Geosci. Model Dev., 16, 4793–4810, 2023
https://doi.org/10.5194/gmd-16-4793-2023

Geoscientific
Model Development
Open Access

CHEEREIO 1.0: a versatile and user-friendly ensemble-based chemical data assimilation and emissions inversion platform for the GEOS-Chem chemical transport model

Drew C. Pendergrass[1], **Daniel J. Jacob**[1], **Hannah Nesser**[1], **Daniel J. Varon**[1], **Melissa Sulprizio**[1], **Kazuyuki Miyazaki**[2], **and Kevin W. Bowman**[2]

[1]School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA
[2]NASA Jet Propulsion Laboratory, Pasadena, CA, USA

Correspondence: Drew C. Pendergrass (pendergrass@g.harvard.edu)

Received: 30 March 2023 – Discussion started: 19 April 2023
Revised: 10 July 2023 – Accepted: 20 July 2023 – Published: 24 August 2023

Abstract. We present a versatile, powerful, and user-friendly chemical data assimilation toolkit for simultaneously optimizing emissions and concentrations of chemical species based on atmospheric observations from satellites or suborbital platforms. The CHemistry and Emissions REanalysis Interface with Observations (CHEEREIO) exploits the GEOS-Chem chemical transport model and a localized ensemble transform Kalman filter algorithm (LETKF) to determine the Bayesian optimal (posterior) emissions and/or concentrations of a set of species based on observations and prior information using an easy-to-modify configuration file with minimal changes to the GEOS-Chem or LETKF code base. The LETKF algorithm readily allows for nonlinear chemistry and produces flow-dependent posterior error covariances from the ensemble simulation spread. The object-oriented Python-based design of CHEEREIO allows users to easily add new observation operators such as for satellites. CHEEREIO takes advantage of the Harmonized Emissions Component (HEMCO) modular structure of input data management in GEOS-Chem to update emissions from the assimilation process independently from the GEOS-Chem code. It can seamlessly support GEOS-Chem version updates and is adaptable to other chemical transport models with similar modular input data structure. A post-processing suite combines ensemble output into consolidated NetCDF files and supports a wide variety of diagnostic data and visualizations. We demonstrate CHEEREIO's capabilities with an out-of-the-box application, assimilating global methane emissions and concentrations at weekly temporal resolution and 2° × 2.5° spatial resolution for 2019 using TROPOspheric Monitoring Instrument (TROPOMI) satellite observations. CHEEREIO achieves a 50-fold improvement in computational performance compared to the equivalent analytical inversion of TROPOMI observations.

1 Introduction

Data assimilation is a field of applied mathematics that studies the most probable combination of a physical model, observational data, and prior information to define the state of a system. Many modern data assimilation algorithms have been motivated by problems in numerical weather prediction (Kalnay, 2003), and the field has more recently expanded to address problems in atmospheric chemistry (Elbern and Schmidt, 2001; Kahnert, 2008; Bocquet et al., 2015). The physical model, also called the forward model, predicts the observations on the basis of knowledge of the state of the system. For assimilation of chemical concentrations (chemical data assimilation), this forward model is a chemical transport model (CTM) that simulates the 3D fields of species concentrations by solving the corresponding continuity equations (Brasseur and Jacob, 2017). With the advent of satellite constellations measuring atmospheric composition together with increasingly dense networks of surface observations, chemical data assimilation is now commonly used to quantify emissions (Miyazaki et al., 2017; Jiang et al., 2018; Qu et al., 2019), to construct 3D concentration fields for chemical re-

Published by Copernicus Publications on behalf of the European Geosciences Union.

analyses and forecasts (Miyazaki et al., 2015, 2020; Flemming et al., 2015; Ma et al., 2019), and to diagnose CTM biases (Emili et al., 2014; Stanevich et al., 2021). The use of chemical data assimilation to quantify emissions is commonly referred to as an inversion in the atmospheric chemistry community.

Most data assimilation algorithms involve the optimization of a Bayesian scalar cost function $J(\boldsymbol{x})$ assuming Gaussian error probability density functions (PDFs) (Brasseur and Jacob, 2017).

$$J(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{x}^{\mathrm{b}})^{\mathrm{T}}(\mathbf{P}^{\mathrm{b}})^{-1}(\boldsymbol{x} - \boldsymbol{x}^{\mathrm{b}}) + (\boldsymbol{y} - H(\boldsymbol{x}))^{\mathrm{T}} \times \mathbf{R}^{-1}(\boldsymbol{y} - H(\boldsymbol{x})) \quad (1)$$

Here $\boldsymbol{x}$ is the state vector to be optimized (consisting of emissions and/or concentrations), $\boldsymbol{x}^{\mathrm{b}}$ is the initial prediction of the state vector based on prior information or a forward model forecast, $\mathbf{P}^{\mathrm{b}}$ is the prior (also called background or forecast) error covariance matrix, $\boldsymbol{y}$ is the suite of observed atmospheric concentrations arranged as a vector, $H(\cdot)$ is an observation operator that transforms the state vector $\boldsymbol{x}$ from the state space to the observation space, and $\mathbf{R}$ is the observational error covariance matrix. In the case of a state vector of emission fluxes, the observation operator $H(\cdot)$ is a CTM mapping emissions to the observed concentrations. Solving for the minimum of the cost function ($\nabla J(\boldsymbol{x}) = 0$) defines the optimized posterior (also called analysis) estimate $\boldsymbol{x}^{\mathrm{a}}$ for the state vector.

In the case in which $H(\cdot)$ is linear (i.e., representable by a matrix), an analytic solution is available with closed-form characterization of the posterior error covariance matrix (Rodgers, 2000). In nonlinear or high-dimensional linear cases, a variational approach can be used instead to iteratively minimize the cost function by numerical methods. The three-dimensional variational approach (3D-Var) calculates the gradient of the cost function for observations $\boldsymbol{y}$ in a time window sufficiently short that the time evolution of the physical system can be neglected (Asch et al., 2016). Four-dimensional variational assimilation (4D-Var) accounts for nonlinear evolution of the system over the course of an assimilation time window through use of the adjoint of the physical model, which requires construction of the tangent linear model (TLM) for the CTM; the TLM aligns the model state with observations in time while preserving the correct evolution of the physical system (Courtier et al., 1994).

Kalman filters are a general class of data assimilation systems where the time evolution of the state vector is optimized by sequential assimilation of a time series of observations in which the optimal solution at a given time step serves as a basis for the prior estimate in the next time step. Assimilation thus proceeds over successive assimilation time windows, though the Kalman filter can also be run backward (Rodgers, 2000). The original Kalman filter requires a linear forward model, but it can be combined with the TLM of the physical model to form the extended Kalman filter (EKF), which applies to nonlinear problems. The EKF has been used for atmospheric chemistry problems such as quantifying emissions of nitrogen oxides ($\mathrm{NO}_x \equiv \mathrm{NO} + \mathrm{NO}_2$) from NO_2 satellite data (Mijling and van der A, 2012; Ding et al., 2017).

Ensemble Kalman filters for chemical data assimilation, including the localized ensemble transform Kalman filter (LETKF) used in this work (Hunt et al., 2007), apply an ensemble of CTM simulations over successive assimilation time windows to approximate the prior error covariance matrix $\mathbf{P}^{\mathrm{b}}$ and its evolution over time. Like EKF and 4D-Var, LETKF can be readily applied to nonlinear problems; however, it avoids the need for a TLM because it is powered by an ensemble of CTM simulations which capture the nonlinearity of the system. Each ensemble member is initialized with random perturbations applied to emissions or concentrations of interest, and the ensemble is evolved for the assimilation time window using the CTM. At assimilation time, the ensemble spread is used to approximate the prior error covariance matrix $\mathbf{P}^{\mathrm{b}}$ and from there solve for the minimum of the cost function. The ensemble is then updated to reflect the optimized state, including emissions and concentrations, and the cycle repeats as in the case of the classic Kalman filter. Localization in LETKF means that optimization of a given state vector element is done using only observations in a localized domain of influence in order to make the problem computationally tractable. In practice, even though the state vector optimized is quite large, the ensemble can be of modest size (typically 32 or 48 members) and the LETKF will converge on the correct solution as time progresses (Hunt et al., 2007).

LETKF has been used extensively in chemical data assimilation and has benefits compared with other algorithms, most notably the ease of implementation for a wide variety of simulations. LETKF and related ensemble Kalman filter methods have been used for CO_2 flux inversions (Liu et al., 2016; Kong et al., 2022); single-species studies of NO_2, SO_2, and NH_3 emissions (Miyazaki et al., 2012a; Dai et al., 2021; van der Graaf et al., 2022); and analysis of methane emission trends (Feng et al., 2022; Zhu et al., 2022). Multi-species assimilation, 4D assimilation of temporally scattered observations, and flexibility in state vector definition are easy to implement under the LETKF framework; the algorithm also provides detailed error characterization including correlations as part of the solution. However, because ensemble methods rely on a relatively small number of simulations to simulate the problem space, the benefits of the LETKF come with issues of undersampling, which will be discussed in Sect. 2.2.

The ability of LETKF to simultaneously assimilate concentrations and emissions is of special importance to atmospheric chemists. In chemical data assimilation for operational forecasting, updates are often only applied to concentrations, but this fails to address the root issue of incorrect emissions, an especially acute problem for species with short lifetimes such as NO_x (Inness et al., 2015). On the other hand, inverse studies focused on optimizing emissions attribute all systematic discrepancies between the model and observations to emissions, even though CTM transport or observing errors may be responsible; indeed, CO_2 flux estimates calculated via inverse methods have been shown to be sensitive to transport errors (Schuh et al., 2019; 2022). Optimizing concentrations as well as emissions allows the data assimilation system to address both issues, assuming that prior error settings are posed appropriately. While this is possible to do with other algorithms, it is easy to do with LETKF due to the ability to add any additional parameter to the prior error covariance matrix and apply variable localization methods to optimize the application of observational constraints on different sets of concentrations and emissions.

Here we present the CHemistry and Emissions REanalysis Interface with Observations (CHEEREIO), a user-friendly tool that provides a platform for versatile LETKF chemical data assimilation powered by the widely used GEOS-Chem CTM. Implemented as a lightweight wrapper for GEOS-Chem, CHEEREIO gives users the ability to design and run chemical data assimilation applications without modifying model source code or learning a new code base. CHEEREIO's flexibility and simple design are enabled by the LETKF algorithm and the modular structure of GEOS-Chem, in particular its Harmonized Emissions Component (HEMCO) data input component (Keller et al., 2014; Lin et al., 2021). CHEEREIO is designed to be easily configurable for a range of applications including multi-species data assimilation, joint optimization of emissions and concentrations, and near-real-time monitoring of emissions. Coded in Python with an object-oriented framework, CHEEREIO readily accommodates new observation operators such as for new satellite instruments. CHEEREIO and all of its components are open-source, ensuring scientific transparency. CHEEREIO complements existing open-source inversion tools including the Joint Effort for Data Assimilation Integration (JEDI), a C++ and Fortran-based platform for model-generic data assimilation (Trémolet and Auligné, 2020), and PyOSSE, another Python-based platform using GEOS-Chem for observing system simulation experiments (https://www.geos.ed.ac.uk/~lfeng/, last access: 22 August 2023) (Feng et al., 2023). For atmospheric chemistry applications, CHEERIO is simpler to use than JEDI and more versatile than PyOSSE. This paper provides a high-level overview and demonstration of CHEEREIO; detailed documentation and user support are available online (cheereio.readthedocs.io).

2 CHEEREIO components

2.1 The physical model: GEOS-Chem

GEOS-Chem is a three-dimensional CTM driven by assimilated meteorological data from the Goddard Earth Observation System (GEOS) of the NASA Global Modeling and Assimilation Office (GMAO). Two alternative data sets can be used, either the GEOS Fast Processing (GEOS-FP) data at 0.25° × 0.3125° native resolution or the GEOS Modern-Era Retrospective Analysis for Research and Applications version 2 (MERRA-2) data at 0.5° × 0.625° native resolution. Both have 1 h temporal resolution for transport (archived winds) and extend from the surface to the mesopause. GEOS-Chem simulates atmospheric species concentrations by solving the coupled 3D Eulerian continuity equations on a global or user-selected nested domain at the native grid resolution of the GEOS data or at degraded resolution for computational economy. Input data files are regridded on the fly to user-specified resolution using the Harmonized Emissions Component (HEMCO) (Keller et al., 2014; Lin et al., 2021). CHEEREIO supports all GEOS-Chem applications from version 13.0.0 and later including oxidant–aerosol chemistry, aerosol only, carbon gases, and mercury, either as global or nested-grid regional simulations. GEOS-Chem High Performance (GCHP) (Eastham et al., 2018; Martin et al., 2022), which uses distributed memory rather than shared memory for parallelization, is not currently supported.

HEMCO is a critical GEOS-Chem module enabling the interface with CHEEREIO. It can apply gridded scaling factors stored in NetCDF files to any input field, such as emissions. This allows emissions updates calculated by CHEEREIO to be seamlessly loaded into GEOS-Chem without modification of source code.

2.2 The data assimilation algorithm: localized ensemble transform Kalman filter (LETKF)

The LETKF algorithm optimizes a state vector of emissions and concentrations to minimize the cost function in Eq. (1) (Hunt et al., 2007). We initialize m ensemble members at time t_o and run the forward model (GEOS-Chem) in parallel for a user-specified time (termed the assimilation window) for each of these ensemble members. Ensemble members can be thought of as a Monte Carlo sample representing the spread of atmospheric conditions resulting from our uncertainty in prior emissions; each member represents the atmospheric conditions from a random emissions perturbation sampled from a user-specified PDF. Before assimilation begins, the ensemble is run for a spin-up period to ensure the propagation of information from the perturbed emissions. Alternatively, the problem can also be set up by perturbing concentrations to represent prior uncertainty in the atmospheric state. Ensemble size is typically between 24 and 48, with the exact number to be determined by sensitivity testing, where the user identifies a size that balances error minimization with computational feasibility. Because ensemble methods randomly sample the parameter space, increasing ensemble size gradually yields diminishing return. This is unlike the analytical approach to inversions, where the number of simulations is set by the size of the state vector and diminishing return is defined by the rank of the problem (Nesser et al., 2021). In general, fewer ensemble members are required if there are fewer parameters to optimize (Miyazaki et al., 2012b; Liu et al., 2019). After the runs are complete, we construct the state vectors $\boldsymbol{x}_i^{\mathrm{b}}$ representing the concentrations and/or emissions to be optimized for each of the m ensemble members (indexed by i).

The LETKF algorithm, which we describe in the remainder of this section, is typically applied to very large state vectors, for which global optimization would be computationally prohibitive. The solution of Hunt et al. (2007) is to localize the calculation within a certain radius of the grid cell being optimized, considering only observations within that radius. Localized state vectors are formed by concatenating emissions at a given grid cell with concentrations within a given radius. Beyond reducing state vector size, this approach creates an embarrassingly parallel problem, where the cost function can be minimized independently for every localized point. The Hunt et al. (2007) localization approach also minimizes spurious correlations, which emerge in ensemble approaches due to a limited sample size; because the Monte Carlo sample is far smaller than the dimensionality of the state vector, random points will be spuriously correlated in the prior covariance matrix $\mathbf{P}^{\mathrm{b}}$ encoded by the ensemble, leading to an incorrect assimilation increment. The spurious correlation problem is especially pronounced between distant grid cells where we would expect correlations to be near zero, a problem eliminated by appropriate localization. Such localization in space is not generally useful in the analytical inversion approach, where distant correlations are set to zero and observations are ingested sequentially (Brasseur and Jacob, 2017). The precise radius used for localization should be determined by the user via sensitivity tests, considering that longer-lived species require larger localization radii; indeed, within a single inversion, multiple localization radii can be used for different components of the state vector (Miyazaki et al., 2012b). For the remainder of the equations in this section, all vectors and matrices are localized and computations are performed in parallel.

To optimize the emissions and concentrations of a given grid cell, we first construct the ensemble state vectors $\boldsymbol{x}_i^{\mathrm{b}}$ using model data. From these prior state vectors the prior perturbation matrix $\mathbf{X}^{\mathrm{b}}$ is formed from the m vector columns $\mathbf{X}_i^{\mathrm{b}}$.

$$\mathbf{X}_i^{\mathrm{b}} = \boldsymbol{x}_i^{\mathrm{b}} - \overline{\boldsymbol{x}^{\mathrm{b}}}; \overline{\boldsymbol{x}^{\mathrm{b}}} = \frac{1}{m}\sum\nolimits_{i=1}^{m}\boldsymbol{x}_i^{\mathrm{b}} \tag{2}$$

Here $\mathbf{X}_i^{\mathrm{b}}$ represents the ith column of the $n \times m$ matrix $\mathbf{X}^{\mathrm{b}}$ where n is the length of the state vector; each column of $\mathbf{X}^{\mathrm{b}}$ consists of the state vector from an ensemble member minus the mean state vector. The prior covariance matrix $\mathbf{P}^{\mathrm{b}}$ can be constructed by multiplying $\mathbf{X}^{\mathrm{b}}$ with its transpose (specifically, $\mathbf{P}^{\mathrm{b}} = (m-1)^{-1}\mathbf{X}^{\mathrm{b}}(\mathbf{X}^{\mathrm{b}})^{\mathrm{T}}$), but this is not used directly in LETKF calculations.

The model predictions made during the assimilation window must be compared to observations. Hence we construct prior vectors of simulated observations $\boldsymbol{y}_i^{\mathrm{b}}$ and a corresponding simulated observation perturbation matrix $\mathbf{Y}^{\mathrm{b}}$ formed from the m vector columns $\mathbf{Y}_i^{\mathrm{b}}$.

$$\mathbf{Y}_i^{\mathrm{b}} = \boldsymbol{y}_i^{\mathrm{b}} - \overline{\boldsymbol{y}^{\mathrm{b}}}; \quad \boldsymbol{y}_i^{\mathrm{b}} = H\left(\boldsymbol{x}_i^{\mathrm{b}}\right); \quad \overline{\boldsymbol{y}^{\mathrm{b}}} = \frac{1}{m}\sum\nolimits_{i=1}^{m}\boldsymbol{y}_i^{\mathrm{b}} \tag{3}$$

4D-LETKF, the method used in CHEEREIO, constructs $\mathbf{Y}_i^{\mathrm{b}}$ such that all simulated observations are timed to line up as closely as possible with actual observations (Hunt et al., 2007). 3D assimilation, by contrast, only aligns observations in space but uses a single model state (in particular, the state at assimilation time) to construct $\mathbf{Y}_i^{\mathrm{b}}$, leading to significant representation error. For 4D-LETKF, we load in model history files closest in time to the observation of interest and accept a modest representation error; the user can specify the time resolution via the CHEEREIO configuration file. Hence in practice we apply the operator $H(\bullet)$ to the forward model history, not to the state vector which represents the model state at a specific point in time. Methods which make use of the TLM, like 4D-Var and EKF, avoid temporal representation error due to the continuous ingestion of observations on the internal time step of the TLM, but they require major time investment in TLM development and maintenance.

Computation of the cost function in Eq. (1) involves inversion of the prior error covariance matrix $\mathbf{P}^{\mathrm{b}}$, but this is not possible in the state space (of dimension n) because by construction $\mathbf{P}^{\mathrm{b}}$ is of rank $m-1$ (the columns of the $n \times m$ matrix $\mathbf{X}^{\mathrm{b}}$ sum to the 0 vector). Hence a posterior error covariance matrix $\mathbf{P}^{\mathrm{a}}$ must be estimated in the $m-1$ dimensional subspace S spanned by the ensemble perturbations, where the inverse is well-defined. The mathematics simplify by treating $\mathbf{X}^{\mathrm{b}}$ as a linear transformation from some m-dimensional space $\tilde{S}$ to S, allowing us to redefine the cost function optimization in $\tilde{S}$ where the relevant quantities are well-behaved (Hunt et al., 2007). The posterior error covariance in $\tilde{S}$ noted with a tilde $\tilde{\mathbf{P}^{\mathrm{a}}}$ is an $m \times m$ matrix computed as follows.

$$\tilde{\mathbf{P}^{\mathrm{a}}} = \left(\frac{(m-1)\cdot I}{1+\Delta} + \gamma(\mathbf{Y}^{\mathrm{b}})^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{Y}^{\mathrm{b}}\right)^{-1} \tag{4}$$

The full derivation of $\tilde{\mathbf{P}^{\mathrm{a}}}$ is given in Hunt et al. (2007). Here $\mathbf{I}$ is the $m \times m$ identity matrix, $\mathbf{R}$ is the observational error covariance matrix, and γ is a regularization constant set by the user. $\tilde{\mathbf{P}^{\mathrm{a}}}$ plays the same role in LETKF assimilation as the posterior covariance matrix $\mathbf{P}^{\mathrm{a}}$ plays in the classical Kalman filter, connecting the state vector entries so that the calculated update is consistent with the internal correlations of the system. The regularization constant effectively scales observational errors and is designed to balance the weight given to observations within the assimilation window. Δ is an inflation factor specified by the user, usually between 0 and 0.1, which accounts for overconfidence in the assimilated ensemble; Δ does not affect the ensemble mean but it does increase the ensemble spread, with larger values pushing ensemble members away from the ensemble mean. In practice, ensemble spread can decrease with each assimilation cycle to values so small that the system is no longer able to update (nearly infinite confidence is given to the prior term, so subsequent observations carry no weight). Indeed, if γ balances the weight given to observations within the present assimilation window, Δ can be thought of as a term that balances the weight given to observations from all previous assimilation windows.

The mean posterior state vector in the original space is then given by

$$\overline{\boldsymbol{x}^{\mathrm{a}}} = \overline{\boldsymbol{x}^{\mathrm{b}}} + \gamma \mathbf{X}^{\mathrm{b}} \tilde{\mathbf{P}^{\mathrm{a}}} (\mathbf{Y}^{\mathrm{b}})^{\mathrm{T}} \mathbf{R}^{-1} \left(\boldsymbol{y} - \overline{\boldsymbol{y}^{\mathrm{b}}} \right), \tag{5}$$

where $\boldsymbol{y}$ is the vector of observations. The posterior perturbation matrix is given by

$$\mathbf{X}^{\mathrm{a}} = \mathbf{X}^{\mathrm{b}} \left((m-1) \tilde{\mathbf{P}^{\mathrm{a}}} \right)^{\frac{1}{2}}. \tag{6}$$

From here, the new ensemble state vectors can be constructed by adding $\overline{\boldsymbol{x}^{\mathrm{a}}}$ back to each column of $\mathbf{X}^{\mathrm{a}}$. The LETKF gives error characterization from the assimilation; to obtain this, we need to transform $\tilde{\mathbf{P}^{\mathrm{a}}}$ back from the space $\tilde{S}$ to the original state space. Since we defined $\tilde{S}$ with the linear transformation $\mathbf{X}^{\mathrm{b}}$, the posterior error covariance matrix is given by

$$\mathbf{P}^{\mathrm{a}} = \mathbf{X}^{\mathrm{b}} \tilde{\mathbf{P}^{\mathrm{a}}} (\mathbf{X}^{\mathrm{b}})^{\mathrm{T}}. \tag{7}$$

With the ensemble updated and errors characterized, the ensemble can be evolved using GEOS-Chem for the next assimilation window. Importantly, the ensemble is not reinitialized for these new runs; the assimilated state of the previous assimilation window becomes the initial prior state of the next assimilation window. When the runs in the new assimilation window are complete, the whole LETKF cycle begins again.

Many variations of the ensemble Kalman filter algorithms have been developed in the chemical data assimilation literature, each designed to better handle the behaviors of certain atmospheric constituents. For example, the Carbon Tracker CH_4 system handles the assimilation of long-lived CH_4 via a sliding-window approach, where surface fluxes from a given time period are estimated several times using a varying set of observations that evolve in time (Peters et al., 2005; Bruhwiler et al., 2014). Similarly, the run-in-place method changes the behavior of the assimilation window to better handle long-lived gases like CH_4 (Liu et al., 2019). With the run-in-place functionality activated, the LETKF assimilation update is calculated using a long period of observations (e.g., 1 week) but the assimilation window is advanced forward for a smaller amount of time (e.g., 1 d). Run-in-place simulations thus maintain linear growth in posterior perturbations and allow the period in which the assimilation update is calculated to experience the emissions adjustment, giving the system more time to correct assimilation errors. CHEEREIO supports many of these variations on the LETKF, as discussed in Sect. 3.

3 Description of the CHEEREIO platform

In this section, we describe the implementation of the LETKF in the CHEEREIO platform. We designed this tool to ensure maximum scientific flexibility for a diverse user base, while maintaining an abstracted interface to make the tool easy to use.

3.1 General workflow

Figure 1 shows a schematic of the CHEEREIO workflow including initialization, spin-up, sequential GEOS-Chem forward model runs and LETKF assimilation, and post-processing. Here we give a high-level overview of how CHEEREIO can be customized and deployed for any chemical data assimilation applications with GEOS-Chem. In subsequent sections, we will offer more detailed descriptions of the software design and structure, omitting technical details provided in the web documentation (https://cheereio.readthedocs.io, last access: 22 August 2023).

The CHEEREIO software package includes a suite of shell and Python scripts for assimilation, run management, observation operations, and post-processing, which can be separated into three main sequential periods in the CHEEREIO workflow, as shown in Fig. 1: initialization time, runtime, and post-processing time. Before initialization begins, users specify the simulation they would like to run via an ensemble configuration file (ens_config.json). CHEEREIO generates a template GEOS-Chem run directory based on user settings, which is copied into an ensemble of m run directories, one per ensemble member, each with a unique set of emissions perturbed according to user settings (Sect. 3.2). After the ensemble is initialized, the user submits a batch script which launches an ensemble of jobs, each running an instance of GEOS-Chem. Once the model simulation for the assimilation window is complete, CHEEREIO gathers model output files and observation data and performs the LETKF assimilation. The cycle of GEOS-Chem runs and LETKF assimilation repeats until the assimilation is complete for the entire user-specified period. Run management and LETKF implementation are discussed in Sect. 3.3. Upon completion, the user can execute a post-process workflow job to make a default set of figures, movies, and consolidated data files; they can also deploy pre-written functions to produce custom output and statistics (Sect. 3.4). In the coming sections, we expand on each of these components of the CHEEREIO workflow.

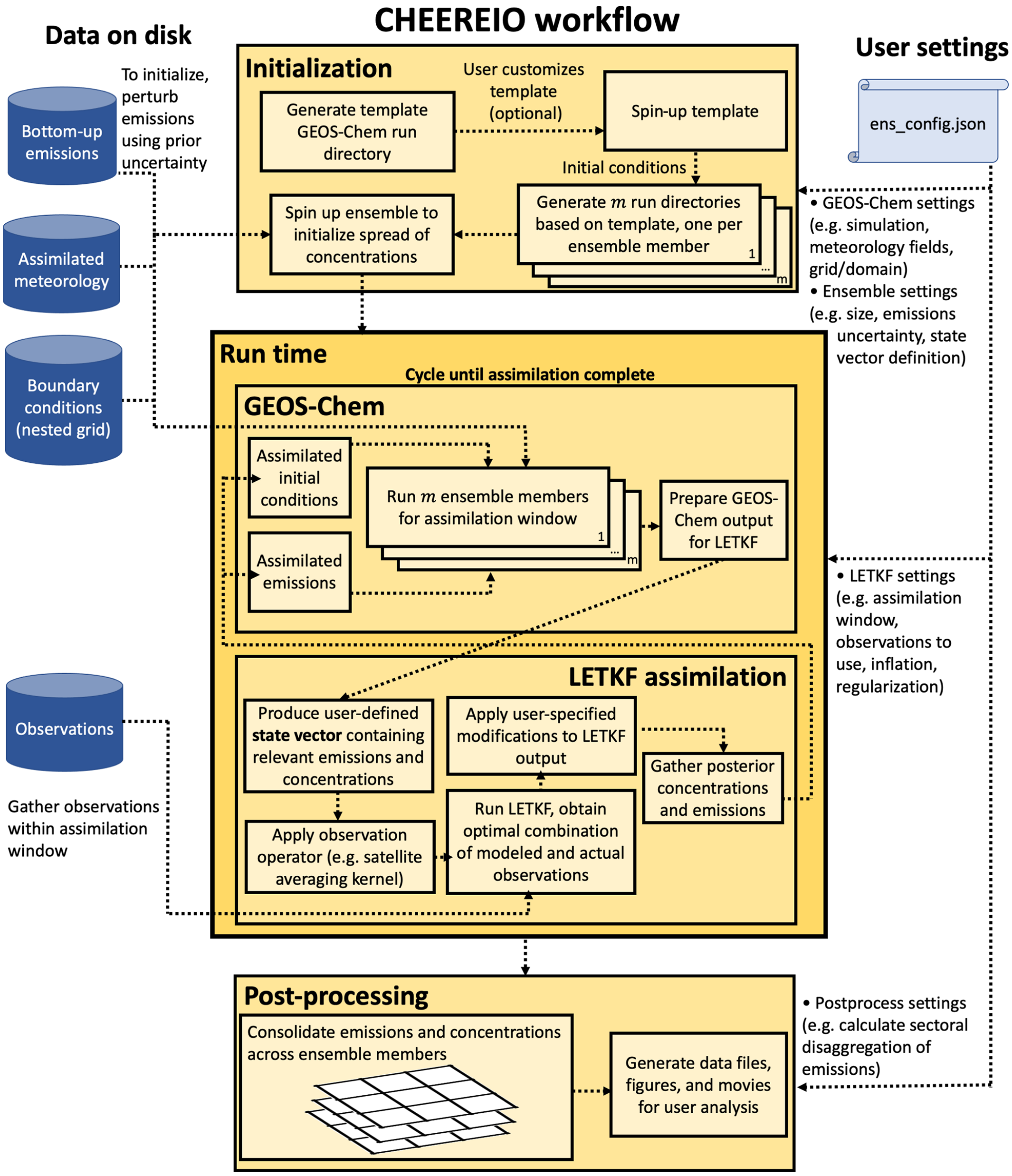
CHEEREIO workflow
Data on disk
User settings
Bottom-up emissions
To initialize, perturb emissions using prior uncertainty
Assimilated meteorology
Boundary conditions (nested grid)
Observations
Gather observations within assimilation window
ens_config.json
Initialization
Generate template GEOS-Chem run directory
User customizes template (optional)
Spin-up template
Initial conditions
Generate m run directories based on template, one per ensemble member
1
...
m
Spin up ensemble to initialize spread of concentrations
• GEOS-Chem settings (e.g. simulation, meteorology fields, grid/domain)
• Ensemble settings (e.g. size, emissions uncertainty, state vector definition)
Run time
Cycle until assimilation complete
GEOS-Chem
Assimilated initial conditions
Assimilated emissions
Run m ensemble members for assimilation window
Prepare GEOS-Chem output for LETKF
• LETKF settings (e.g. assimilation window, observations to use, inflation, regularization)
LETKF assimilation
Produce user-defined state vector containing relevant emissions and concentrations
Apply user-specified modifications to LETKF output
Gather posterior concentrations and emissions
Apply observation operator (e.g. satellite averaging kernel)
Run LETKF, obtain optimal combination of modeled and actual observations
Post-processing
Consolidate emissions and concentrations across ensemble members
Generate data files, figures, and movies for user analysis
• Postprocess settings (e.g. calculate sectoral disaggregation of emissions)

Figure 1. Schematic of the CHEEREIO workflow, divided into three steps: initialization, runtime, and post-processing. All simulations are initialized with a template GEOS-Chem run directory generated by CHEEREIO according to user-specified settings, which is then copied into an ensemble of m run directories, one per ensemble member, each with a unique set of emissions perturbed according to user settings. At runtime, GEOS-Chem simulates atmospheric concentrations reflecting these perturbed emissions for each ensemble member, which are then compared to observations to generate an assimilated suite of concentrations and emissions via the LETKF procedure. After the specified period is assimilated, CHEEREIO post-processing scripts consolidate the ensemble into a set of data files, figures, animations, and statistics for user analysis. Input data files are shown by dark blue cylinders on the left, while user settings are shown on the right.

3.2 Ensemble initialization

The CHEEREIO ensemble initialization workflow is divided into four phases, as shown in Fig. 1: (1) template run directory creation, (2) spin-up of template run directory, (3) ensemble initialization and prior emissions sampling, and (4) spin-up of the ensemble spread.

Initialization begins when the user specifies the simulation they would like to run by modifying a configuration file (ens_config.json), which includes all model and assimilation settings. Table 1 lists important parameters that can be tuned in this configuration file. LETKF results respond strongly to the localization radius (LOCALIZATION_RADIUS_km), the regularization factor γ (regular-

Table 1. Selected parameters set by the CHEEREIO configuration file.

Parameter	Description
General parameters	
sim_name	Type of GEOS-Chem simulation (oxidant–aerosol chemistry, aerosol only, carbon species, mercury)
res	Horizontal resolution
region	For nested simulations, specifications for the nested domain
start_date, end_date	Start and end dates for the assimilation
burn_in_end	Discard results prior to this date
assim_time	Length of the assimilation window
State vector settings	
state_vector_conc	Species concentrations in state vector
control_vector_emis	2D emissions in state vector
state_vector_conc_representation	Representation of concentrations in the state vector (all 3D values, column sums, surface values, etc.)
Output configuration	
HistorySpeciesConcToSave	Species concentrations to save to history files
HemcoDiagsToProcess	Which HEMCO diagnostics to include in post-processing, such as total anthropogenic emissions of a given species.
Observation configuration	
observed_species	Species observed
OBSERVER_dirs	Directory storing observation files
LETKF settings	
regularizing_factor_gamma	Parameter γ, adjusts weight assigned to observations
inflation_factor	Ensemble inflation factor Δ
LOCALIZATION_RADIUS_km	Localization radius, in kilometers (km)

ization_factor_gamma), and the ensemble inflation factor Δ (inflation_factor), all of which modulate the weight given to observations in the assimilation calculation. The assimilation window length (assim_time) governs the timescale of the response of the LETKF system to changes in observations and also strongly influences results. Detailed instructions on all settings are in the online documentation (cheereio.readthedocs.io). CHEEREIO then creates a template GEOS-Chem run directory reflecting user settings, which will eventually be copied into an ensemble of m run directories, one per ensemble member; if users modify the template before the ensemble is created, such as to customize emissions inventories, their adjustments will be reflected in each ensemble member. Hence the template run directory allows users to customize their simulations beyond the parameters available in the ens_config.json configuration file. Like any atmospheric model, CHEEREIO must be spun up before any run begins so that it reflects realistic atmospheric conditions; spinning up the template run directory allows the user to run one universal spin-up simulation for all m ensemble members.

With the template initialized, compiled, and spun up, CHEEREIO copies the template into an ensemble of m run directories. Each ensemble member is differentiated by a unique initial perturbation to user-specified emissions, reflecting prior uncertainty in emissions. For example, users interested in assimilating NO_2 observations might specify that they have some prior uncertainty in NO_x emissions; CHEEREIO will then initialize unique grids of NO_x emissions in each ensemble member run directory by drawing samples from a user-specified PDF that perturb existing emissions inventories. These prior errors can be sampled from a normal distribution and can include spatial correlations specified by a correlation distance.

Users can choose to use emissions sampled from either a normal or lognormal spread. If users opt for lognormal emissions, then CHEEREIO samples multiplicative perturbations from a normal distribution centered on zero and then exponentiates to obtain a lognormally distributed sample with a mode of one (i.e., the prior). To meet the LETKF algorithm assumptions, in the lognormal case emissions are transformed back into a normal distribution during the LETKF calculation, before being exponentiated back to a lognormal for use in GEOS-Chem. Benefits of using a lognormal spread include natural protection against negative scaling factors (the lognormal distribution is positive) and a more realistic representation of high-tail uncertainties in emissions inventories.

CHEEREIO grants wide flexibility to users in how emissions perturbations are defined across the ensemble. Users can group emissions of multiple species together into one consolidated entry in the state vector (e.g., NO_x), updated at once at assimilation time. Users can also differentiate emissions by source by separately perturbing subsets of emissions, such as methane from oil and gas and methane from agriculture. The resulting assimilation will provide the user with separate emissions updates for each source, allowing users to easily run source attribution studies. Sectoral separation of emissions is implemented naturally in the LETKF formulation by defining the state vector so that separate source sectors have separate 2D representations; if sources overlap in space, the assimilation update will increment both according to the correlation strength in the prior error covariance matrix, which the user must keep in mind while interpreting source separation results.

Before the assimilation cycle begins, users must run a CHEEREIO-specific spin-up process to create a spread in simulated atmospheric conditions across ensemble members, reflecting the initial perturbations in emissions. Because the LETKF algorithm uses spreads in simulated concentrations across the ensemble to approximate the prior error covariance matrix $\mathbf{P}^{b}$, the model must be run for some period before assimilation begins in order to ensure that variations in concentrations across ensemble members reflect variations in emissions. If this ensemble-wide spin-up is neglected or run for too short a period, $\mathbf{P}^{b}$ will be too small and observations will be neglected (because they will be weighted negligibly in the cost function).

3.3 Runtime

Figure 2 shows a schematic of the CHEEREIO runtime processes. From a computational cluster perspective, CHEEREIO is an array of m jobs, where m is the number of ensemble members specified by the user; each job is allocated p cores as specified by the user. Each job alternates between running GEOS-Chem for an ensemble member and running assimilation scripts for a subset of grid cells – each parallelized separately across the p cores allocated to each job. In this section, we discuss the implementation and control of this complex of processes.

3.3.1 Job control

As shown in Fig. 2, CHEEREIO begins when the user submits a job array initializing m jobs, one for each ensemble member, each consisting of $p$cores within a single node. Within each of the m jobs, the CHEEREIO runtime process is implemented as a shell loop that repeats until the user-specified period of interest is processed, switching smoothly between running GEOS-Chem and running LETKF assimilation calculations. Because the LETKF algorithm is an embarrassingly parallel algorithm, there is no need for complex cross-node parallelization schemes powered by the Message Passing Interface (MPI). Instead, each of the m jobs (parallelized independently across p cores) is coordinated by a job controller, which executes the processes shown in light red in Fig. 2. The job controller synchronizes GEOS-Chem runs and LETKF assimilation routines across the ensemble, ensuring that all jobs remain connected to one another.

At the start of a given assimilation window, each of the m jobs calls GEOS-Chem for the current assimilation window. GEOS-Chem is parallelized within each job across p cores with OpenMP. After completing GEOS-Chem for the assimilation window, each individual job hangs until the job controller indicates that assimilation can begin. Once all GEOS-Chem runs are complete, the job controller initializes the LETKF routine. Each computational core within each job (a total of mp cores) is pre-assigned a set of grid cells to assimilate, as the LETKF algorithm is embarrassingly parallel by grid cell. As a result, the LETKF can make use of multi-node parallelization without MPI; assimilated grid cells are written to a temporary directory, which will be used to update the entire ensemble once all mp cores finish the LETKF calculation. Internal parallelization of p cores within each of the m jobs is handled by GNU Parallel (Tange, 2018). Once all expected grid cells are present, the job controller gathers assimilated grid cell files, which represent assimilated concentrations and emissions, and overwrites GEOS-Chem restart files (representing initial concentrations) and emissions for each ensemble member. The job controller then cleans up temporary files, advances the time period of interest to the next assimilation window, and signals the job array to begin another GEOS-Chem run. If the entire period of interest is complete, then the job controller ends the job array. Different LETKF options, activated from the configuration file, change the behavior of the job control scripts; for example, with the run-in-place functionality activated (Sect. 2.2), CHEEREIO computes the LETKF assimilation update using a long period of observations (e.g., 1 week) but advances the assimilation forward for a smaller amount of time (e.g., 1 d).

CHEEREIO can easily handle emissions updates without GEOS-Chem source code modification because of the HEMCO input module (Keller et al., 2014; Lin et al., 2021). Emission updates are represented by a gridded set of scaling factors, initially randomized for each ensemble member in the initialization process, which are present in each ensemble member run directory in gridded COARDS-compliant NetCDF format. After each assimilation calculation, the file is updated by CHEEREIO to include the latest scaling factors and corresponding time stamp. HEMCO can read and regrid these latest scaling factors on the fly, apply them to the emissions fields, and feed the scaled emissions directly

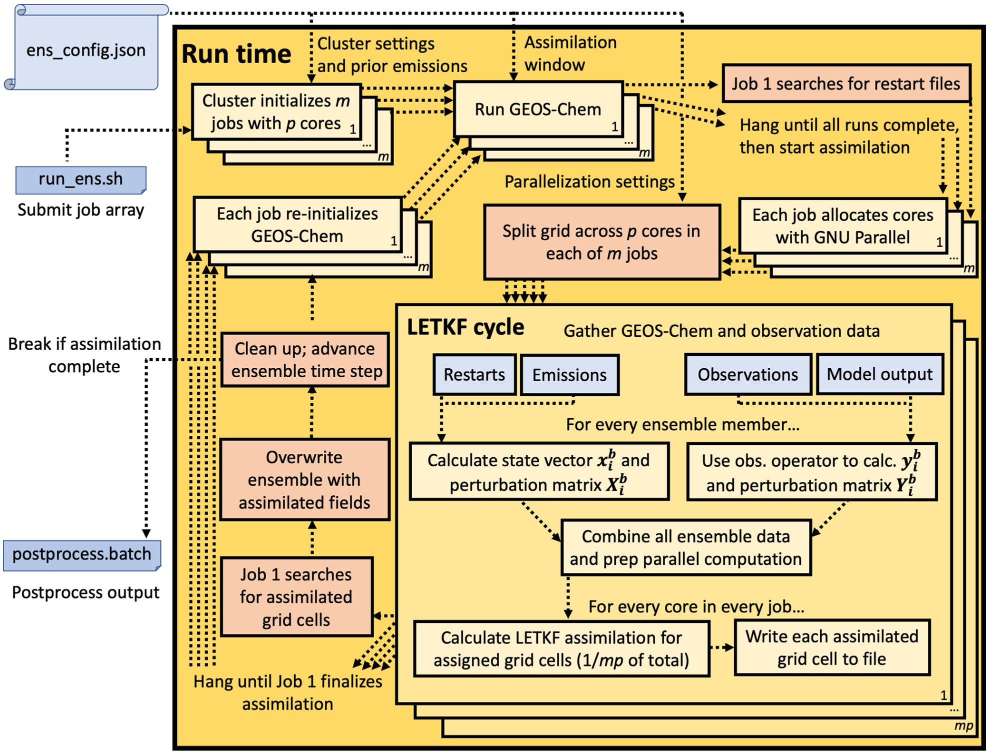
ens_config.json
Run time
Cluster settings and prior emissions
Assimilation window
Job 1 searches for restart files
Cluster initializes m jobs with p cores
1
...
m
Run GEOS-Chem
Hang until all runs complete, then start assimilation
run_ens.sh
Submit job array
Each job re-initializes GEOS-Chem
Parallelization settings
Split grid across p cores in each of m jobs
Each job allocates cores with GNU Parallel
LETKF cycle
Gather GEOS-Chem and observation data
Break if assimilation complete
Clean up; advance ensemble time step
Restarts
Emissions
Observations
Model output
For every ensemble member...
Overwrite ensemble with assimilated fields
Calculate state vector $\boldsymbol{x}_i^b$ and perturbation matrix $\boldsymbol{X}_i^b$
Use obs. operator to calc. $\boldsymbol{y}_i^b$ and perturbation matrix $\boldsymbol{Y}_i^b$
Combine all ensemble data and prep parallel computation
postprocess.batch
Postprocess output
Job 1 searches for assimilated grid cells
For every core in every job...
Calculate LETKF assimilation for assigned grid cells (1/mp of total)
Write each assimilated grid cell to file
Hang until Job 1 finalizes assimilation
mp

Figure 2. Schematic of CHEEREIO runtime routines and job control procedures. CHEEREIO is run as an array of m separate jobs on a computational cluster, one for each ensemble member. These m jobs, operating in parallel, alternate between running GEOS-Chem and running the LETKF algorithm for a subset of grid cells, as shown by the light yellow boxes; the m jobs are coordinated by a single job controller shared by the entire ensemble (shown in light red), ensuring that the ensemble remains synchronized. Boxes in blue show data input into CHEEREIO processes.

into GEOS-Chem, enabling seamless interoperability across CHEEREIO runtime processes.

3.3.2 Assimilation computation

LETKF assimilation is implemented in CHEEREIO using a structure of nested Python objects, designed primarily to ensure that new observation operators can immediately plug into CHEEREIO and work automatically, without requiring users to have deep knowledge of the CHEEREIO code structure. We use Python because of its familiarity to a broad user base, because of its ease of use, and because the object-oriented structure of the language makes it well-suited to the modular design of CHEEREIO.

CHEEREIO works by creating a suite of objects called translators, which load data from gridded NetCDF files used by GEOS-Chem runs, form one-dimensional ensemble state vectors $\boldsymbol{x}_i^{\mathrm{b}}$ and prior vectors of simulated observations $\boldsymbol{y}_i^{\mathrm{b}}$ that are acceptable to the CHEEREIO LETKF routine, and convert assimilated state vectors back into a format acceptable to HEMCO for input into GEOS-Chem. Translator objects are assembled in a nested structure, with low-level translators performing I/O operations and basic calculations to form vectors like $\boldsymbol{x}_i^{\mathrm{b}}$ and $\boldsymbol{y}_i^{\mathrm{b}}$, which are then passed to objects that operate at a higher level of abstraction. Abstract objects do the actual LETKF calculations without any knowledge of the GEOS-Chem simulation or even the user-defined rules on how to construct the state vector, enabled by the fully general nature of the LETKF. Because all the details of a specific simulation are handled by low-level translators, which are designed to easily expand to include new capabilities added by the community, users are able to modify only one small part of CHEEREIO without compromising the overall workflow.

For example, CHEEREIO handles observations by using objects inheriting from the Observation_Translator class, a low-level translator which loads observations from file and compares them to GEOS-Chem output. In object-oriented programming, inheritance can be thought of as a sophisticated form of templating. Indeed, the Observation_Translator class itself is mostly empty and contains instructions to the user on how to write two standardized methods to (1) read observations from file and process them into a Python dictionary formatted for CHEEREIO and (2) generate simulated observations $\boldsymbol{y}_i^{\mathrm{b}}$ from GEOS-Chem output. Users can easily write their own class inheriting from Observation_Translator for a specific use case (like a particular surface or satellite instrument) by implementing these two methods, optionally employing a provided observation toolkit. Any class written with this strict template will then plug in automatically to the rest of the CHEEREIO workflow and can be activated from the main configuration file. CHEEREIO also comes with some pre-written observation operators (such as for the TROPOspheric Monitoring Instrument – TROPOMI – and the Ozone Monitoring Instrument – OMI). Many different observation operators can be used simultaneously, making it straightforward to perform multi-species data assimilation or assimilation using both surface and satellite data within the CHEEREIO framework. Again, because Observation_Translators handle the details of interpreting a specific observation type, the rest of CHEEREIO can remain ignorant of specifics and operate in a fully abstract environment that can be reused for all simulations.

The Observation_Translator template includes tools that support aggregating observations into "super-observations" (Eskes et al., 2003; Miyazaki et al., 2012a). If super-observations are enabled, CHEEREIO will average observations onto the GEOS-Chem spatiotemporal grid. Users can opt to supply a relative or absolute error for observations and opt to either (1) apply these values consistently regardless of whether observations are aggregated or (2) reduce errors as observations are aggregated following a square root law or another functional form supplied by the user (such as an empirical curve) to account for correlations and model transport error. Users can also use error statistics supplied with the observations (such as retrieval errors), with the super-observation error standard deviation calculated according to a function they specify. The default super-observation error standard deviation σ_{super} is calculated as follows.

$$\sigma_{\mathrm{super}} = \sqrt{\left[\left(\frac{1}{n}\sum_{i=1}^{n}\sigma_i\right)\cdot\left(\frac{1-c}{n}+c\right)\right]^2+\sigma_{\mathrm{transport}}^2} \qquad (8)$$

Here σ_i is an individual observation error standard deviation (in the same units as the observation), n is the number of observations aggregated into a super-observation, c is the error correlation between the individual observations averaged into the super-observation, and $\sigma_{\mathrm{transport}}$ represents model transport errors that can be supplied by the user. Model transport error is included as a separate term because transport errors are perfectly correlated for a given model grid cell and therefore irreducible by averaging. Model transport errors in GEOS-Chem can be estimated by the residual error method (Heald et al., 2004; Lu et al., 2021; Chen et al., 2023), though this does not account for any systematic biases from GEOS meteorology or the chemical mechanism. Biases in meteorology could be addressed in the future by using the online version of GEOS-Chem coupled to the GEOS model (Long et al., 2015; Hu et al., 2018; Keller et al., 2021).

3.4 Post-processing

When the CHEEREIO runtime is process complete, users can execute the post-process batch script to automatically consolidate GEOS-Chem diagnostic and emissions output into NetCDF files, along with a file pairing actual observations with simulated observations from the ensemble. Users can also run a control ("prior") simulation with no assimilation within the CHEEREIO environment; output from this run is automatically handled by the post-processing utility to produce plots and data that compare assimilated output to control output. CHEEREIO also produces a suite of graphs and animations depicting a variety of output including scaling factors, concentrations, emissions, and observation information. All plots of results in Sect. 4 of this paper are generated by the CHEEREIO post-processing utility with no additional code. To facilitate additional analysis, a post-processing toolkit is provided for user processing of both consolidated output files and raw ensemble output.

4 Example application: global optimization of methane emissions

Here we demonstrate an end-to-end example application of CHEEREIO to the problem of optimizing global emissions of methane with high temporal (weekly) resolution by assimilation of TROPOMI satellite observations for the full year of 2019. The application uses the standard CHEEREIO configuration files, and all figures and statistics are automatically produced by CHEEREIO with no additional programming. There are weaknesses in the inversion parameters that we identify but do not try to resolve as the application is for demonstration purposes only.

4.1 Demonstration simulation setup

For our demonstration, we use the methane simulation from GEOS-Chem version 14.0.2 (https://doi.org/10.5281/zenodo.7383492, The International GEOS-Chem User Community, 2022) at 2.0° × 2.5° spatial resolution and following the setup described by Qu et al. (2021). Methane emissions come from anthropogenic sources including livestock, oil and gas, coal mining, landfills, wastewater, and rice cultivation, as well as from natural sources including wetlands and termites. Loss is primarily due to oxidation by OH, with additional minor terms from oxidation by Cl atoms, stratospheric oxidation, and soil uptake.

Methane observations used for data assimilation are from the TROPOspheric Monitoring Instrument (TROPOMI) scientific product 2.2.0, shown in Fig. 3 (Lorente et al., 2021a). TROPOMI retrieves daily global dry methane column mixing ratios (XCH_4) at 5.5 km × 7 km nadir pixel resolution at $\sim$13:30 local solar time. In our demonstration, we filter TROPOMI observations to include only those over land below 60° latitude with a quality assurance value > 0.5, shortwave infrared (SWIR) albedo between 0.05 and 0.4 to avoid biases over dark scenes or highly reflective (often

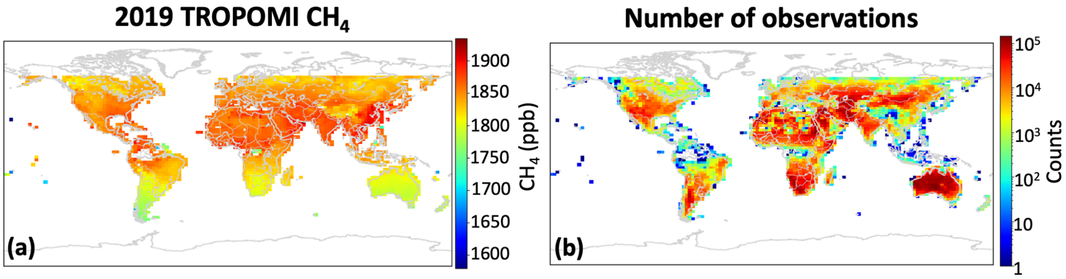

Figure 3. TROPOMI observations used in the CHEEREIO demo for weekly inversion of methane emissions. **(a)** Average TROPOMI XCH_4 for 2019, after filtering as described in the text. **(b)** Number of TROPOMI observations used. Values are plotted on the GEOS-Chem 2° × 2.5° grid.

Table 2. Selected parameters from the CHEEREIO configuration file for methane demonstration.

Parameter[a]	Value
General parameters	
sim_name	CH_4
res	2.0° × 2.5°
start_date, end_date	20181101, 20200101
burn_in_end	20181225
assim_time	168[b]
State vector settings	
state_vector_conc	CH_4
control_vector_emis	CH_4
state_vector_conc_representation	3D
Output configuration	
HemcoDiagsToProcess	EmisCH$_4$_Total
Observation configuration	
observed_species[c]	CH$_4$_TROPOMI : CH_4
TROPOMI_dirs[c]	CH_4 : /path/to/tropomi/netcdf/files
LETKF settings	
regularizing_factor_gamma	1
inflation_factor	0.03
LOCALIZATION_RADIUS_km	500

[a] Parameter descriptions are in Table 1. [b] Hours, equal to 1 week. [c] Many parameters are supplied in key : value form; details in the online documentation.

desert) scenes, a low blended albedo (< 0.75) to avoid snow-covered scenes (Wunch et al., 2011; Lorente et al., 2021a), and SWIR aerosol optical thickness less than 0.1. As shown in Fig. 3, retrieval count after filtering varies strongly by location. CHEEREIO regrids the native XCH_4 observations on the fly to the GEOS-Chem grid resolution, as discussed later in this section.

A subset of the assimilation settings for this demonstration, passed to CHEEREIO through the configuration file, is listed in Table 2. The state vector includes weekly time-dependent global 3D concentrations as well as emissions on the 2° × 2.5° grid over land excluding poleward of 60°. We use a 24-member ensemble, consistent with the LETKF ensemble size used for carbon fluxes in Liu et al. (2019). Each ensemble member is initialized with randomized methane emissions on the 2° × 2.5° grid that range from approximately 50 % to 150 % of prior values, based on a user-specified prior error parameter. Initial emissions for individual members are sampled from a normal distribution with spatial correlation and are normalized so that the initial ensemble mean emissions equal the prior emissions on the 2.0° × 2.5° grid. We spin up the model for each ensemble member for 4 months with these initial emissions and then further multiplicatively increase the ensemble standard deviation of methane concentrations by a factor of 5; the goal of this scaling is to emulate a much longer spin-up run of GEOS-Chem. We then adjust each ensemble member by the same global multiplicative factor so that the ensemble mean methane concentrations are equal to TROPOMI observations at the start of the assimilation period. Furthermore, we discard the first 2 months of assimilated output; we find that the LETKF system has a lag time between when assimilation begins (November 2018) and when emissions updates begin to stabilize, which we call the burn-in period. To reduce the time required for burn-in, for November and December 2018 we use a high regularization constant of $\gamma = 5$ to artificially increase the weight of observations during the burn-in period.

The user does not specify how assimilation increments are split between emissions and concentrations: the LETKF formalism simultaneously updates different aspects of the state vector, emissions and concentrations included, solely according to the correlations between state vector elements represented in the prior error covariance matrix $\mathbf{P}^b$, which is determined by the spread of the CTM ensemble. The prior error in concentrations is determined by the spread in concentrations resulting from the perturbed emissions in each ensemble member. Nevertheless, performance can be enhanced by establishing different parameters for different components of the state vector, such as using different localization radii or inflation schemes (Miyazaki et al., 2012b; Bisht et al., 2023).

Following ensemble spin-up and the burn-in months, we run the model with assimilation for 1 year (2019). We simultaneously assimilate 3D concentrations of methane as well as emissions; the LETKF algorithm natively computes prior

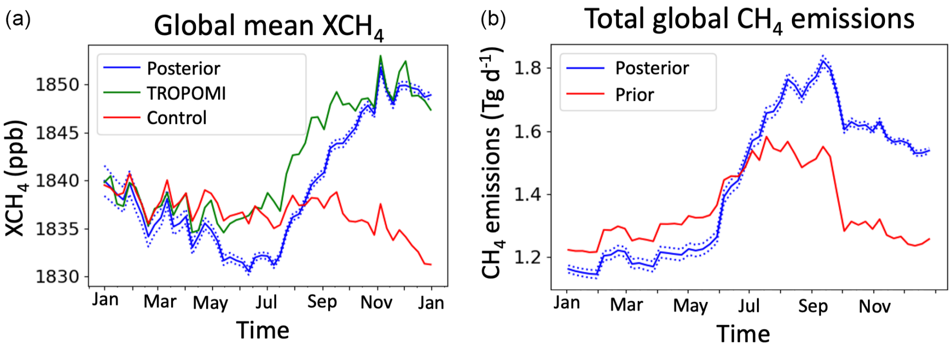

Figure 4. Time-dependent corrections to global methane concentrations and emissions from the weekly LETKF assimilation of TROPOMI observations as demonstrated by CHEEREIO. Panel **(a)** shows the global mean methane dry column mixing ratios (XCH_4) in the TROPOMI observations, the control simulation using prior emissions, and the simulation using posterior emissions. Panel **(b)** shows the global prior and posterior emissions. Posterior values are ensemble means from the assimilation (standard deviations from ensemble members as dotted lines). The assimilation was conducted for 1 year from January through December 2019.

error variance from the ensemble spread (for example, accounting for strong error correlation in the vertical). We use an assimilation period of 1 week and optimize grid cells following a horizontal localization radius of 500 km. We use an inflation factor $\Delta = 0.03$ and a regularization constant $\gamma = 1$. Moreover, we impose a zero floor on emissions; a lognormal emissions spread (Sect. 3.2) would be a better way to prevent negative emissions (Maasakkers et al., 2019). We aggregate TROPOMI methane observations into super-observations on the GEOS-Chem 2.0° × 2.5° grid, reducing errors following Eq. (8), with individual observation error $\sigma_i = 17$ ppb, transport error $\sigma_{transport} = 6.1$ ppb, and error correlation $c = 0.28$, which are values empirically determined for TROPOMI methane by Chen et. al. (2023).

4.2 Posterior solution and evaluation

Figure 4 shows the adjustment of global methane concentrations (Fig. 4a) and emissions (Fig. 4b) from the assimilation calculation relative to the prior emission inventory used in the control simulation. The control simulation produces methane concentrations slightly higher than observed by TROPOMI in January–June, leading to a downward correction of emissions. In July the situation reverses sharply as the control simulation falls well below TROPOMI observations, likely because of a seasonal underestimate of prior emissions from boreal wetlands and rice cultivation (Maasakkers et al., 2019). The assimilation responds with increased emissions but with a 1-month time lag reflecting the need to accumulate sufficient observations to inform the state vector. CHEEREIO's run-in-place capability (Sect. 3.3.1) would allow the LETKF algorithm to mitigate this lag, as would a sliding-window approach such as that used by the Carbon Tracker CH_4 system (Bruhwiler et al., 2014; Liu et al., 2019).

Figure 5 shows the prior and posterior emissions for December 2019, along with the posterior error standard deviation. Figure 6 evaluates the ability of the posterior simulation to better fit the TROPOMI observations in that same month. Model bias is reduced by the LETKF assimilation procedure, with a mean bias of 1.2 ± 10.6 ppb in the ensemble (assimilated) mean compared to −16.1 ± 12.3 ppb in the prior (no assimilation) simulation.

The use of CHEEREIO to optimize methane emissions represents a substantial improvement in computational performance relative to an analytical inversion approach. Qu et al. (2021) previously applied the analytical approach with GEOS-Chem to optimize global methane emissions at 2° × 2.5° resolution for 2019. Their formulation had 4190 state vector elements, which required a total of 4190 perturbed GEOS-Chem simulations. By contrast, our approach only required 24 GEOS-Chem simulations to form the ensemble. Within CHEEREIO, the model spent an average of 29.8 % of wall time running GEOS-Chem and 70.2 % running LETKF routines. The relatively high overhead of LETKF routines occurs in part because the methane GEOS-Chem simulations are relatively fast; full oxidant–aerosol chemistry simulations are considerably more expensive but will have a similar LETKF overhead cost. Accounting for the relatively high LETKF overhead of CHEEREIO, we achieve a factor of 52× reduction in computational costs relative to an equivalent analytical inversion. On an annual basis, our results suggest a 6.5 % increase in global emissions relative to the prior, while Qu et al. (2021) suggest an increase of 2.6 % in global emissions for 2019, but they also decreased global tropospheric OH by 5.7 % (86 % of the total methane sink). Thus, our results are globally consistent. Some spatial patterns are consistent between the two inversions (increases in South and East Asia), while others are not (North America and Europe). Qu et al. (2021) used an older version of TROPOMI data more subject to retrieval artifacts, as documented by Lorente et al. (2021a). Further comparison of LETKF and analytical inversions using the same observations would be of interest.

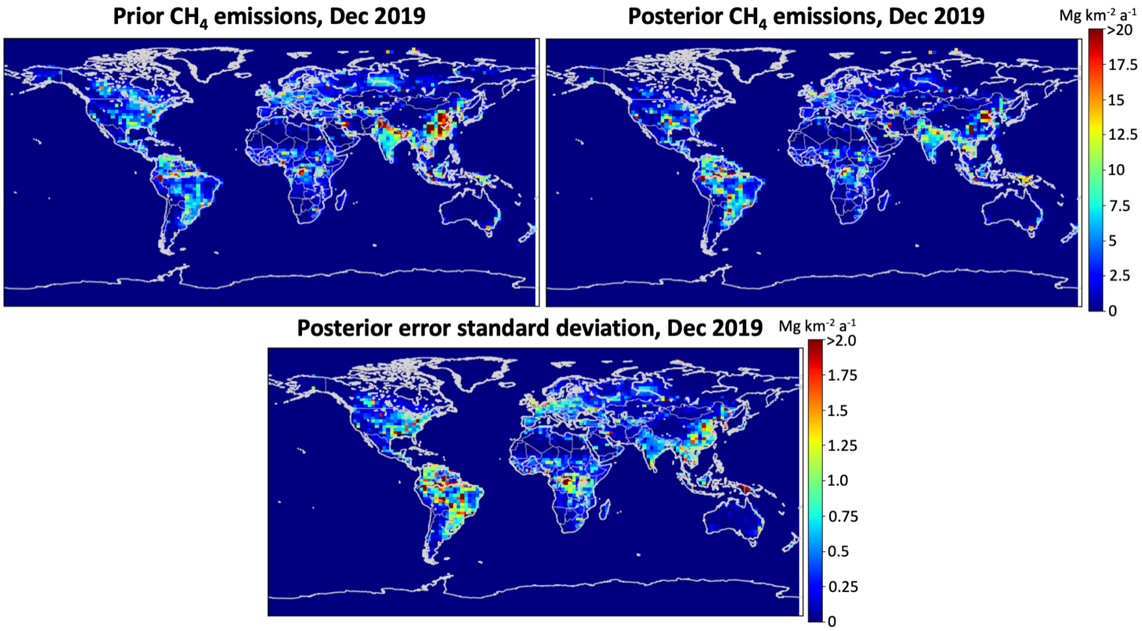
Prior CH_4 emissions, Dec 2019
Posterior CH_4 emissions, Dec 2019
Mg km^{-2} a^{-1}
>20
17.5
15
12.5
10
7.5
5
2.5
0
Posterior error standard deviation, Dec 2019
Mg km^{-2} a^{-1}
>2.0
1.75
1.5
1.25
1.0
0.75
0.5
0.25
0

Figure 5. Prior and posterior estimates of methane emissions in December 2019, as well as error standard deviations for the posterior estimates. The posterior estimates are the means of the 24-member ensemble, and the posterior error standard deviations are defined by the spread in the ensemble.

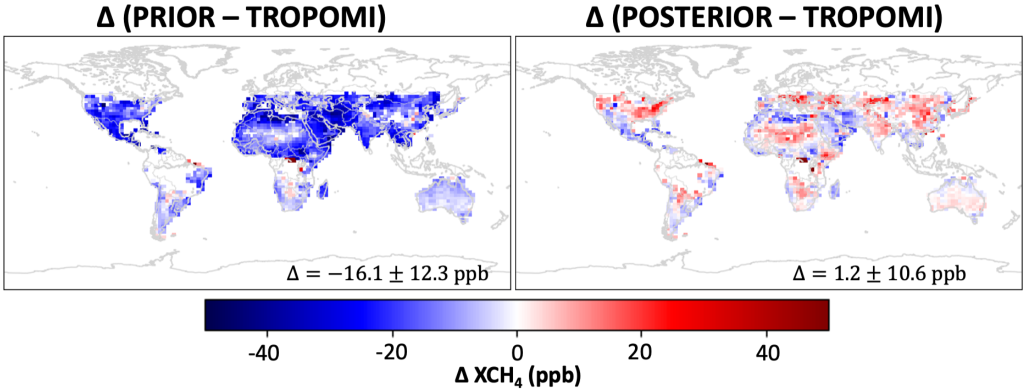

Figure 6. Comparison of simulated dry column mixing ratios (XCH_4) with prior or posterior emissions to TROPOMI observations for December 2019. Values are monthly mean differences between the simulated XCH_4 (with TROPOMI observation operator applied) and TROPOMI observations. Mean bias and standard deviation are given in the inset.

5 Conclusions and future development

We presented the CHemistry and Emissions REanalysis Interface with Observations (CHEEREIO), a user-friendly Python-based tool that supports localized ensemble transform Kalman filter (LETKF) chemical data assimilation (including emissions inversion) powered by the GEOS-Chem chemical transport model (CTM). CHEEREIO provides application-ready and versatile software for users to exploit observations of atmospheric composition from satellites and other platforms to infer emissions and optimize 3D concentration fields, including error characterization. The CHEEREIO source code is available for download at https://github.com/drewpendergrass/CHEEREIO (last access: 22 August 2023) and is documented at https://cheereio.readthedocs.io (last access: 22 August 2023).

We choose the LETKF algorithm because of its general applicability for linear and nonlinear problems, multiple observational data streams, flexible state vector definition, and error characterization of the solution. Its ensemble-based structure is well-suited to developing a simple but powerful tool that requires neither the forward model adjoint nor modifications to model source code and that can be run on supercomputing clusters as an embarrassingly parallel task. Use of GEOS-Chem as a forward model allows a wide range of applications to tropospheric and stratospheric chemistry, as well as simpler linear problems (such as CO_2 or methane inversions), on regional scales with spatial resolution down to 25 km (native resolution of GEOS-Chem) as well as global scales. A critical component of GEOS-Chem is its data input module HEMCO, which allows emissions updates from the assimilation steps to pass seamlessly to GEOS-Chem without code modification.

We designed CHEEREIO so that users can specify their data assimilation problem through a basic configuration file expressing the state vector to be optimized, the prior information, the GEOS-Chem specifications (type of simulation, resolution, assimilation period), and the LETKF parameter information. LETKF implementation is handled under the hood by a suite of CHEEREIO scripts that do not require user familiarity. Users can readily add new observation operators as needed without modifying the rest of the CHEEREIO code base.

We demonstrated CHEEREIO's ability with an example application of assimilating concentrations and emissions of atmospheric methane for 1 full year using observations from TROPOMI. The entire demonstration was run out of the box, with no additional coding beyond the base CHEEREIO code. Output figures and statistics presented here were autogenerated by the CHEEREIO post-processing utility. Accounting for the relatively high overhead of the LETKF computation in the methane case, our approach represents a factor of 52 reduction in computational cost relative to an equivalent analytical inversion. Because of computational cost savings, we envision CHEEREIO's methane data assimilation serving as a global complement to the regional nested-grid simulations offered by the Integrated Methane Inversion (IMI), a similar software platform designed for analytical methane inversions (Varon et al., 2022).

More work can be done to improve CHEEREIO and expand its capability. Although CHEEREIO is designed as a lightweight software wrapper that is accessible to the GEOS-Chem community, future development will incorporate software components from the Joint Effort for Data Assimilation Integration (JEDI). In particular, we plan to support observation operators implemented as part of the JEDI Unified Forward Operator (UFO) initiative, allowing users to leverage the wide library of instruments supported by JEDI without duplicating code themselves. The LETKF algorithm is agnostic to the forward model, making it practical in theory to use any chemical transport model as a forward model for CHEEREIO. In practice, models that use HEMCO for emissions input would be easiest to support. The NASA GEOS and NCAR CESM Earth system models have adopted HEMCO (Lin et al., 2021), and the LETKF approach implemented in CHEEREIO would allow optimization of emissions as part of chemical data assimilation in these models. A benefit of assimilation within coupled meteorology–chemistry models is that transport errors could be explicitly represented.

Because CHEEREIO is designed to take advantage of the embarrassingly parallel LETKF algorithm without using shared memory, it is reasonably straightforward to extend the system to models parallelized with MPI such as the high-performance version of GEOS-Chem (GCHP). Further improvements to the LETKF parallelization routine, in particular methods to share memory resources within Python, can also be applied to reduce I/O overhead, reduce memory use, and improve assimilation wall time. CHEEREIO can be ported on the cloud, taking advantage of GEOS-Chem and satellite data already hosted there (Zhuang et al., 2019, 2020; Varon et al., 2022), thus bringing computing capacity to big data rather than requiring cumbersome data downloads. Cloud implementation would facilitate the development of near-real-time chemical data assimilation products for emissions monitoring and air quality forecasts.

Code availability. The CHEEREIO 1.0 source code is available at https://github.com/drewpendergrass/CHEEREIO (last access: 22 August 2023) and is documented at https://cheereio.readthedocs.io (last access: 22 August 2023). The version of CHEEREIO used in this paper is archived at https://doi.org/10.5281/zenodo.7781437 (Pendergrass, 2023a). GEOS-Chem version 14.0.2 source code is archived at https://doi.org/10.5281/zenodo.7383492 (The International GEOS-Chem User Community, 2022).

Data availability. The CHEEREIO model output from the "Demonstration simulation setup" section of the paper is available at https://doi.org/10.5281/zenodo.7806312 (Pendergrass, 2023b) and contains all necessary data for reproducing Figs. 3–6 including prior methane emissions, posterior methane emissions, and TROPOMI XCH_4 paired with simulated prior and posterior GEOS-Chem XCH_4. The raw TROPOMI science data fed into CHEEREIO are available from SRON (https://ftp.sron.nl/open-access-data-2/TROPOMI/tropomi/ch4/, last access: 6 April 2023, Lorente et al., 2021b) or on request.

Author contributions. DCP and DJJ contributed to the study conceptualization. DCP developed the model code, with contributions from DJV, HN, and MS. KM, KWB, DJJ, and DCP contributed to the method development. DCP performed the data analysis. DCP wrote the original draft, and all authors reviewed and edited the paper.

Competing interests. The contact author has declared that none of the authors has any competing interests.

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Acknowledgements. We thank the two anonymous reviewers for their comments. Part of the research was conducted at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with NASA.

Financial support. This research has been supported by the National Aeronautics and Space Administration (NASA; Carbon Monitoring System; grant no. 80NSSC21K1057). Drew C. Pendergrass was funded by an NSF Graduate Research Fellowship Program (GRFP) grant.

Review statement. This paper was edited by Po-Lun Ma and reviewed by two anonymous referees.

References

Asch, M., Bocquet, M., and Nodet, M.: Data Assimilation, in: Fundamentals of Algorithms, Society for Industrial and Applied Mathematics, https://doi.org/10.1137/1.9781611974546, 2016.

Bisht, J. S. H., Patra, P. K., Takigawa, M., Sekiya, T., Kanaya, Y., Saitoh, N., and Miyazaki, K.: Estimation of CH_4 emission based on an advanced 4D-LETKF assimilation system, Geosci. Model Dev., 16, 1823–1838, https://doi.org/10.5194/gmd-16-1823-2023, 2023.

Bocquet, M., Elbern, H., Eskes, H., Hirtl, M., Žabkar, R., Carmichael, G. R., Flemming, J., Inness, A., Pagowski, M., Pérez Camaño, J. L., Saide, P. E., San Jose, R., Sofiev, M., Vira, J., Baklanov, A., Carnevale, C., Grell, G., and Seigneur, C.: Data assimilation in atmospheric chemistry models: current status and future prospects for coupled chemistry meteorology models, Atmos. Chem. Phys., 15, 5325–5358, https://doi.org/10.5194/acp-15-5325-2015, 2015.

Brasseur, G. and Jacob, D.: Modeling of Atmospheric Chemistry, Cambridge University Press, https://doi.org/10.1017/9781316544754, 2017.

Bruhwiler, L., Dlugokencky, E., Masarie, K., Ishizawa, M., Andrews, A., Miller, J., Sweeney, C., Tans, P., and Worthy, D.: CarbonTracker$-CH_4$: an assimilation system for estimating emissions of atmospheric methane, Atmos. Chem. Phys., 14, 8269–8293, https://doi.org/10.5194/acp-14-8269-2014, 2014.

Chen, Z., Jacob, D. J., Gautam, R., Omara, M., Stavins, R. N., Stowe, R. C., Nesser, H., Sulprizio, M. P., Lorente, A., Varon, D. J., Lu, X., Shen, L., Qu, Z., Pendergrass, D. C., and Hancock, S.: Satellite quantification of methane emissions and oil–gas methane intensities from individual countries in the Middle East and North Africa: implications for climate action, Atmos. Chem. Phys., 23, 5945–5967, https://doi.org/10.5194/acp-23-5945-2023, 2023.

Courtier, P., Thépaut, J.-N., and Hollingsworth, A.: A strategy for operational implementation of 4D-Var, using an incremental approach, Q. J. Roy. Meteor. Soc., 120, 1367–1387, https://doi.org/10.1002/qj.49712051912, 1994.

Dai, T., Cheng, Y., Goto, D., Li, Y., Tang, X., Shi, G., and Nakajima, T.: Revealing the sulfur dioxide emission reductions in China by assimilating surface observations in WRF-Chem, Atmos. Chem. Phys., 21, 4357–4379, https://doi.org/10.5194/acp-21-4357-2021, 2021.

Ding, J., van der A, R. J., Mijling, B., and Levelt, P. F.: Space-based NO_x emission estimates over remote regions improved in DECSO, Atmos. Meas. Tech., 10, 925–938, https://doi.org/10.5194/amt-10-925-2017, 2017.

Eastham, S. D., Long, M. S., Keller, C. A., Lundgren, E., Yantosca, R. M., Zhuang, J., Li, C., Lee, C. J., Yannetti, M., Auer, B. M., Clune, T. L., Kouatchou, J., Putman, W. M., Thompson, M. A., Trayanov, A. L., Molod, A. M., Martin, R. V., and Jacob, D. J.: GEOS-Chem High Performance (GCHP v11-02c): a next-generation implementation of the GEOS-Chem chemical transport model for massively parallel applications, Geosci. Model Dev., 11, 2941–2953, https://doi.org/10.5194/gmd-11-2941-2018, 2018.

Elbern, H. and Schmidt, H.: Ozone episode analysis by four-dimensional variational chemistry data assimilation, J. Geophys. Res.-Atmos., 106, 3569–3590, https://doi.org/10.1029/2000JD900448, 2001.

Emili, E., Barret, B., Massart, S., Le Flochmoen, E., Piacentini, A., El Amraoui, L., Pannekoucke, O., and Cariolle, D.: Combined assimilation of IASI and MLS observations to constrain tropospheric and stratospheric ozone in a global chemical transport model, Atmos. Chem. Phys., 14, 177–198, https://doi.org/10.5194/acp-14-177-2014, 2014.

Eskes, H. J., Velthoven, P. F. J. V., Valks, P. J. M., and Kelder, H. M., Assimilation of GOME total-ozone satellite observations in a three-dimensional tracer-transport model, Q. J. Roy. Meteor. Soc., 129, 1663–1681, https://doi.org/10.1256/qj.02.14, 2003.

Feng, L., Palmer, P. I., Zhu, S., Parker, R. J., and Liu, Y.: Tropical methane emissions explain large fraction of recent changes in global atmospheric methane growth rate, Nat. Commun., 13, 1378, https://doi.org/10.1038/s41467-022-28989-z, 2022.

Feng, L., Palmer, P. I., Parker, R. J., Lunt, M. F., and Bösch, H.: Methane emissions are predominantly responsible for record-breaking atmospheric methane growth rates in 2020 and 2021, Atmos. Chem. Phys., 23, 4863–4880, https://doi.org/10.5194/acp-23-4863-2023, 2023.

Flemming, J., Huijnen, V., Arteta, J., Bechtold, P., Beljaars, A., Blechschmidt, A.-M., Diamantakis, M., Engelen, R. J., Gaudel, A., Inness, A., Jones, L., Josse, B., Katragkou, E., Marecal, V., Peuch, V.-H., Richter, A., Schultz, M. G., Stein, O., and Tsikerdekis, A.: Tropospheric chemistry in the Integrated Forecasting System of ECMWF, Geosci. Model Dev., 8, 975–1003, https://doi.org/10.5194/gmd-8-975-2015, 2015.

Heald, C. L., Jacob, D. J., Jones, D. B. A., Palmer, P. I., Logan, J. A., Streets, D. G., Sachse, G. W., Gille, J. C., Hoffman, R. N., and Nehrkorn, T.: Comparative inverse analysis of satellite (MOPITT) and aircraft (TRACE-P) observations to estimate Asian sources of carbon monoxide, J. Geophys. Res.-Atmos., 109, D23306, https://doi.org/10.1029/2004JD005185, 2004.

Hu, L., Keller, C. A., Long, M. S., Sherwen, T., Auer, B., Da Silva, A., Nielsen, J. E., Pawson, S., Thompson, M. A., Trayanov, A. L., Travis, K. R., Grange, S. K., Evans, M. J., and Jacob, D. J.: Global simulation of tropospheric chemistry at 12.5 km resolution: performance and evaluation of the GEOS-Chem chemical module (v10-1) within the NASA GEOS Earth system model (GEOS-5 ESM), Geosci. Model Dev., 11, 4603–4620, https://doi.org/10.5194/gmd-11-4603-2018, 2018.

Hunt, B. R., Kostelich, E. J., and Szunyogh, I.: Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter, Physica D, 230, 112–126, https://doi.org/10.1016/j.physd.2006.11.008, 2007.

Inness, A., Blechschmidt, A.-M., Bouarar, I., Chabrillat, S., Crepulja, M., Engelen, R. J., Eskes, H., Flemming, J., Gaudel, A., Hendrick, F., Huijnen, V., Jones, L., Kapsomenakis, J., Katragkou, E., Keppens, A., Langerock, B., de Mazière, M., Melas, D., Parrington, M., Peuch, V. H., Razinger, M., Richter, A., Schultz, M. G., Suttie, M., Thouret, V., Vrekoussis, M., Wagner, A., and Zerefos, C.: Data assimilation of satellite-retrieved ozone, carbon monoxide and nitrogen dioxide with ECMWF's Composition-IFS, Atmos. Chem. Phys., 15, 5275–5303, https://doi.org/10.5194/acp-15-5275-2015, 2015.

Jiang, Z., McDonald, B. C., Worden, H., Worden, J. R., Miyazaki, K., Qu, Z., Henze, D. K., Jones, D. B. A., Arellano, A. F., Fischer, E. V., Zhu, L., and Boersma, K. F.: Unexpected slowdown of US pollutant emission reduction in the past decade, P. Natl. Acad. Sci. USA, 115, 5099–5104, https://doi.org/10.1073/pnas.1801191115, 2018.

Kahnert, M.: Variational data analysis of aerosol species in a regional CTM: Background error covariance constraint and aerosol optical observation operators, Tellus B, 60, 753–770, https://doi.org/10.1111/j.1600-0889.2008.00377.x, 2008.

Kalnay, E.: Atmospheric Modeling, Data Assimilation and Predictability, Cambridge University Press, Cambridge, https://doi.org/10.1017/CBO9780511802270, 2003.

Keller, C. A., Long, M. S., Yantosca, R. M., Da Silva, A. M., Pawson, S., and Jacob, D. J.: HEMCO v1.0: a versatile, ESMF-compliant component for calculating emissions in atmospheric models, Geosci. Model Dev., 7, 1409–1417, https://doi.org/10.5194/gmd-7-1409-2014, 2014.

Keller, C. A., Knowland, K. E., Duncan, B. N., Liu, J., Anderson, D. C., Das, S., Lucchesi, R. A., Lundgren, E. W., Nicely, J. M., Nielsen, E., Ott, L. E., Saunders, E., Strode, S. A., Wales, P. A., Jacob, D. J., and Pawson, S.: Description of the NASA GEOS Composition Forecast Modeling System GEOS-CF v1.0, J. Adv. Model. Earth Sy., 13, e2020MS002413, https://doi.org/10.1029/2020MS002413, 2021.

Kong, Y., Zheng, B., Zhang, Q., and He, K.: Global and regional carbon budget for 2015–2020 inferred from OCO-2 based on an ensemble Kalman filter coupled with GEOS-Chem, Atmos. Chem. Phys., 22, 10769–10788, https://doi.org/10.5194/acp-22-10769-2022, 2022.

Lin, H., Jacob, D. J., Lundgren, E. W., Sulprizio, M. P., Keller, C. A., Fritz, T. M., Eastham, S. D., Emmons, L. K., Campbell, P. C., Baker, B., Saylor, R. D., and Montuoro, R.: Harmonized Emissions Component (HEMCO) 3.0 as a versatile emissions component for atmospheric models: application in the GEOS-Chem, NASA GEOS, WRF-GC, CESM2, NOAA GEFS-Aerosol, and NOAA UFS models, Geosci. Model Dev., 14, 5487–5506, https://doi.org/10.5194/gmd-14-5487-2021, 2021.

Liu, J., Bowman, K. W., and Lee, M.: Comparison between the Local Ensemble Transform Kalman Filter (LETKF) and 4D-Var in atmospheric CO_2 flux inversion with the Goddard Earth Observing System-Chem model and the observation impact diagnostics from the LETKF, J. Geophys. Res.-Atmos., 121, 13066–13-087, https://doi.org/10.1002/2016JD025100, 2016.

Liu, Y., Kalnay, E., Zeng, N., Asrar, G., Chen, Z., and Jia, B.: Estimating surface carbon fluxes based on a local ensemble transform Kalman filter with a short assimilation window and a long observation window: an observing system simulation experiment test in GEOS-Chem 10.1, Geosci. Model Dev., 12, 2899–2914, https://doi.org/10.5194/gmd-12-2899-2019, 2019.

Long, M. S., Yantosca, R., Nielsen, J. E., Keller, C. A., da Silva, A., Sulprizio, M. P., Pawson, S., and Jacob, D. J.: Development of a grid-independent GEOS-Chem chemical transport model (v9-02) as an atmospheric chemistry module for Earth system models, Geosci. Model Dev., 8, 595–602, https://doi.org/10.5194/gmd-8-595-2015, 2015.

Lorente, A., Borsdorff, T., Butz, A., Hasekamp, O., aan de Brugh, J., Schneider, A., Wu, L., Hase, F., Kivi, R., Wunch, D., Pollard, D. F., Shiomi, K., Deutscher, N. M., Velazco, V. A., Roehl, C. M., Wennberg, P. O., Warneke, T., and Landgraf, J.: Methane retrieved from TROPOMI: improvement of the data product and validation of the first 2 years of measurements, Atmos. Meas. Tech., 14, 665–684, https://doi.org/10.5194/amt-14-665-2021, 2021a.

Lorente, A., Borsdorff, T., aan de Brugh, J., Landgraf, J., and Hasekamp, O.: SRON S5P – RemoTeC scientific TROPOMI XCH4 dataset, Zenodo [data set], https://doi.org/10.5281/zenodo.4447228, 2021b.

Lu, X., Jacob, D. J., Zhang, Y., Maasakkers, J. D., Sulprizio, M. P., Shen, L., Qu, Z., Scarpelli, T. R., Nesser, H., Yantosca, R. M., Sheng, J., Andrews, A., Parker, R. J., Boesch, H., Bloom, A. A., and Ma, S.: Global methane budget and trend, 2010–2017: complementarity of inverse analyses using in situ (GLOBALVIEWplus CH_4 ObsPack) and satellite (GOSAT) observations, Atmos. Chem. Phys., 21, 4637–4657, https://doi.org/10.5194/acp-21-4637-2021, 2021.

Ma, C., Wang, T., Mizzi, A. P., Anderson, J. L., Zhuang, B., Xie, M., and Wu, R.: Multiconstituent Data Assimilation With WRF-Chem/DART: Potential for Adjusting Anthropogenic Emissions and Improving Air Quality Forecasts Over Eastern China, J. Geophys. Res.-Atmos., 124, 7393–7412, https://doi.org/10.1029/2019JD030421, 2019.

Maasakkers, J. D., Jacob, D. J., Sulprizio, M. P., Scarpelli, T. R., Nesser, H., Sheng, J.-X., Zhang, Y., Hersher, M., Bloom, A. A., Bowman, K. W., Worden, J. R., Janssens-Maenhout, G., and Parker, R. J.: Global distribution of methane emissions, emission trends, and OH concentrations and trends inferred from an inversion of GOSAT satellite data for 2010–2015, Atmos. Chem. Phys., 19, 7859–7881, https://doi.org/10.5194/acp-19-7859-2019, 2019.

Martin, R. V., Eastham, S. D., Bindle, L., Lundgren, E. W., Clune, T. L., Keller, C. A., Downs, W., Zhang, D., Lucchesi, R. A., Sulprizio, M. P., Yantosca, R. M., Li, Y., Estrada, L., Putman, W. M., Auer, B. M., Trayanov, A. L., Pawson, S., and Jacob, D. J.: Improved advection, resolution, performance, and community access in the new generation (version 13) of the high-performance GEOS-Chem global atmospheric chemistry model (GCHP), Geosci. Model Dev., 15, 8731–8748, https://doi.org/10.5194/gmd-15-8731-2022, 2022.

Mijling, B. and van der A, R. J.: Using daily satellite observations to estimate emissions of short-lived air pollutants on a mesoscopic scale, J. Geophys. Res.-Atmos., 117, D17302, https://doi.org/10.1029/2012JD017817, 2012.

Miyazaki, K., Eskes, H. J., and Sudo, K.: Global NO_x emission estimates derived from an assimilation of OMI tropospheric NO_2 columns, Atmos. Chem. Phys., 12, 2263–2288, https://doi.org/10.5194/acp-12-2263-2012, 2012a.

Miyazaki, K., Eskes, H. J., Sudo, K., Takigawa, M., van Weele, M., and Boersma, K. F.: Simultaneous assimilation of satellite NO_2, O_3, CO, and HNO_3 data for the analysis of tropospheric chemical composition and emissions, Atmos. Chem. Phys., 12, 9545–9579, https://doi.org/10.5194/acp-12-9545-2012, 2012b.

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, Atmos. Chem. Phys., 15, 8315–8348, https://doi.org/10.5194/acp-15-8315-2015, 2015.

Miyazaki, K., Eskes, H., Sudo, K., Boersma, K. F., Bowman, K., and Kanaya, Y.: Decadal changes in global surface NO_x emissions from multi-constituent satellite data assimilation, Atmos. Chem. Phys., 17, 807–837, https://doi.org/10.5194/acp-17-807-2017, 2017.

Miyazaki, K., Bowman, K., Sekiya, T., Eskes, H., Boersma, F., Worden, H., Livesey, N., Payne, V. H., Sudo, K., Kanaya, Y., Takigawa, M., and Ogochi, K.: Updated tropospheric chemistry reanalysis and emission estimates, TCR-2, for 2005–2018, Earth Syst. Sci. Data, 12, 2223–2259, https://doi.org/10.5194/essd-12-2223-2020, 2020.

Nesser, H., Jacob, D. J., Maasakkers, J. D., Scarpelli, T. R., Sulprizio, M. P., Zhang, Y., and Rycroft, C. H.: Reduced-cost construction of Jacobian matrices for high-resolution inversions of satellite observations of atmospheric composition, Atmos. Meas. Tech., 14, 5521–5534, https://doi.org/10.5194/amt-14-5521-2021, 2021.

Pendergrass, D.: drewpendergrass/CHEEREIO: CHEEREIO v1.0.0 release (v1.0.0), Zenodo [code], https://doi.org/10.5281/zenodo.7781437, 2023a.

Pendergrass, D.: Replication Data for: CHEEREIO 1.0: a versatile and user-friendly ensemble-based chemical data assimilation and emissions inversion platform for the GEOS-Chem chemical transport model, Zenodo [data set], https://doi.org/10.5281/zenodo.7806312, 2023b.

Peters, W., Miller, J. B., Whitaker, J., Denning, A. S., Hirsch, A., Krol, M. C., Zupanski, D., Bruhwiler, L., and Tans, P. P.: An ensemble data assimilation system to estimate CO_2 surface fluxes from atmospheric trace gas observations, J. Geophys. Res.-Atmos., 110, D24304, https://doi.org/10.1029/2005JD006157, 2005.

Qu, Z., Henze, D. K., Li, C., Theys, N., Wang, Y., Wang, J., Wang, W., Han, J., Shim, C., Dickerson, R. R., and Ren, X.: SO_2 Emission Estimates Using OMI SO_2 Retrievals for 2005–2017, J. Geophys. Res.-Atmos., 124, 8336–8359, https://doi.org/10.1029/2019JD030243, 2019.

Qu, Z., Jacob, D. J., Shen, L., Lu, X., Zhang, Y., Scarpelli, T. R., Nesser, H., Sulprizio, M. P., Maasakkers, J. D., Bloom, A. A., Worden, J. R., Parker, R. J., and Delgado, A. L.: Global distribution of methane emissions: a comparative inverse analysis of observations from the TROPOMI and GOSAT satellite instruments, Atmos. Chem. Phys., 21, 14159–14175, https://doi.org/10.5194/acp-21-14159-2021, 2021.

Rodgers, C. D.: Inverse Methods for Atmospheric Sounding: Theory and Practice, Series on Atmospheric, Oceanic and Planetary Physics – Vol. 2, World Scientific Publishing Co. Pte. Ltd., Singapore, https://doi.org/10.1142/3171, 2000.

Schuh, A. E., Jacobson, A. R., Basu, S., Weir, B., Baker, D., Bowman, K., Chevallier, F., Crowell, S., Davis, K. J., Deng, F., Denning, S., Feng, L., Jones, D., Liu, J., and Palmer, P. I.: Quantifying the Impact of Atmospheric Transport Uncertainty on CO_2 Surface Flux Estimates, Global Biogeochem. Cy., 33, 484–500, https://doi.org/10.1029/2018GB006086, 2019.

Schuh, A. E., Byrne, B., Jacobson, A. R., Crowell, S. M. R., Deng, F., Baker, D. F., Johnson, M. S., Philip, S., and Weir, B.: On the role of atmospheric model transport uncertainty in estimating the Chinese land carbon sink, Nature, 603, E13–E14, https://doi.org/10.1038/s41586-021-04258-9, 2022.

Stanevich, I., Jones, D. B. A., Strong, K., Keller, M., Henze, D. K., Parker, R. J., Boesch, H., Wunch, D., Notholt, J., Petri, C., Warneke, T., Sussmann, R., Schneider, M., Hase, F., Kivi, R., Deutscher, N. M., Velazco, V. A., Walker, K. A., and Deng, F.: Characterizing model errors in chemical transport modeling of methane: using GOSAT XCH_4 data with weak-constraint four-dimensional variational data assimilation, Atmos. Chem. Phys., 21, 9545–9572, https://doi.org/10.5194/acp-21-9545-2021, 2021.

Tange, O.: GNU Parallel 2018, Zenodo [code], https://doi.org/10.5281/zenodo.1146014, 2018.

The International GEOS-Chem User Community: geoschem/GC-Classic: GEOS-Chem Classic 14.0.2 (14.0.2), Zenodo [code], https://doi.org/10.5281/zenodo.7383492, 2022.

Trémolet, Y. and Auligné, T.: The Joint Effort for Data Assimilation Integration (JEDI), JCSDA Quarterly Newsletter, 66, 1–5, https://doi.org/10.25923/RB19-0Q26, 2020.

van der Graaf, S., Dammers, E., Segers, A., Kranenburg, R., Schaap, M., Shephard, M. W., and Erisman, J. W.: Data assimilation of CrIS NH_3 satellite observations for improving spatiotemporal NH_3 distributions in LOTOS-EUROS, Atmos. Chem. Phys., 22, 951–972, https://doi.org/10.5194/acp-22-951-2022, 2022.

Varon, D. J., Jacob, D. J., Sulprizio, M., Estrada, L. A., Downs, W. B., Shen, L., Hancock, S. E., Nesser, H., Qu, Z., Penn, E., Chen, Z., Lu, X., Lorente, A., Tewari, A., and Randles, C. A.: Integrated Methane Inversion (IMI 1.0): a user-friendly, cloud-based facility for inferring high-resolution methane emissions from TROPOMI satellite observations, Geosci. Model Dev., 15, 5787–5805, https://doi.org/10.5194/gmd-15-5787-2022, 2022.

Wunch, D., Wennberg, P. O., Toon, G. C., Connor, B. J., Fisher, B., Osterman, G. B., Frankenberg, C., Mandrake, L., O'Dell, C., Ahonen, P., Biraud, S. C., Castano, R., Cressie, N., Crisp, D., Deutscher, N. M., Eldering, A., Fisher, M. L., Griffith, D. W. T., Gunson, M., Heikkinen, P., Keppel-Aleks, G., Kyrö, E., Lindenmaier, R., Macatangay, R., Mendonca, J., Messerschmidt, J., Miller, C. E., Morino, I., Notholt, J., Oyafuso, F. A., Rettinger, M., Robinson, J., Roehl, C. M., Salawitch, R. J., Sherlock, V., Strong, K., Sussmann, R., Tanaka, T., Thompson, D. R., Uchino, O., Warneke, T., and Wofsy, S. C.: A method for evaluating bias in global measurements of CO_2 total columns from space, Atmos. Chem. Phys., 11, 12317–12337, https://doi.org/10.5194/acp-11-12317-2011, 2011.

Zhu, S., Feng, L., Liu, Y., Wang, J., and Yang, D.: Decadal Methane Emission Trend Inferred from Proxy GOSAT XCH_4 Retrievals: Impacts of Transport Model Spatial Resolution, Adv. Atmos.

Sci., 39, 1343–1359, https://doi.org/10.1007/s00376-022-1434-6, 2022.

Zhuang, J., Jacob, D. J., Gaya, J. F., Yantosca, R. M., Lundgren, E. W., Sulprizio, M. P., and Eastham, S. D.: Enabling Immediate Access to Earth Science Models through Cloud Computing: Application to the GEOS-Chem Model, B. Am. Meteorol. Soc., 100, 1943–1960, https://doi.org/10.1175/BAMS-D-18-0243.1, 2019.

Zhuang, J., Jacob, D. J., Lin, H., Lundgren, E. W., Yantosca, R. M., Gaya, J. F., Sulprizio, M. P., and Eastham, S. D.: Enabling High-Performance Cloud Computing for Earth Science Modeling on Over a Thousand Cores: Application to the GEOS-Chem Atmospheric Chemistry Model, J. Adv. Model. Earth Sy., 12, e2020MS002064, https://doi.org/10.1029/2020MS002064, 2020.