}^f + \mathbf{K}(\mathbf{Y} - \mathbf{HA}^f) \\
&= \mathbf{A}^f + \mathbf{P}^f\mathbf{H}'(\mathbf{HP}^f\mathbf{H}' + \mathbf{R})^{-1}(\mathbf{Y} - \mathbf{HA}^f) \\
&= \mathbf{A}^f + \frac{1}{N-1}\mathbf{L}^f(\mathbf{HL}^f)'[\frac{1}{N-1}(\mathbf{HL}^f)(\mathbf{HL}^f)' + \frac{1}{N-1}(\mathbf{YB})(\mathbf{YB})']^{-1}(\mathbf{Y} - \mathbf{HA}^f) \\
&= \mathbf{A}^f + \mathbf{A}^f\mathbf{B}(\mathbf{O}^f\mathbf{B})'[(\mathbf{O}^f\mathbf{B})(\mathbf{O}^f\mathbf{B})' + (\mathbf{YB})(\mathbf{YB})']^{-1}(\mathbf{Y} - \mathbf{O}^f) \\
&= \mathbf{A}^f\{\mathbf{I} + \mathbf{B}(\mathbf{O}^f\mathbf{B})'[(\mathbf{O}^f\mathbf{B})(\mathbf{O}^f\mathbf{B})' + (\mathbf{YB})(\mathbf{YB})']^{-1}(\mathbf{Y} - \mathbf{O}^f)\} \\
&= \mathbf{A}^f_{n\times N}\mathbf{X}_{N\times N} \quad ,
\end{aligned} \tag{11}$$

where

$$\mathbf{X}_{N\times N} = \{\mathbf{I} + \mathbf{B}(\mathbf{O}^f\mathbf{B})'[(\mathbf{O}^f\mathbf{B})(\mathbf{O}^f\mathbf{B})' + (\mathbf{YB})(\mathbf{YB})']^{-1}(\mathbf{Y} - \mathbf{O}^f)\}. \tag{12}$$

Eq. (11) shows how the analysis step is performed in a volcanic ash assimilation system. In order to accelerate the analysis step, the most time-consuming part must be reduced. Fig. 2b shows estimations of the computational cost for each procedure in the analysis step. Considering that the state number n ($\sim 10^6$) is significantly larger than the measurement number m (m=2 here) and the ensemble size N (N=100), thus the most time-consuming procedure in the analysis step is the last one, that is $\mathbf{A^a} = \mathbf{A^f X}$ with computational cost of O(nN^2). Therefore, in our volcanic ash assimilation system, this part is the most time-consuming part in the analysis step. Note that the procedure $[(\mathbf{O}^f\mathbf{B})(\mathbf{O}^f\mathbf{B})' + (\mathbf{YB})(\mathbf{YB})']^{-1}$ for Singular Value Decomposition (SVD) in our study is not time-consuming, which is quite different from some other applications, such as reservoir history matching (Tavakoli et al., 2013; Khairullah et al., 2013). This is because the SVD procedure costs O(m^3), and due to the measurement size in the order of the size of the state in those cases, SVD procedure thus requires a huge computational cost for reservoir assimilation.

4 The mask-state algorithm for acceleration of the analysis step

4.1 Characteristic of ensemble state matrix $\mathbf{A^f}$

Analysis in the previous section shows that $\mathbf{A^a} = \mathbf{A^f X}$ is most expensive in the analysis step. Each column of $\mathbf{A^f}$ is constructed from a forecasted ensemble state, thus the dimension of $\mathbf{A^f}$ is $n \times N$. In each column, the element values correspond to volcanic ash concentrations in a 3D domain. Fig. 3 shows the coverage of all ensemble forecast states at a selected time 10:00 UTC 18 May, 2010, without loss of generality. A common phenomenon can be observed, that is only a part of the 3D domain are filled with volcanic ash. The ash clouds only concentrate in a plume which is transported over time. This is because volcanic eruption is a fast and strong process. The advection dominates the transport, and the volcanic ash plume is transported with the wind. This is a particular characteristic for volcanic ash transport, in contrast to other atmospheric related applications such as ozone (Curier et al., 2012), SO_2 (Barbu et al., 2009), CO_2 (Chatterjee et al., 2012). For those applications, the concentrations are everywhere in the domain, the emission sources are also everywhere, and observations are available throughout the domain too (especially for satellite data). Whereas for application of volcanic ash transport, the source emission is only at the volcano, thus usually only a limited domain is polluted by ash. As shown in Fig. 3, in the 3D domain with grid size of 3.888×10^6, the number of grids in the area with volcanic ash is counted as 1.528×10^6, whereas the number of no-ash grids is 2.36×10^6. Note that shown in the figure are accumulated ash coverages of all ensemble states, thus in the no-ash grids, there are no ash for all the ensemble states. Thus a very large number of rows in $\mathbf{A^f}$ are zero corresponding to the no-ash grids. These zero rows in $\mathbf{A^f}$ have no contributions to $\mathbf{A^a} = \mathbf{A^f X}$, because a zero row in $\mathbf{A^f}$ always results in a zero row in $\mathbf{A^a}$. Therefore, for the case of Fig. 3, $\frac{2}{3}$ of the computations are redundant and can be avoided. To realize this, one may think to limit the domain for the entire assimilation steps, then the number of zero rows certainly would be largely reduced. This is actually incorrect, because

these zero rows are changing along with the transport of ash clouds, and not constant at each analysis step. So the full domain must be considered and it should be adaptive (choose different zero rows according to different $\mathbf{A^f}$ at different analysis time).

4.2 Derivation of the mask-state algorithm (MS)

Here we introduce item n_{noash} to represent the number of zero rows in the ensemble state matrix $\mathbf{A^f}$, and use n_{ash} to represent the number of other rows (also n_{ash} represents the grid size of ash plume). When computing $\mathbf{A^a} = \mathbf{A^f X}$, to avoid all the computations related to n_{noash} rows with zero elements, the index of other n_{ash} rows must be first decided. This index is meant to reduce the dimensions of $\mathbf{A^f}$. After getting a $\mathbf{A^a}$ with a dimension of $n_{ash} \times N$, the index will be used again to reconstruct the full matrix $\mathbf{A^a}$ with the dimension of $n \times N$. Based on this idea, we propose a mask-state algorithm (MS) which deals with the time-consuming analysis update. MS includes five steps:

(i) **Compute ensemble mean state $\bar{\mathbf{A}}^\mathbf{f}$**: The mean state $\bar{\mathbf{A}}^\mathbf{f}{}_{n\times 1}$ can be easily computed by averaging $\mathbf{A^f}_{n\times N}$ along N columns. Due to all elements in $\mathbf{A^f}_{n\times N}$ corresponding to ash concentrations, thus all elements in $\mathbf{A^f}_{n\times N}$ are larger than zero, so that the index of non-zero rows in $\bar{\mathbf{A}}^\mathbf{f}{}_{n\times 1}$ is equivalent to that in $\mathbf{A^f}_{n\times N}$. The computational cost for this step is O(nN).

(ii) **Construct mask array $\mathbf{z}$**: Based on previously obtained $\bar{\mathbf{A}}^\mathbf{f}{}_{n\times 1}$, we search the non-zero elements of $\bar{\mathbf{A}}^\mathbf{f}{}_{n\times 1}$ and record the index into a mask array $\mathbf{z}_{n_{ash}\times 1}$. With this strategy, we don't need to search the full matrix $\mathbf{A^f}_{n\times N}$ and build an index matrix for storage. This is a benefit for saving memory. The computational cost for this step is O(n).

(iii) **Construct masked ensemble state matrix $\tilde{\mathbf{A}}^\mathbf{f}$**: Using the mask array $\mathbf{z}_{n_{ash}\times 1}$ obtained from step (ii), $\tilde{\mathbf{A}}^\mathbf{f}{}_{n_{ash}\times N}$ can be constructed column by column according to Eq. (13), and the computational cost (overhead) for this step is O($n_{ash}N$).

$$\tilde{\mathbf{A}}^\mathbf{f}(1:n_{ash}, 1:N) = \mathbf{A^f}(\mathbf{z}(1:n_{ash}), 1:N), \tag{13}$$

(iv) **Compute $\tilde{\mathbf{A}}^\mathbf{a}$ by multiplying $\tilde{\mathbf{A}}^\mathbf{f}$ and $\mathbf{X}$**: Perform matrix computation $\tilde{\mathbf{A}}^\mathbf{a}{}_{n_{ash}\times N} = \tilde{\mathbf{A}}^\mathbf{f}{}_{n_{ash}\times N}\mathbf{X}_{N\times N}$. This step is similar to $\mathbf{A^a} = \mathbf{A^f X}$, as described in Section 3.2, but the computational cost now becomes O($n_{ash}N^2$) instead of O(nN^2).

(v) **Construct analyzed ensemble state matrix $\mathbf{A^a}$**: With the computed $\tilde{\mathbf{A}}^\mathbf{a}$ from step (iv) and the mask array $\mathbf{z}$ from step (ii), the final analyzed ensemble state matrix $\mathbf{A^a}_{n\times N}$ can be constructed based on Eq. (14). The computational cost (overhead) for this step is O(nN).

$$\mathbf{A^a}(\mathbf{z}(1:n_{ash}), 1:N) = \tilde{\mathbf{A}}^\mathbf{a}(1:n_{ash}, 1:N), \tag{14}$$

According to the derivations of MS, the computational cost related to zero rows are avoided. Here the "zero rows" doesn't equal to "zero elements". The former corresponds to the regions where there are no ash for all the ensemble members, while the latter also counts the no-ash regions specifically for some ensembles. Certainly the consideration of all "zero elements"

can include all the sparsity information of the ensemble state matrix, but extra computations and memories must be spent on searching the full matrix $\mathbf{A^f}_{n\times N}$ with a computational cost of O(nN) and storing a mask state matrix with dimensions of $n \times N$. This is expensive compared to construct the mask array in the procedure (ii). Actually, after a careful check on the volcanic ash ensemble plumes, there is no "bad" ensemble which is really different from others. Although the concentration level in ensemble members are distinct, the main direction and the occurrence to the grid cells are more or less same. This means, the "zero rows" actually more or less equals to "zero elements", but much faster than the way with "zero elements", which confirms the suitability and advantage of procedure (ii). Probably when there are big meteorological uncertainties, the "zero elements" will be much larger than "zero rows". In this case, how to make use of the sparsity information in the ensemble state matrix, will be considered in future.

Based on procedures of MS, the computational cost of $\mathbf{A^a} = \mathbf{A^f X}$ can be reduced. However, without a careful evaluation, we cannot conclude MS is fast, because the algorithm also employs other procedures. If these procedures (i)(ii) (iii)(v) are much cheaper than the main procedure (iv), MS can definitely speed up the analysis step, and vice versa. Now we analyze MS's computational cost, which can be summed as O(nN)+ O(n)+ O($n_{ash}N$)+ O($n_{ash}N^2$)+ O(nN), i.e., O($nN + n_{ash}N^2$). Thus, the computational overhead involved to transform the full matrix to a small one (i.e., O($n_{ash}N$) for procedure (iii)) has little effect in the total computation cost of MS (i.e., O($nN + n_{ash}N^2$)). However, the computational overhead of transforming the small matrix to the full one (i.e., O(nN) for procedure (v)) does contribute a part, which cannot be ignored, to the total MS's computational cost. The computational cost without MS is O(nN^2).

The comparison between both cost (with and without MS, i.e., O($nN + n_{ash}N^2$) and O(nN)) indicates when the number of non-zero rows (n_{ash}, i.e., the number of grids with ash) of the forecasted ensemble state matrix satisfies $n_{ash} < \frac{N-1}{N}n$, then MS can accelerate $\mathbf{A^a} = \mathbf{A^f X}$. Here, O($nN + n_{ash}N^2$) and O($nN$) are of the same order when $n_{ash} < \frac{N-1}{N}n$. The larger the difference between n_{ash} and $\frac{N-1}{N}n$, the better the speedup can be achieved. According to this analysis, and the characteristic (e.g., $\frac{n_{ash}}{n}$ approximately equals to $\frac{1}{3}$ in this case) of volcanic ash transport as described in Section 4.1, the relation is certainly satisfied and is actually $n_{ash} \ll \frac{N-1}{N}n$ (significantly smaller) for our study. Therefore, for our volcanic ash assimilation system, with MS, the computational cost for the time-consuming part $\mathbf{A^a} = \mathbf{A^f X}$ is O($n_{ash}N^2$), which is much reduced compared to O(nN^2) with conventional computations.

The relation $n_{ash} < \frac{N-1}{N}n$ indicates whether we would have speedup by the MS method, actually it can be extended to Eq. (15),

$$S_{\mathbf{ms}} = \frac{\mathbf{O}(nN^2)}{\mathbf{O}(nN + n_{ash}N^2)} = \mathbf{O}(\frac{n}{n_{ash}}), \tag{15}$$

which explicitly specifies the expected amount of speedup ($S_{\mathbf{ms}}$) of $\mathbf{A^a} = \mathbf{A^f X}$ by the MS algorithm. In this case study, N is taken at 100 and $\frac{n_{ash}}{n} \approx \frac{1}{3}$, so $S_{\mathbf{ms}}$ is approximately 3.0.

According to Amdahl's law (Amdahl, 1967), the total computational speedup ($S_{\mathbf{total}}$) by MS can be predicted by Eq. (16),

$$S_{\mathbf{total}} = \frac{1}{(1 - p_{\mathbf{ms}}) + \frac{p_{\mathbf{ms}}}{S_{\mathbf{ms}}}}, \tag{16}$$

where $p_{\mathbf{ms}}$ is the proportion of the computational cost of $\mathbf{A^a} = \mathbf{A^f X}$ in the overall data assimilation computations. It has been evaluated that the computational cost of $\mathbf{A^a} = \mathbf{A^f X}$ dominates the analysis step (see Fig. 2**b**), thus the proportion of the computational cost of $\mathbf{A^a} = \mathbf{A^f X}$ approximates the proportion of the analysis step in the total data assimilation computations (i.e., $p_{\mathbf{ms}} \approx 72\%$ in this case, as described in Section 3.1). Therefore, based on Eq. (16), the maximum ("ideal") computational speedup can be predicted to be $\frac{1}{1-p_{\mathbf{ms}}}$ (i.e., ≈3.57 for this case study) when $S_{\mathbf{ms}}$ approximates infinity. However, this is not the actual speedup because $S_{\mathbf{ms}}$ is in fact specified by Eq. (15). (Based on discussions above, $S_{\mathbf{total}}$ can be therefore estimated by Eq. (15) at ≈2.0 in this case.)

4.3 Experimental results

Analysis of the algorithmic complexity of the mask-state algorithm (MS) shows MS is an efficient approach to reduce the computational cost of the time-consuming $\mathbf{A^a} = \mathbf{A^f X}$. Now MS will be applied in the real volcanic ash assimilation system, to investigate whether in practice it can well speed up the analysis step. We perform MS in the conventional EnKF, which means initialization, forecast steps are all computed as the conventional EnKF. The only difference between MS-EnKF and conventional EnKF is that in the former MS is employed for analysis step, and in the latter is standard analysis step. The result and related specifications are shown in Table 1. As introduced in Section 2, the forecast step has been configured with the conventional parallelization, thus N+2 (102 here) cores are actually used (one core for the data assimilation algorithm, the other N+1 cores for the parallel forecast of N ensemble members and one ensemble mean). It can be seen that MS indeed largely accelerates the analysis step (as expected, by a factor of about 3.0 for this study) which confirms the theoretical cost evaluation. The mask-state algorithm is now experimentally proven as efficient to significantly reduce the computational time for the analysis step during volcanic ash assimilation.

Note that it can also be observed that the computational time for the "other" parts in Table 1 (such as operations for setting environmental variables, starting and finalizing data assimilation algorithms, as mentioned in Section 3.1) is slightly reduced by the MS method (i.e., 0.03 h in this case). This is because in the conventional EnKF, the ensemble mean state $\bar{\mathbf{A}}^{\mathbf{f}}$ is calculated in the "other" parts as an output to finalize the data assimilation algorithms, while in MS-EnKF, the calculations of $\bar{\mathbf{A}}^{\mathbf{f}}$ are needed and directly involved in the "Analysis" part.

The result shows that benefiting from the success of reduced analysis step, the overall computational cost indeed gets significantly reduced. The total execution time is 1.95 h which is less than the simulation window of 3 h (09:00 – 12:00 UTC, May 18, 2010). This result satisfies our goal to accelerate the computation to an acceptable runtime (i.e., requires less run time than the time period of the data assimilation application). Therefore, aviation advices based on the MS-EnKF can be provided as not only accurate, but also sufficiently fast. Note that the result (1.95 h) is obtained after the volcanic ash is transported to the continental Europe. If the assimilation is performed in the starting phase of volcanic ash eruption (when aircraft measurements are available), a more significant acceleration would be obtained. This is because in this case the volcanic ash is only transported in an area near to the volcano, thus the number of no-ash grid cells will take a large proportion (much higher than $\frac{2}{3}$ for this case study) of the full domain.

Another note is that in this study, we only perform the commonly used ensemble parallelization for the forecast step (already efficient compared to the expensive analysis step), but do not choose model-based parallelization (e.g., tracer or domain decomposition). As specified in Table 1, no parallelization is implemented on the 6 tracers. This is because due to the important aggregation process (Folch et al., 2010), there are big dependencies between different ash components and thus it doesn't make much sense to parallelize them. As for domain-decomposed parallelization (Segers, 2002), it is not efficient here. This is because volcanic ash is special in the sense that the model is only doing computations in a small part of the domain (i.e., there is no data in a rather large part of domain), but this part is continuously changing. Thus, a fixed domain decomposition is not very useful here because of the changing plume position. In this sense, some advanced approach such as adaptive domain-decomposed parallelization (Lin et al., 1998) might add additional acceleration to the volcanic ash forecast stage. This is an interesting subject for future in case, when a more complicated model is employed, only ensemble parallelization may be not enough for the forecast stage.

5 Discussions

5.1 Applicability

For volcanic ash forecasts, only a relatively small domain is polluted compared to the full 3D domain, so that the mask-state algorithm (MS) can work efficiently. Using MS is also applicable for many other assimilation problems, where the domain is not fully polluted by the species. It does not matter what the emission looks like and whether the releases are short- or long-lived species. Given an assimilation problem, the only restriction for MS to gain an acceleration is whether the whole domain is fully polluted or partly polluted. The assimilation problems where MS can can achieve the acceleration effect on the computations of $\mathbf{A^a} = \mathbf{A^f X}$ are such as all the volcanic-related ash/gas assimilation, e.g., assimilation of satellite data/LIDAR data/in situ data; (sand/desert) dust storm related assimilation; tornado-related assimilation; assimilation of exploding nuclear plants or factories; chemicals or oils leaking on seas; global (forecest) fire assimilation; assimilation of environmental pollutant transport, e.g., severe smog.

It has been analyzed that when the number of non-zero rows (n_{ash}, i.e., the number of ash grids in a 3D domain) of $\mathbf{A^f}$ satisfies $n_{ash} < n$, MS can work faster than standard EnKF. For volcanic ash application, because n_{ash} is much less than n, the acceleration is thus quite large. Hence in this case, we propose to embed the mask-state algorithm (MS) in all ensemble-based data assimilation methods because it is fast and the implementation using MS is exact to the standard ensemble-based methods, i.e., it doesn't introduce any approximation in view of MS procedures. Actually this proposal can be extended to all real applications, even if the condition is not satisfied. This is because, in this case the computational cost of MS for $\mathbf{A^a} = \mathbf{A^f X}$ becomes O(nN^2), which is the same as that of using the standard assimilation (shown in Fig. 2b). Therefore, if the state numbers equal to or close to the number of the total number of grid points in the domain, the added computational cost by using MS is very small (neglectable), so that the computational time with MS is almost the same as the time of using the standard approach. Whereas, when the condition $n_{ash} < n$ is satisfied, MS will accelerate the analysis step. Thus MS is generic and can be directly used in any ensemble-based data assimilation, and this acceleration can be automatically realized

for some potential applications, without spending time investigating if the condition is satisfied. In a real (or operational) 3D assimilation system, MS can be easily included, i.e., we only need to invoke the MS module when computing $\mathbf{A^a} = \mathbf{A^f X}$, without any other change to the current framework.

As stated in Eq. (15), the speedup of the MS method is approximately the inverse of $\frac{n_{ash}}{n}$. So far there is no statistical data on the value of $\frac{n_{ash}}{n}$. Consider the problem of volcanic ash transport, there is only one emission point (at the volcano), all the ashes in atmospheres are transported by the directional wind drive from the same source point. Thus volcanic ash cloud is actually transported in a shape of a plume, which in general doesn't cover the full but only a small part of the 3D domain. At the start phase of a volcanic ash eruption, $\frac{n_{ash}}{n}$ is much smaller than 1.0 (started from 0). During transport over a long time (one and a half months for this case study), $\frac{n_{ash}}{n}$ increases to approximately $\frac{1}{3}$. Therefore, the speedup of MS on volcanic ash data assimilation will be significant.

5.2 MS and localization

Based on the formulation of MS, one may think it can be taken as a localization approach ((Hamill et al., 2001)). There is indeed a similarity between MS and the localization approach, in a sense that when computing $\mathbf{A^a} = \mathbf{A^f X}$, both get rid of a large number of cells, and only do computations related to the selected grids. These two algorithms are however functionally different. This is because the localization approach is meant for reducing spurious covariances outside a local region which is built up around the measurement, thus the results with and without localization approaches are different. While, MS is developed for the acceleration purpose. The masked region is discontinuous and independent of locations of measurement, but dependent on the model domain. Thus, there is no difference on the assimilation results between using MS and without using it. Therefore, based on the functional difference, MS cannot be taken as a localization approach.

In this study, we don't employ the localization strategy in the analysis step, because we use a rather large ensemble size of 100 to guarantee the accuracy, as introduced in Section 2. But for some applications (e.g., ozone, CO_2, sulfur dioxide) especially when assimilating satellite data, localization is a necessary approach and has been widely used in reducing spurious covariances (Barbu et al., 2009; Chatterjee et al., 2012; Curier et al., 2012). In these cases, because the localization approach forces the analysis only to update state within a localization region, one may think that localization could replace MS and there would be no significance to employ MS. Actually this is not correct. We explain the reason as follows.

The localization approach is usually realized in Eq. (6) by employing a Schur product of a localization matrix and the forecast error covariance matrix (Houtekamer and Mitchell, 1998, 2001) given by:

$$\mathbf{K}(k) = (\mathbf{f} \circ \mathbf{P}^f(k))\mathbf{H}(k)^T[\mathbf{H}(k)(\mathbf{f} \circ \mathbf{P}^f(k))\mathbf{H}(k)^T + \mathbf{R}]^{-1}. \tag{17}$$

The Schur product $\mathbf{f} \circ \mathbf{P}^f$ in Eq. (17) is defined by the element-wise multiplication of the covariance matrix $\mathbf{P}^f$ and a localization matrix $\mathbf{f}$. $\mathbf{f}$ is defined based on the distance between two locations, thus it is dependent on the domain and needs information of the full ensemble state locations. In this way, $\mathbf{f} \circ \mathbf{P}^f$ can contain more zeros than $\mathbf{P}^f$, but the dimensions are not changed, so that the computations related to $\mathbf{f} \circ \mathbf{P}^f$ are actually not reduced. Therefore, we can understand the localization approach in the analysis step as that the state within and outside a local region are both updated with increments, but just the

increments outside the region are zero (which seems like not updating). This is also the reason why the localization approach is not meant for acceleration but only for reducing spurious covariances. Now it is clear that localization cannot replace MS. Actually both can be performed together in dealing with the time-consuming part $\mathbf{A^a} = \mathbf{A^f X}$. The localization approach can first transfer $\mathbf{A^f}$ to a localized matrix with more zero rows. Then MS can be used to accelerate the multiplication of the localized matrix and $\mathbf{X}$. In this way, MS is expected to accelerate $\mathbf{A^a} = \mathbf{A^f X}$ with a high speedup rate, because the computational cost of more zero rows in the localized ensemble state matrix are avoided.

6 Conclusions

In this study, based on evaluations on the computational cost of volcanic ash data assimilation, the analysis step turned out to be very expensive. Although some potential approaches can accelerate the initialization and forecast step, there would be no notable improvement to the total computational time due to the dominant analysis step. Therefore, to get an acceptable computational cost, the key is to efficiently reduce the execution time of the analysis step.

After a detailed evaluation on various parts of the analysis stage, the most time-consuming part was revealed. The mask-state algorithm was developed based on a study on the characteristic of the ensemble ash states. The algorithm transforms the full ensemble state matrix into a relatively small matrix using a constructed mask array. Subsequently, the computation of the analysis step was sufficiently reduced. The mask-state algorithm is developed as a generic approach, thus it can be embedded in all ensemble-based data assimilation implementations. The extra computational cost of the algorithm is small and usually neglectable. The mask-state algorithm currently is only designed for the sequential case. Actually this approach can also be adapted for parallel implementation. This is because the related matrix multiplication can be easily parallelized on multiple processors. Optimization and evaluation on the parallelized mask-state algorithm will be considered in future.

The conventional ensemble-based data assimilation with the mask-state algorithm is shown to successfully reduce the total computational time to an acceptable level, i.e., less than the time period of the data assimilation. Consequently, timely and accurate volcanic ash forecasts can be provided for aviation advices. This approach is flexible. It boosts the performance without considering any model-based parallelization, such as domain or component decomposition. Thus, when a parallel model is available, the mask-state approach can be easily combined with the model to gain a further speedup. It implements exactly the standard data assimilation without any approximation and with easy configurations, so that it can be used to accelerate the standard data assimilation in a wide range of applications.

The use of aircraft in situ measurements is the essential reason why the mask-state algorithm perfectly works. For each analysis step, the number of measurements are quite small, and the procedure of the singular value decomposition (SVD) costs little. However, in other applications when many measurements are assimilated (e.g., satellite-based or seismic-based data), and the number of measurements is of the same order as the number of state variables, the most time-consuming part will be the SVD. In these cases, the contributions of the mask-state algorithm will be limited. The reduction of the total computing time using the mask-state algorithm therefore is less significant, an effective acceleration algorithm for the analysis step must be used and should consider the computationally-expensive SVD in the first place.

7 Data and code availability

The averaged aircraft in situ data used in this study are available from Fig. 1**c**. The used continuous aircraft data and the model output data can be accessed by request (G.Fu@tudelft.nl). The mask-state algorithm is implemented in OpenDA (the open source software for data assimilation, www.openda.com) and the software can be downloaded from sourceforge (https://sourceforge.net/projects/openda).

Author contributions. Guangliang Fu, Sha Lu and Arjo Segers simulated the volcanic ash transport using the LOTOS-EUROS model. Guangliang Fu, Hai Xiang Lin, Tongchao Lu evaluated the computational efforts. Guangliang Fu, Hai Xiang Lin, Arnold Heemink and Shiming Xu developed the mask-state algorithm. Guangliang Fu and Nils van Velzen carried out computer experiments and analyzed the performance of the mask-state algorithm in OpenDA. Guangliang Fu and Hai Xiang Lin wrote the paper.

Acknowledgements. We are very grateful to the editor and reviewers for their reviews and insightful comments. We thank the Netherlands Supercomputing Center to support us the "Cartesius" cluster for the experiments in our study. We are grateful to Professor Konradin Weber for providing the aircraft measurements.

References

Amdahl, G. M.: Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities, in: Proceedings of the April 18-20, 1967, Spring Joint Computer Conference, AFIPS '67 (Spring), pp. 483–485, ACM, New York, NY, USA, doi:10.1145/1465482.1465560, http://dl.acm.org/citation.cfm?id=1465560, 1967.

Barbu, A. L., Segers, A. J., Schaap, M., Heemink, A. W., and Builtjes, P. J. H.: A multi-component data assimilation experiment directed to sulphur dioxide and sulphate over Europe, Atmospheric Environment, 43, 1622–1631, doi:10.1016/j.atmosenv.2008.12.005, 2009.

Casadevall, T. J.: The 1989 – 1990 eruption of Redoubt Volcano, Alaska: impacts on aircraft operations, Journal of Volcanology and Geothermal Research, 62, 301–316, doi:10.1016/0377-0273(94)90038-8, 1994.

Chatterjee, A., Michalak, A. M., Anderson, J. L., Mueller, K. L., and Yadav, V.: Toward reliable ensemble Kalman filter estimates of CO2 fluxes, J. Geophys. Res., 117, D22 306+, doi:10.1029/2012jd018176, 2012.

Curier, R. L., Timmermans, R., Calabretta-Jongen, S., Eskes, H., Segers, A., Swart, D., and Schaap, M.: Improving ozone forecasts over Europe by synergistic use of the LOTOS-EUROS chemical transport model and in-situ measurements, Atmospheric Environment, 60, 217–226, doi:10.1016/j.atmosenv.2012.06.017, 2012.

Eliasson, J., Palsson, A., and Weber, K.: Monitoring ash clouds for aviation, Nature, 475, 455, doi:10.1038/475455b, 2011.

Evensen, G.: Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics, J. Geophys. Res., 99, 10 143–10 162, doi:10.1029/94jc00572, 1994.

Evensen, G.: The Ensemble Kalman Filter: theoretical formulation and practical implementation, Ocean Dynamics, 53, 343–367, doi:10.1007/s10236-003-0036-9, 2003.

Filgueira, R., Atkinson, M., Tanimura, Y., and Kojima, I.: Applying Selectively Parallel I/O Compression to Parallel Storage Systems, in: Euro-Par 2014 Parallel Processing, edited by Silva, F., Dutra, I., and Santos Costa, V., vol. 8632 of *Lecture Notes in Computer Science*, pp. 282–293, Springer International Publishing, doi:10.1007/978-3-319-09873-9_24, 2014.

Folch, A., Costa, A., Durant, A., and Macedonio, G.: A model for wet aggregation of ash particles in volcanic plumes and clouds: 2. Model application, J. Geophys. Res., 115, B09 202+, doi:10.1029/2009jb007176, 2010.

Fu, G., Heemink, A., Lu, S., Segers, A., Weber, K., and Lin, H.-X.: Model-based aviation advice on distal volcanic ash clouds by assimilating aircraft in situ measurements, Atmospheric Chemistry and Physics, 16, 9189–9200, doi:10.5194/acp-16-9189-2016, 2016.

Gudmundsson, M. T., Thordarson, T., Höskuldsson, A., Larsen, G., Björnsson, H., Prata, F. J., Oddsson, B., Magnússon, E., Högnadóttir, T., Petersen, G. N., Hayward, C. L., Stevenson, J. A., and Jónsdóttir, I.: Ash generation and distribution from the April-May 2010 eruption of Eyjafjallajökull, Iceland, Scientific Reports, 2, doi:10.1038/srep00572, 2012.

Hamill, T. M., Whitaker, J. S., and Snyder, C.: Distance-Dependent Filtering of Background Error Covariance Estimates in an Ensemble Kalman Filter, Mon. Wea. Rev., 129, 2776–2790, doi:10.1175/1520-0493(2001)129%3C2776:ddfobe%3E2.0.co;2, 2001.

Houtekamer, P. L. and Mitchell, H. L.: Data Assimilation Using an Ensemble Kalman Filter Technique, Mon. Wea. Rev., 126, 796–811, doi:10.1175/1520-0493(1998)126%3C0796:dauaek%3E2.0.co;2, 1998.

Houtekamer, P. L. and Mitchell, H. L.: A Sequential Ensemble Kalman Filter for Atmospheric Data Assimilation, Mon. Wea. Rev., 129, 123–137, doi:10.1175/1520-0493(2001)129%3C0123:asekff%3E2.0.co;2, 2001.

Houtekamer, P. L., He, B., and Mitchell, H. L.: Parallel Implementation of an Ensemble Kalman Filter, Mon. Wea. Rev., 142, 1163–1182, doi:10.1175/mwr-d-13-00011.1, 2014.

Kalnay, E., Ota, Y., Miyoshi, T., and Liu, J.: A simpler formulation of forecast sensitivity to observations: application to ensemble Kalman filters, Tellus A, 64, doi:10.3402/tellusa.v64i0.18462, 2012.

Keppenne, C. L.: Data Assimilation into a Primitive-Equation Model with a Parallel Ensemble Kalman Filter, Mon. Wea. Rev., 128, 1971–1981, doi:10.1175/1520-0493(2000)128%3C1971:daiape%3E2.0.co;2, 2000.

Keppenne, C. L. and Rienecker, M. M.: Initial Testing of a Massively Parallel Ensemble Kalman Filter with the Poseidon Isopycnal Ocean General Circulation Model, Mon. Wea. Rev., 130, 2951–2965, doi:10.1175/1520-0493(2002)130%3C2951:itoamp%3E2.0.co;2, 2002.

Khairullah, M., Lin, H., Hanea, R. G., and Heemink, A. W.: Parallelization of Ensemble Kalman Filter (EnKF) for Oil Reservoirs with Time-lapse Seismic Data, International Journal of Mathematical, Computational Science and Engineering, 7, doi:waset.org/Publication/16317, 2013.

Liang, B., Sepehrnoori, K., and Delshad, M.: An Automatic History Matching Module with Distributed and Parallel Computing, Petroleum Science and Technology, 27, 1092–1108, doi:10.1080/10916460802455962, 2009.

Lin, H.-X., Cosman, A., Heemink, A., Stijnen, J., and van Beek, P.: Parallelization of the Particle Model SIMPAR, in: Advances in Hydro-Science and Engineering, edited by Holz, K. P., Bechteler, W., Wang, S. S. Y., and Kawahara, M., vol. 3, Center for Computational Hydro-science and Engineering, https://www.researchgate.net/publication/252671025_Parallelization_of_the_Particle_Model_SIMPAR, 1998.

Miyazaki, K., Eskes, H. J., and Sudo, K.: A tropospheric chemistry reanalysis for the years 2005–2012 based on an assimilation of OMI, MLS, TES, and MOPITT satellite data, Atmospheric Chemistry and Physics, 15, 8315–8348, doi:10.5194/acp-15-8315-2015, 2015.

Nerger, L. and Hiller, W.: Software for ensemble-based data assimilation systems—Implementation strategies and scalability, Computers & Geosciences, 55, 110–118, doi:10.1016/j.cageo.2012.03.026, 2013.

Oxford-Economics: The Economic Impacts of Air Travel Restrictions Due to Volcanic Ash, Report for Airbus, Tech. rep., http://www.oxfordeconomics.com/my-oxford/projects/129051, 2010.

Petrie, R. E. and Dance, S. L.: Ensemble-based data assimilation and the localisation problem, Weather, 65, 65–69, doi:10.1002/wea.505, 2010.

Quinn, J. C. and Abarbanel, H. D. I.: Data assimilation using a GPU accelerated path integral Monte Carlo approach, Journal of Computational Physics, 230, 8168–8178, doi:10.1016/j.jcp.2011.07.015, 2011.

Riishojgaard, L. P.: A direct way of specifying flow-dependent background error correlations for meteorological analysis systems, Tellus A, 50, 42–57, doi:10.1034/j.1600-0870.1998.00004.x, 1998.

Schaap, M., Timmermans, R. M. A., Roemer, M., Boersen, G. A. C., Builtjes, P. J. H., Sauter, F. J., Velders, G. J. M., and Beck, J. P.: The LOTOS EUROS model: description, validation and latest developments, International Journal of Environment and Pollution, 32, 270+, doi:10.1504/ijep.2008.017106, 2008.

Segers, A. J.: Data Assimilation in Atmospheric Chemistry Models Using Kalman Filtering, Delft Univ Pr, http://repository.tudelft.nl/islandora/object/uuid%3A113b6229-c33a-4100-93be-22e1c8912672?collection=research, 2002.

Tavakoli, R., Pencheva, G., and Wheeler, M. F.: Multi-level Parallelization of Ensemble Kalman Filter for Reservoir History Matching, in: SPE Reservoir Simulation Symposium, Society of Petroleum Engineers, doi:10.2118/141657-ms, 2013.

Weber, K., Eliasson, J., Vogel, A., Fischer, C., Pohl, T., van Haren, G., Meier, M., Grobéty, B., and Dahmann, D.: Airborne in-situ investigations of the Eyjafjallajökull volcanic ash plume on Iceland and over north-western Germany with light aircrafts and optical particle counters, Atmospheric Environment, 48, 9–21, doi:10.1016/j.atmosenv.2011.10.030, 2012.

Zehner, C., ed.: Monitoring Volcanic Ash From Space, ESA communication Production Office, doi:10.5270/atmch-10-01, 2010.

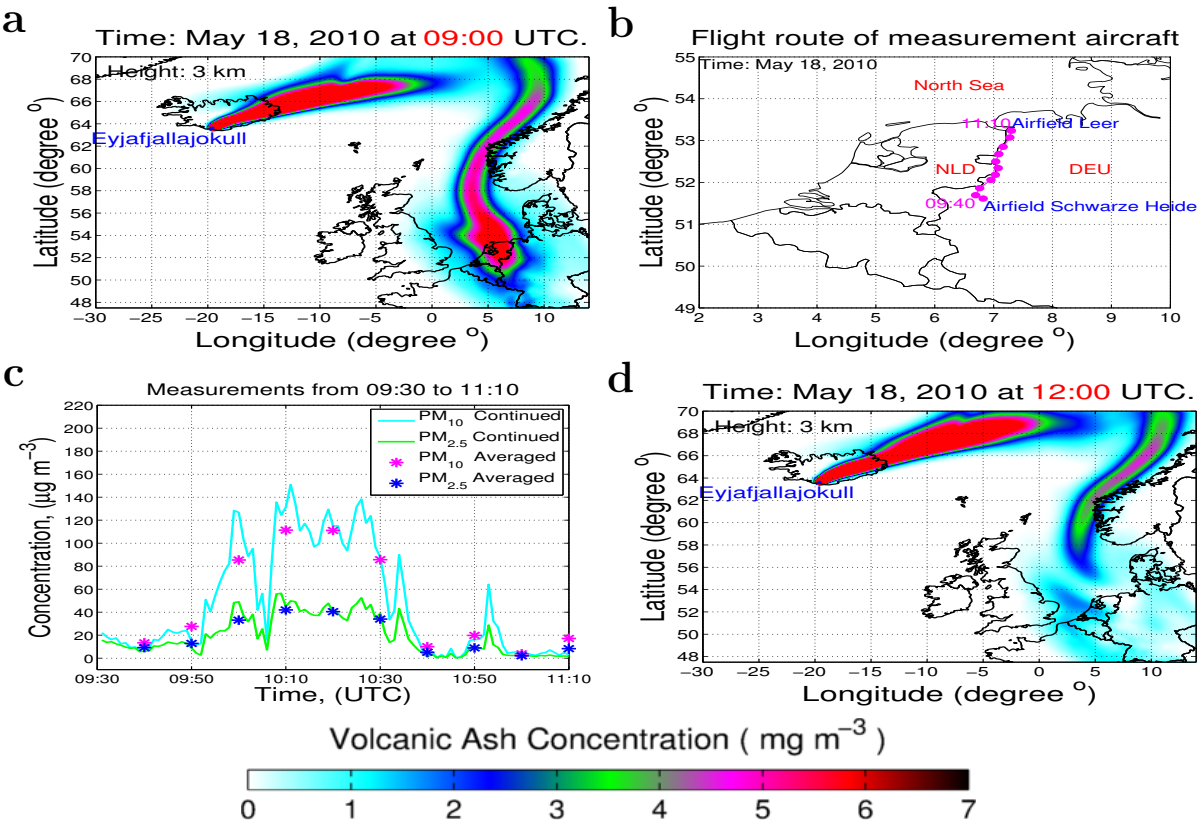
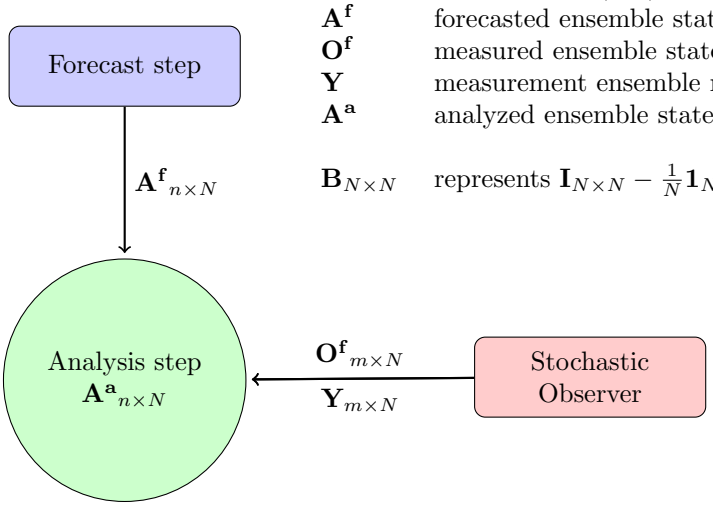


Figure 1. Methodology of ensemble-based data assimilation. a, The initial volcanic ash state at 09:00 UTC. **b**, Flight route of measurement aircraft. **c**, Aircraft in situ measurements of PM_{10} and $PM_{2.5}$ from 09:30 to 11:10 UTC May 18, 2010. **d**, Volcanic ash assimilation result at 12:00 UTC.

a



n	ash state number (3.888×10^6)
m	measurement number (2)
N	ensemble size (100)
$\mathbf{A^f}$	forecasted ensemble state matrix ($n \times N$)
$\mathbf{O^f}$	measured ensemble state matrix ($m \times N$)
$\mathbf{Y}$	measurement ensemble matrix ($m \times N$)
$\mathbf{A^a}$	analyzed ensemble state matrix ($n \times N$)
$\mathbf{B}_{N\times N}$	represents $\mathbf{I}_{N\times N} - \frac{1}{N}\mathbf{1}_{N\times N}$

b

Computational cost of analysis step

Procedures	Cost
$\mathbf{X_1} = \mathbf{O}^f\mathbf{B}$	$\mathrm{O}(mN^2)$
$\mathbf{X_2} = \mathbf{YB}$	$\mathrm{O}(mN^2)$
$\mathbf{X_3} = \mathbf{X_1}\mathbf{X_1}' + \mathbf{X_2}\mathbf{X_2}'$	$\mathrm{O}(m^2N)$
$\mathbf{X_4} = \mathbf{X_3}^{-1}$ (Singular Value Decomposition (SVD))	$\mathrm{O}(m^3)$
$\mathbf{X_5} = \mathbf{B}\mathbf{X_1}'$	$\mathrm{O}(mN^2)$
$\mathbf{X_6} = \mathbf{X_5}\mathbf{X_4}$	$\mathrm{O}(m^2N)$
$\mathbf{X} = \mathbf{I} + \mathbf{X_6}(\mathbf{Y} - \mathbf{O}^f)$	$\mathrm{O}(mN^2)$
$\mathbf{A^a} = \mathbf{A^f}\mathbf{X}$	$\mathrm{O}(nN^2)$
	(n=3.888×10^6, m=2, N=100.)

Figure 2. Computational evaluation of the analysis step. **a**, Illustration of the analysis step. **b**, Computational cost of all sub-part of the analysis step.

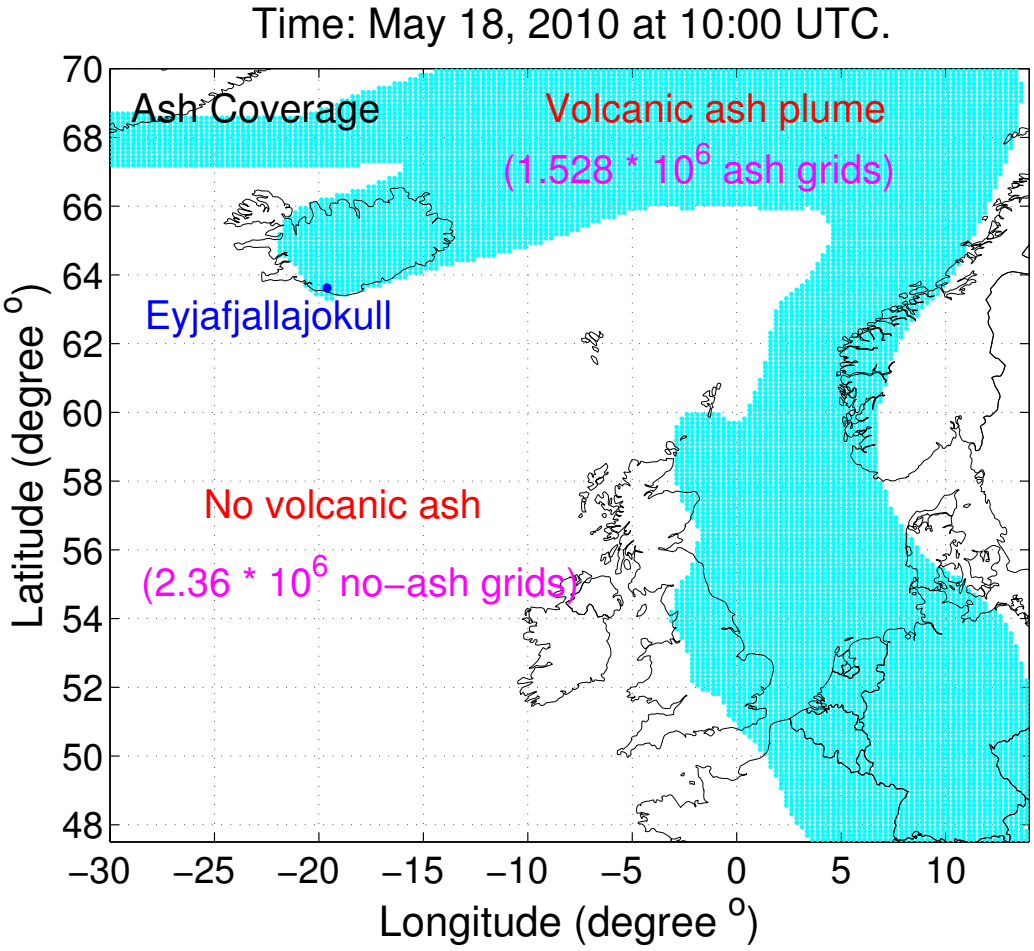


Figure 3. Characteristic of volcanic ash state.

Table 1. Comparison of the computational cost of conventional EnKF and MS-EnKF. (The results are obtained from the bullx B720 thin nodes of the Cartesius cluster, which is a computing facility of SURFsara, the Netherlands Supercomputing Centre. Each node is configured with 2 $\times$ 12-core 2.6 GHz Intel Xeon E5-2690 v3 (Haswell) CPUs and with memory 64 GB.)

Case	Conventional EnKF	MS-EnKF
Cores used	102	102
Tracer number (n_{spec})	6	6
Measurements of tracers (m)	2	2
Ensemble size (N)	100	100
Parallel in forecast step	Yes	Yes
Parallel in analysis step	No	No
Mask-state in analysis step	No	Yes
Initialization	0.42 h	0.42 h
Forecast	0.65 h	0.65 h
Analysis	3.14 h	0.88 h
Others	0.15 h	0.12 h
Total Runtime	4.36 h	1.95 h

h=hour, simulation window = 3.0 h, the time is Wall clock time.