https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022

Geoscientific Model Development Discussions

ICLASS 1.0: a variational Inverse modelling framework for the Chemistry Land-surface Atmosphere Soil Slab model: description, validation and application

Peter J.M. Bosman[1] and Maarten C. Krol[1,2]

[1]Meteorology and Air Quality Group, Wageningen University, Wageningen, the Netherlands
[2]Institute for Marine and Atmospheric Research, Utrecht University, Utrecht, The Netherlands

Correspondence: Peter Bosman (peter.bosman.publicaddress@gmail.com)

Abstract. This paper provides a description of ICLASS 1.0: a variational Inverse modelling framework for the Chemistry Land-surface Atmosphere Soil Slab model. This framework can be used to study the atmospheric boundary layer, surface layer or the exchange of gases, moisture, heat and momentum between the land surface and the lower atmosphere. The general aim of the framework is to allow to assimilate various streams of observations (fluxes, mixing ratios at multiple heights, ...) to estimate model parameters, thereby obtaining a physical model that is consistent with a diverse set of observations. The framework allows to retrieve parameters in an objective manner, and enables to estimate information that is difficult to obtain directly by observations, for example free-tropospheric mixing ratios or stomatal conductances. Furthermore it allows to estimate possible biases in observations. Modelling the carbon cycle at ecosystem level is one of the main intended fields of application. The physical model around which the framework is constructed is relatively simple, yet contains the core physics to model the essentials of a well-mixed boundary layer and of land–atmosphere exchange. The model includes an explicit description of the atmospheric surface layer, a region where scalars show relatively large gradients with height. An important challenge is the strong non-linearity of the model, which complicates the estimation of best parameter values. The constructed adjoint of the tangent linear model can be used to mitigate this challenge. The adjoint allows for an analytical gradient of the objective cost function, used for minimisation of this function. An implemented Monte-Carlo way of running ICLASS can further help to handle non-linearity, and provides posterior statistics on the estimated parameters. The paper provides a technical description of the framework, includes a validation of the adjoint code, as well as tests for the full inverse modelling framework and a successful example application for a grassland in the Netherlands.

1 Introduction

Exchange of heat, mass and momentum between the land surface and the atmosphere plays an essential role for weather, climate, air quality and biogeochemical cycles. Under sunny daytime conditions a relatively well-mixed layer usually forms close to the land surface, the well-known atmospheric boundary layer (ABL). This layer is directly impacted by exchange processes with the land surface and is also a layer where humans live in. Modelling the composition and thermodynamic state of the ABL in its full interaction with the land surface is the target of the Chemistry Land-surface Atmosphere Soil

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022


Geoscientific Model Development Discussions

Slab model (CLASS; Vilà-Guerau De Arellano et al., 2015). This and similar models have been applied frequently, e.g. for understanding the daily cycle of evapotranspiration (van Heerwaarden et al., 2010), studying the effects of aerosols on boundary layer dynamics (Barbaro et al., 2014), studying the effects of elevated CO_2 on boundary layer clouds (Vilà-Guerau De Arellano et al., 2012) or for studying the ammonia budget (Schulte et al., 2021). Next to a representation of the ABL, the CLASS model includes a simple representation for the exchange of gases, heat, moisture and momentum between the land surface and the lower atmosphere. The model explicitly accounts for the surface layer, which is a layer within the ABL close to the surface with relatively strong vertical gradients of scalars and momentum (Stull, 1988). Since the ABL model physics are relatively simple and only include the essential boundary layer processes, the model performs best on what might be called "golden days". Those are days in which advection is either absent or uniform in time and space, deep convection is absent, and sufficient incoming shortwave radiation heats the surface allowing for the formation of a prototypical convective boundary layer.

To further our understanding of land–atmosphere exchange, tall tower observational sites have been established (Cabauw, the Netherlands (Bosveld et al., 2020; Vermeulen et al., 2011); Hyytiälä, Finland (Vesala et al., 2005); Harvard Forest, USA (Commane et al., 2015); ...), providing time series of different types of measurements (observation streams). Even so, many studies only use a small fraction of the different streams of observations available for a specific day and location (e.g. Vilà-Guerau De Arellano et al., 2012). A model like CLASS can be used to fit an extensive set of observation streams simultaneously. When model results are consistent with a diverse set of measurements, this gives more confidence that the internal physics are robust and the model has been adequately parameterised to reliably simulate reality. However, an important difficulty in the application of a model like CLASS concerns parameter tuning to obtain a good fit to observations. Some parameters can be obtained directly from observations (e.g. inital mixed layer humidity), but e.g. estimating free-tropospheric lapse rates or stomatal conductances is often more challenging. When many parameters need to be determined, the feasible parameter space becomes vast. If this vast parameter space is not properly explored, the obtained parameters can be subjective and sub-optimal. Next to that, some of the available ecosystem/ABL-level observations may suffer from biases. An example is the closure of the surface energy balance, where the available energy is often larger than the sum of the latent and sensible heat flux (Foken, 2008). This energy balance closure problem is a known issue with eddy-covariance observations (Foken, 2008; Oncley et al., 2007; Renner et al., 2019), and various explanations have been suggested (Foken, 2008).

The above text illustrates the need for an objective optimisation framework, capable of correcting observations for biases. We therefore present here a description of ICLASS, an inverse modelling framework built around the CLASS model, including a bias-correction scheme. This framework can estimate model parameters, by minimising an objective cost function using a variational (Chevallier et al., 2005) framework. ICLASS uses a Bayesian approach, in the sense that it combines information, both from observations and from prior knowledge about the parameters, to come to a solution with a reduced uncertainty in the optimised parameters. A major strength of this framework is that it allows to incorporate several streams of observations, for instance chemical fluxes, mixing ratios and temperatures at multiple heights, and radiosonde observations of the boundary-layer height. By optimizing a number of predefined key parameters of the model, we aim to obtain a diurnal simulation that is consistent with a diverse set of measurements. Additionally, error statistics that are estimated provide information about the constraints the measurements place on the model parameters. Modelling the carbon cycle at ecosystem level is one of the main

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022


Geoscientific Model Development Discussions

intended fields of application. As an example, with some extensions to the framework, ICLASS could be applied to ecosystem observations of the coupled exchange of CO_2 and carbonyl sulfide (a tracer for obtaining stomatal conductance, Whelan et al., 2018).

An important challenge for the optimisation framework is the strong non-linearity of the model. As an example, a stronger evapotranspiration flux leads to an increased specific humidity in the mixed layer, which in turn reduces the evapotranspiration flux again (van Heerwaarden et al., 2009). The non-linearity causes numerically-calculated cost function gradients to deviate from the true analytical gradients, since the cost function can vary erratically with a changing model parameter value. This is hampering proper minimization of the cost function when using numerically calculated gradients. Raoult et al. (2016) used an *adjoint* to optimise parameters of a land surface model. Constructing the adjoint of the tangent linear model is a way to obtain more accurate gradient calculations, as the adjoint provides a locally exact analytical gradient of the cost function at the locations where the function is differentiable. Margulis and Entekhabi (2001a) constructed an adjoint model framework of a coupled land-surface boundary-layer model, which they used to study differences in daytime sensitivity of surface turbulent fluxes for the same model in coupled and uncoupled modes (Margulis and Entekhabi, 2001b). However, their ABL model did not include carbon dioxide nor a height-dependent surface layer. We expect these to be important for our framework that aims to make optimal use of several information streams. For doing gradient calculations within the ICLASS framework, we constructed the adjoint of CLASS.

The paper is structured as follows: First we give some information on the forward model CLASS (Sect. 2). The inverse modelling framework built around CLASS is described in Sect. 3. The *adjoint* of the CLASS model is described separately in Sect. 4. Information on how error statistics are employed and produced follows in Sect. 5, after which we provide a description of the model output (Sect. 6) and technical details of the code (Sect. 7). Afterwards we present the results of the adjoint and gradient tests that serve as validation for the constructed adjoint model (Sect. 8). In Sect. 9 we perform observation system simulation experiments that validate the full inverse modelling framework. In the last section before the concluding discussion we present an example application, for a grassland site in the Netherlands (Cabauw), where a very comprehensive meteorological dataset is complemented with detailed measurements of CO_2 mixing ratios and surface fluxes.

2 Forward model

The employed forward model in our inverse modelling framework is the (slightly adapted) Chemistry Land-surface Atmosphere Soil Slab model (CLASS; Vilà-Guerau De Arellano et al., 2015). The model code is freely available on GitHub (https://classmodel.github.io/). We made use of the Python version of CLASS to construct our inverse modelling framework. We will shortly describe the essentials of the model which are relevant for the inverse modelling framework.

The model consists of several parts, namely the mixed layer, the surface layer and the land surface (Fig. 1). It is a conceptual model that uses a relatively small set of differential equations (Wouters et al., 2019). The core of the model is a box-model representation of an atmospheric mixed layer. Therefore an essential assumption of the model is that during daytime turbulence is strong enough to maintain well-mixed conditions in this layer (Ouwersloot et al., 2012). The mixed-layer tendency equation

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022


Geoscientific Model Development Discussions

for any scalar (e.g. CO_2, heat) is:

$$\text{storage flux} = \frac{(\text{surface flux} + \text{entrainment flux})}{\text{mixed layer height}} + \text{advection} \quad (1)$$

The surface flux is the exchange flux with the land surface (including vegetation and soil). The entrainment flux is the exchange flux between the mixed layer and the overlying free troposphere, and is parameterised in a simple way as a fraction of the surface flux (Stull, 1988, p 478). For moisture and chemical species, a cloud mass flux can also be included in the equation. The mixed layer height is dynamic during the day and evolves under the driving force of the surface heat fluxes and large scale subsidence. Cloud effects on the boundary-layer height and growth due to mechanical turbulence can also be accounted for.

Above the mixed layer a discontinuity occurs in the scalar quantities, representing an infinitely small inversion layer. Above the inversion, the scalars are assumed to follow a linear profile with height in the free troposphere (Fig. 1).

The surface layer is defined in the model as the lowest 10% of the boundary layer. In this (optional) layer Monin-Obukhov similarity theory (Monin and Obukhov, 1954; Stull, 1988) is employed. In the original CLASS surface layer, scalars, the zonal wind speed and the meridional wind speed are evaluated at 2 m height. For some scalars, we have extended this to multiple user-specified heights, as this allows to compare model output to observations of chemical mixing ratios and temperatures at different heights (e.g. along tower). Since the steepness of vertical profiles depends on wind speed and roughness of the surface, these gradients reflect information about these quantities.

The (optional) land surface includes a simple soil representation as well as an a-gs module. This a-gs module (Jacobs, 1994; Ronda et al., 2001) is a big-leaf method to calculate the exchange of CO_2 and H_2O between atmosphere and biosphere. As an alternative for a-gs, a Jarvis-Stewart approach (Jarvis, 1976; Stewart, 1988) can also be used for calculating H_2O exchange. The land surface part is responsible for calculating the exchange fluxes of sensible heat, latent heat and CO_2 between the mixed layer and the land surface. The model has a module for calculating long- and shortwave radiation dynamically. The resulting net radiation is used implicitly in the calculation of the heat fluxes, thereby obtaining a closed energy balance in the model. Soil temperature and moisture are also simulated, based on a force-restore model. The soil heat flux to the atmosphere is calculated based on the gradient between soil and surface temperatures.

More details on the equations in the model can be found in Vilà-Guerau De Arellano et al. (2015). Note that some relatively small changes with respect to the original CLASS model have been implemented, as documented in the ICLASS manual, which is part of the material that can be downloaded via the Zenodo link in the "Code and data availability" section.

3 Inverse modelling framework

3.1 General

Inverse modelling is based on using observations and prior information to statistically optimise a set of variables driving a physical system (Brasseur and Jacob, 2017). The variables to be optimised are contained in a state vector $\boldsymbol{x}$. In our framework, this vector can be subdivided in two other vectors. Those are $\boldsymbol{x_m}$, containing state variables belonging to the input of CLASS (e.g. CO_2 advection, albedo,..), and $\boldsymbol{x_b}$, containing state variables belonging to our bias-correction scheme. The forward-model

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022

Geoscientific Model Development Discussions
Open Access
EGU

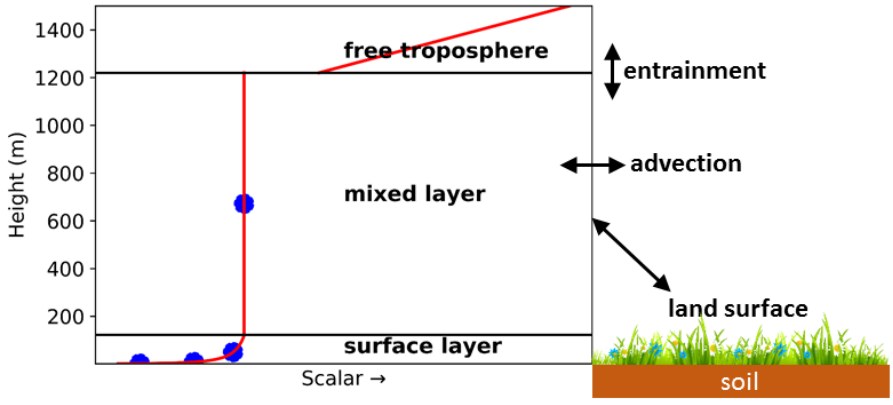


Figure 1. Sketch of the employed forward model: the (slightly adapted) CLASS model. The blue dots in the surface layer represent user-specified heights where the model calculates scalars, e.g. the CO_2 mixing ratio. The mixed layer is represented by a single bulk value in the model (blue dot in mixed layer). At the top of the mixed layer, a discontinuity (jump) occurs in the profiles. The free troposphere is not explicitly modelled, but is taken into account for the exchange with the mixed layer (entrainment). The slope of the free troposphere line is the free troposheric lapse rate. A constant advection can be taken into account as a source/sink.

H (CLASS) projects vector $\boldsymbol{x_m}$ to provide model output that can be compared to observations, e.g. temperatures at different heights (full list in ICLASS manual). This output does not only depend on $\boldsymbol{x_m}$, but also on model parameters that are not part of the state. Those are contained in a vector $\boldsymbol{p}$. The result is contained in vector $H(\boldsymbol{x_m}, \boldsymbol{p})$. We initially define a cost function $J\,(-)$ as (Brasseur and Jacob, 2017):

$$J(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{x_A})^T \mathbf{S_A}^{-1} (\boldsymbol{x} - \boldsymbol{x_A}) + (\boldsymbol{y} - \boldsymbol{H}(\boldsymbol{x_m}, \boldsymbol{p}))^T \mathbf{S_O}^{-1} (\boldsymbol{y} - \boldsymbol{H}(\boldsymbol{x_m}, \boldsymbol{p})) \tag{2}$$

where $\boldsymbol{x_A}$ represents the a-priori estimate of the state vector, $\boldsymbol{y}$ is the vector of observations used within the modelled time window, and $\boldsymbol{H}(\boldsymbol{x_m}, \boldsymbol{p})$ is the vector of model results at the times of the observations. The latter vector is the model equivalent of the observation vector $\boldsymbol{y}$. The superscript T means transpose, $\mathbf{S_A}$ represents the a-priori error covariance matrix and $\mathbf{S_O}$ represents the matrix of observational error covariances. This cost function quantifies two aspects, namely the fit between model output and observations as well as how well the posterior state matches with prior information about the state. The a-priori error covariance matrix $\mathbf{S_A}$ is defined as:

$$\begin{bmatrix} \mathrm{var}((\boldsymbol{x}^{\{t\}} - \boldsymbol{x_A})_1) & \dots & \mathrm{cov}((\boldsymbol{x}^{\{t\}} - \boldsymbol{x_A})_1, (\boldsymbol{x}^{\{t\}} - \boldsymbol{x_A})_n) \\ \vdots & \ddots & \vdots \\ \mathrm{cov}((\boldsymbol{x}^{\{t\}} - \boldsymbol{x_A})_1, (\boldsymbol{x}^{\{t\}} - \boldsymbol{x_A})_n) & \dots & \mathrm{var}((\boldsymbol{x}^{\{t\}} - \boldsymbol{x_A})_n) \end{bmatrix} \tag{3}$$

where $\boldsymbol{x}^{\{t\}}$ is the unknown vector of 'true' values of the parameters in the state vector, and n is the number of variables in the state vector. The matrix has size n x n. Regarding the observational error matrix $\mathbf{S_O}$ (fully defined in Brasseur and Jacob, 2017), we assume for simplicity the observational errors to be uncorrelated (as in e.g. McNorton et al., 2018; Chevallier et al.,

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022


Geoscientific Model Development Discussions

2007; Ma et al., 2021). This simplifies the matrix $\mathbf{S_O}$ to a diagonal matrix, with observational error variances as diagonal elements. This way Eq. (2) simplifies to:

$$J(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{x_A})^T \mathbf{S_A}^{-1} (\boldsymbol{x} - \boldsymbol{x_A}) + \sum_{i=1}^{m} \frac{(H(\boldsymbol{x_m}, \boldsymbol{p})_i - y_i)^2}{\sigma_{O,i}^2} \tag{4}$$

Here, $\sigma_{O,i}^2$ is the i$^{\text{th}}$ diagonal element of $\mathbf{S_O}$, m is the number of observations. It is customary to refer to the first term of this cost function as the background part and to the second part of this function as the data part. Note that if also the a-priori errors are uncorrelated, the first term in the equation can be simplified in a similar way as the first term. The observational error variance linked to the i$^{\text{th}}$ observation ($\sigma_{O,i}^2$) can be further split up as follows:

$$\sigma_{O,i}^2 = \sigma_{I,i}^2 + \sigma_{M,i}^2 + \sigma_{R,i}^2 \tag{5}$$

where $\sigma_{I,i}$ is the instrument (measurement) error, $\sigma_{M,i}$ the model error and $\sigma_{R,i}$ the representation error (see Brasseur and Jacob, 2017). These errors are assumed to be independent of each other.

At this point we introduce two extra features to the cost function. Firstly we allow the user to specify a weight for each individual observation, in case some observations are deemed less important than others. Those weights can also be used to manipulate the relative importance of the background term and the data term. This is similar to the "regularization factor" explained in Brasseur and Jacob (2017). Secondly, we introduce part of our bias-correction scheme in the data part of the cost function, namely scaling factors for observations. These factors can also be optimised. With the additions mentioned above, the cost function as given in Eq. (4) modifies to:

$$J(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{x_A})^T \mathbf{S_A}^{-1} (\boldsymbol{x} - \boldsymbol{x_A}) + \sum_{i=1}^{m} w_i \frac{(H(\boldsymbol{x_m}, \boldsymbol{p})_i - s_i\, y_i)^2}{\sigma_{O,i}^2} \tag{6}$$

where w_i $(-)$ is a weight for each individual observation in the cost function. s_i $(-)$ is a scaling factor for observation y_i, identical for each timestep but differing for each observation stream. The introduction of the scaling factors means we need to adapt the observational error variances as well, the i$^{\text{th}}$ observational error variance is now given by:

$$\sigma_{O,i}^2 = \text{var}(s_i^{\{t\}} y_i - H(\boldsymbol{x_m^{\{t\}}}, \boldsymbol{p})_i) \tag{7}$$

where $\boldsymbol{x_m^{\{t\}}}$ is the unknown vector of 'true' values of the model parameters in the state vector, and $s_i^{\{t\}}$ is the 'true' value of the scaling factor for observation stream i. The decomposition from Eq. (5) remains valid.

In the statistical optimization, we attempt to find the values of the state vector x such that the function in Eq. (6) reaches its absolute minimum. This is done starting from an initial guess ($\boldsymbol{x} = \boldsymbol{x_A}$), after which the state vector is improved iteratively. The cost function and the gradient of the cost function (derivatives with respect to all parameters) are computed for different combinations of parameters in the state vector (Fig. 2). The framework uses by default a truncated Newton method (*tnc;* The SciPy community; Nash, 2000) for the optimisations. This method allows for specifying hard bounds on the state vector parameters, preventing unphysical parameter values for individual parameters in the state vector. Raoult et al. (2016) used similar constraints in their inverse modelling system. The analytical gradient calculations are described in 3.4, a basic numerical derivative option (3.5) is available as well, although we expect this in general to be outperformed by the analytical derivative.

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022

Geoscientific Model Development Discussions
Open Access
EGU

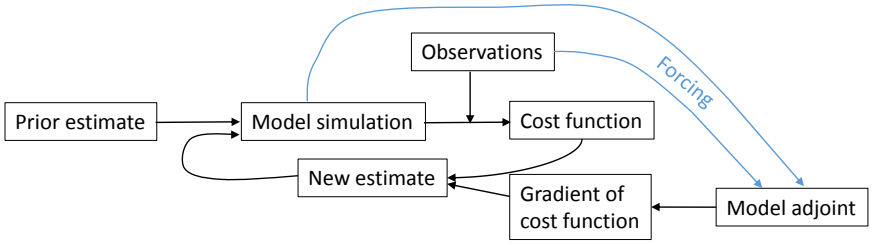


Figure 2. Sketch of the components of the inverse modelling framework when using the adjoint model for the derivative.

3.2 State vector parameters

As mentioned before, the state vector can be decomposed into a vector $\boldsymbol{x_m}$ and a vector $\boldsymbol{x_b}$. Vector $\boldsymbol{x_m}$ contains state variables related to the input of CLASS, such as initial conditions (e.g. mixed-layer potential temperature), boundary conditions (e.g. CO_2 advection), and uncertain model constants (e.g. roughness length for heat). The full list of model parameters that can be optimised is given in the ICLASS manual.

Vector $\boldsymbol{x_b}$ contains parameters belonging to our bias-correction scheme in the data part of the cost function. There are two ways of bias-correcting, the first one is by using observation scaling factors for observation streams. These scaling factors (s_i) have been introduced in Eq. (6). As an example, in the state vector to be optimised, one can include a single scaling factor for all CO_2 surface flux observations. The second possible method of bias correcting is implemented specifically for the energy balance closure problem (Foken, 2008; Oncley et al., 2007; Renner et al., 2019), it involves a parameter "Frac$_H$" $(-)$ that can be optimised. This is the topic of the next section.

3.3 Bias correction for energy balance closure

We first define an observational energy balance closure residual (Foken, 2008):

$$\varepsilon_{\text{eb}} = \boldsymbol{R_n} - (\boldsymbol{H}_{\text{orig}} + \boldsymbol{LE}_{\text{orig}} + \boldsymbol{G}) \tag{8}$$

where $\boldsymbol{R_n}$ is the time series of net radiation measurements, $\boldsymbol{H}_{\text{orig}}$ and $\boldsymbol{LE}_{\text{orig}}$ are the measured sensible and latent heat fluxes and $\boldsymbol{G}$ is the measured soil heat flux. The difference between measured net radiation and the sum of measured heat fluxes is calculated for every time step and this represents the energy balance closure residual. If desired, the user can easily specify an expression of their own for ε_{eb}. Subsequently, the observations for the sensible and latent heat flux are adapted as follows:

$$\boldsymbol{y_H} = \boldsymbol{H}_{\text{orig}} + \text{Frac}_H \; \varepsilon_{\text{eb}} \tag{9}$$

$$\boldsymbol{y_{LE}} = \boldsymbol{LE}_{\text{orig}} + (1 - \text{Frac}_H)\varepsilon_{\text{eb}} \tag{10}$$

This implies that the energy balance closure residual is added partly to the sensible, partly to the latent heat flux. This partitioning is determined by parameter Frac$_H$, which is taken to be constant during the day. This approach of closing the energy

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022

Geoscientific Model Development Discussions

balance is similar to Renner et al. (2019), but we optimise a parameter, instead of using the evaporative fraction for partitioning ε_{eb}.

3.4 Analytical derivative

For the optimisations, we do not only compute a cost function, but we also use the gradient of the cost function with respect to the state vector elements. This informs us on the direction in which the cost function lowers. How the gradient with respect to an individual element is calculated depends on which state vector element is considered. In case the i$^{\text{th}}$ state vector element is a model input parameter, the gradient with respect to this element is computed as (similar to Brasseur and Jacob, 2017, their Eq. 11.105):

$$\frac{\partial J}{\partial x_i} = 2(\mathbf{S_A}^{-1}(\boldsymbol{x}-\boldsymbol{x_A}))_i + 2((\boldsymbol{\nabla_x} H(\boldsymbol{x_m},\boldsymbol{p}))^T \boldsymbol{F})_i \tag{11}$$

Where $\boldsymbol{\nabla_x} H(\boldsymbol{x_m},\boldsymbol{p})$ is the local Jacobian matrix of H and $\boldsymbol{F}$ is the forcing vector, with elements F_k defined as:

$$F_k = w_k \frac{(H(\boldsymbol{x_m},\boldsymbol{p})_k - s_k\, y_k)}{\sigma^2_{O,k}} \tag{12}$$

There is one forcing element in the vector for every observation that is used, F_k is the forcing related to the observation y_k. For calculating this part of the analytical derivative we constructed the adjoint of the model, $(\boldsymbol{\nabla_x} H(\boldsymbol{x_m},\boldsymbol{p}))^T$, which provides us a locally exact analytical gradient (but see Sect. 4.4). More information on the adjoint is given in Sect. 4.

In case the i$^{\text{th}}$ state vector element is an observation scaling factor (s_i from Eq. 6) for observation stream j, the gradient of the cost function with respect to the i$^{\text{th}}$ state vector element is computed as:

$$\frac{\partial J}{\partial x_i} = 2(\mathbf{S_A}^{-1}(\boldsymbol{x}-\boldsymbol{x_A}))_i + \sum_{k=1}^{m_j} -2F_k\, y_{j,k} \tag{13}$$

where m_j is the number of observations of the type (stream) where the observation scaling factor is applied to, e.g. if the observation scale is a scaling factor for surface CO_2 flux observations, m_j is the number of surface CO_2 flux observations. $y_{j,k}$ is the k$^{\text{th}}$ observation of this observation stream. F_k is the forcing related to the observation $y_{j,k}$.

Finally, if the i$^{\text{th}}$ state vector element is Frac_H, the gradient is calculated as

$$\frac{\partial J}{\partial x_i} = 2(\mathbf{S_A}^{-1}(\boldsymbol{x}-\boldsymbol{x_A}))_i - 2\left(\boldsymbol{F_H}\cdot\frac{d\boldsymbol{y_H}}{d\text{Frac}_H} + \boldsymbol{F_{LE}}\cdot\frac{d\boldsymbol{y_{LE}}}{d\text{Frac}_H}\right) \tag{14}$$

Where $\boldsymbol{F_H}$ and $\boldsymbol{F_{LE}}$ are the forcing vectors for the sensible and latent heat flux respectively, $\boldsymbol{y_H}$ and $\boldsymbol{y_{LE}}$ are the observation vectors for the sensible and latent heat flux respectively and "$\cdot$" is the Euclidian inner product. Note that when the Frac_H parameter is included in the state, the observation scaling factors for sensible and latent heat flux observations will be set equal to one, and are not allowed to be included in the state. The terms $\frac{d\boldsymbol{y_H}}{d\text{Frac}_H}$ and $\frac{d\boldsymbol{y_{LE}}}{d\text{Frac}_H}$ represent respectively the derivatives of the sensible and latent heat flux observations to the Frac_H parameter, which follow from equations 9 and 10 as follows:

$$\frac{d\boldsymbol{y_H}}{d\text{Frac}_H} = \varepsilon_{\text{eb}} \tag{15}$$

$$\frac{d\boldsymbol{y_{LE}}}{d\text{Frac}_H} = -\varepsilon_{\text{eb}} \tag{16}$$

Note that equations 11, 13 and 14 all have the same first term that originates from the background part of the cost function.

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022


Geoscientific Model Development Discussions

3.5 Numerical derivative

A simple numerical derivative is available as alternative to the analytical gradient. The derivative of the cost function to the ith state element is numerically calculated as

$$\frac{\partial J}{\partial x_i} = \frac{J(x_i + \alpha) - J(x_i - \alpha)}{2\alpha} \tag{17}$$

where α is a very small perturbation to state parameter x_i, with a default value of 10^{-6}, and has the units of x_i.

3.6 Handling convergence challenges

The highly nonlinear nature of the optimisation problem can cause the optimisation to get stuck in a local minimum of the cost function. This means that the resulting posterior state vector can depend on the prior starting point (Raoult et al., 2016), and the resulting posterior state can remain far from optimal. In the worst case, the non-linearity of the model can even lead to a crash of the forward model. This happens with certain combinations of input parameters, that lead to unphysical situations or undesired numerical behaviour. After starting from a user-specified prior state vector, the *tnc* algorithm autonomously decides on which state vectors are tested during the rest of the optimisation. It is possible to place hard bounds on individual parameters when using the *tnc* algorithm, but this does not always prevent all possible problematic combinations of parameters.

To obtain statistics about the posterior solution, and to deal with the challenges described above, the framework also allows a Monte-Carlo (Tarantola, 2005) approach. This entails that the framework does not start at a single state vector with prior estimates, but instead uses an *ensemble* of prior state vectors $\boldsymbol{x_A}$, leading to an ensemble of posterior parameter estimates. The ensemble of optimisations can be executed in parallel on multiple processors, thereby reducing the time it takes to perform the total optimisation. More details on the Monte-Carlo mode of ICLASS are given in the Sect. 5.2.

As an additional way to improve the posterior solution, we have implemented a "restart" algorithm. If the optimisation results in a cost function that is higher than a user-specified number, the framework will restart the optimisation from the best state reached so far. This "fresh start" of the *tnc* algorithm, whereby the algorithm's memory is cleared, often leads to a further lowering of the cost function. The maximum number of restarts (>=0) is specified by the user. If an ensemble is used, every individual member with a too high posterior cost function will be restarted.

4 Adjoint model

In order to obtain an analytical local derivative, we constructed the adjoint of CLASS. To illustrate the employed technique of adjoint coding, we show in this section some examples of the implementation for a number of forward-model code structures. We have slightly simplified the examples for clarity. For a more detailed background on adjoint models, see Errico (1997) and Giering and Kaminski (1998). Note that the tangent linear code itself is not used in the optimisation framework, only for the construction of the adjoint and for validating the written adjoint code.

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022


Geoscientific Model Development Discussions

4.1 Single line code example

We start with a short example for one line of forward-model (CLASS) code. For this we chose the calculation of net radiation:

```
Q = Swin - Swout + Lwin - Lwout
```

Where *Swin*, *Swout*, *Lwin* and *Lwout* are the incoming shortwave, outgoing shortwave, incoming longwave and outgoing longwave radiation respectively. We construct the tangent linear code of this statement as follows. We first calculate the derivatives of Q to the variables at the RHS of the equation. We then multiply these derivatives with a perturbation in the parameter to which we took the derivative. Finally, we add everything up, resulting in the following expression:

```
dQ = dSwin - dSwout + dLwin - dLwout
```

where *dSwin*, *dSwout*, *dLwin* and *dLwout* are small perturbations to variables *Swin*, *Swout*, *Lwin* and *Lwout* respectively. *dQ* is the change in Q due to these perturbations. Generally, this tangent linear equation is exact only if the perturbations are infinitesimally small, otherwise the result is a linear approximation to the change in Q. In other words, the tangent linear model provides an exact approximation to the change in Q at location (*Swin*=*Swin*$_0$, *Swout*=*Swout*$_0$,*Lwin*=*Lwin*$_0$,*Lwout*=*Lwout*$_0$) in parameter space. Note that for this case the forward-model equation is linear, meaning that the tangent linear of the forward-model statement provides an exact approximation for any perturbation size. For general adjoint coding a matrix notation of a tangent linear statement is convenient:

$$\begin{bmatrix} dSwin \\ dSwout \\ dLwin \\ dLwout \\ dQ \end{bmatrix}_{\mathrm{k}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & -1 & 1 & -1 & 0 \end{bmatrix} \times \begin{bmatrix} dSwin \\ dSwout \\ dLwin \\ dLwout \\ dQ \end{bmatrix}_{\mathrm{k-1}} \tag{18}$$

Here, k-1 and k are indices for the corresponding position in the forward-model code, i.e. before and after execution of the forward-model code line respectively. Now we switch from the tangent linear variables *dSwin* etc. to the adjoint variables *adSwin*, *adSwout*, *adLwin*, *adLwout* and *adQ*. While doing so we also transpose the matrix with numbers:

$$\begin{bmatrix} adSwin \\ adSwout \\ adLwin \\ adLwout \\ adQ \end{bmatrix}_{\mathrm{k-1}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} adSwin \\ adSwout \\ adLwin \\ adLwout \\ adQ \end{bmatrix}_{\mathrm{k}} \tag{19}$$

Note that we have also reversed the indices k-1 and k now (Giering and Kaminski, 1998), the adjoint model is "reversed" compared to the tangent linear and forward models. From the matrix we obtain five adjoint equations:

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022


Geoscientific Model Development Discussions

```
adSwin = adQ + adSwin
adSwout = -adQ + adSwout
adLwin = adQ + adLwin
adLwout = -adQ + adLwout
adQ = 0
```

Observations of variable Q could then be used to obtain the model-observation mismatch, which can subsequently be used in the cost function and forcing vector (Eq. 12). The forcing vector can subsequently be used to force the adjoint model (Eq. 11). In that case, the variables *adSwin*, *adSwout*, *adLwin* and *adLwout* contain the derivatives of the cost function to the incoming shortwave, outgoing shortwave, incoming longwave and outgoing longwave radiation respectively. Regarding the physical units, when the units of the forward-model variable are Wm^{-2} (e.g. *Swin*), the units of the adjoint variable (e.g. *adSwin*) will be $\mathrm{m}^2\mathrm{W}^{-1}$.

4.2 Multiple lines code

For two consecutive forward-model statements, two consecutive tangent linear statements are constructed using the principle described above. Two adjoint statements are also constructed, however, those statements are placed in reverse order. The forward model consists of different Python functions, and we wrote a separate tangent linear and adjoint function for each of these functions. In the forward model a timestep includes a sequence of calls to these functions. In the tangent linear the same sequence of calls to the tangent linear functions belonging to the forward-model functions is executed. In the adjoint model however, the adjoint functions belonging to the tangent linear functions are called in reverse order. For example, when in the forward model calls are made to the statistics, radiation and surface layer functions, then the adjoint will call the adjoint surface layer function before the adjoint radiation and adjoint statistics functions. The adjoint model also runs backward in time, in contrast to the forward model or tangent linear model, so the order of the time steps is also reversed.

Note that there is no need to calculate derivatives of all model variables, only of those that depend on one of the possible state variables and have a possible influence on the cost function. Those are our "active variables" (Giering and Kaminski, 1998). Similarly, we only need to calculate derivatives with respect to possible state variables and to variables depending on those state variables.

4.3 Storing variables and functions with adjoint arguments

Some model statements are slightly more complex, requiring additional techniques. The first one is the storing of the value of forward-model variables, for use in the tangent linear and adjoint models. As an example, take the following line of forward-model code:

```
ueff = np.sqrt(u ** 2. + v ** 2. + wstar**2.)
```

Where *u* and *v* are the zonal and meridional wind components, *wstar* is the convective velocity scale (Stull, 1988) and *np.sqrt* is a function that calculates the square root of a variable. In the tangent linear model the statement becomes:

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022


Geoscientific Model Development Discussions

```
dueff = 0.5 * (u**2. + v**2. + wstar**2.)**(-0.5) *
(2 * u * du + 2 * v * dv + 2 * wstar * dwstar)
```

Where *du*, *dv* and *dwstar* are perturbations just as in the simple example. We give also the corresponding adjoint code here:

```
adu += 0.5 * (u**2. + v**2. + wstar**2.) **(-0.5) * 2 * u * adueff
adv += 0.5 * (u**2. + v**2. + wstar**2.) **(-0.5) * 2 * v * adueff
adwstar += 0.5 * (u**2. + v**2. + wstar**2.) **(-0.5) * 2 * wstar * adueff
adueff = 0
```

Note that *u*, *v* and *wstar* occur also in the tangent linear and adjoint equations. This means that we need to have the values of these variables at all modelled times and iterations available for use in the adjoint and tangent linear models. As the adjoint should provide us a local analytical derivative, we need to have the value of the model variables at exactly the point in state space where we want to calculate the gradient. Therefore, before every call to the adjoint model, we store the necessary information while running the forward model at the same point in state space as the upcoming adjoint call. Since our memory requirements for this are not excessive, we can store all required variables at all time steps and there is no need to employ a more advanced checkpointing scheme (Charpentier, 2001).

The second technique we want to illustrate is the use of a function with adjoint arguments. We take the following example from the forward-model code:

```
esatvar     = esat(theta)
```

Which calculates the saturation vapour pressure based on mixed-layer potential temperature (*theta*). It refers to a function *esat*, which is a simple function given below:

```
def esat(T):
    return 0.611e3 * np.exp(17.2694 * (T - 273.16) / (T - 35.86))
```

Where *T* is temperature and np.exp refers to the exponential function from the Numpy (van der Walt et al. (2011); http://www.numpy.org) library. For the tangent linear model this leads to the following code:

```
def desat(T,dT):
    return 0.611e3 * np.exp(17.2694 * (T - 273.16) / (T - 35.86)) *
    (-17.2694)*(35.86-273.16)/(T-35.86)**2 * dT
desatvar     = desat(theta,dtheta)
```

For the adjoint code, there is no need to write an adjoint function for *desat*, instead we can write the adjoint code as follows:

```
adtheta += desat(theta,adesatvar)
adesatvar = 0
```

where function *desat* is called with variable *adesatvar* as second argument, instead of the argument *dtheta* that is used in the tangent linear model.

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022


Geoscientific Model Development Discussions

4.4 If-statements

The adjoint code provides us in principle a locally exact analytical derivative. However, since the equations implemented in the model are not all continuous, they are consequently not all differentiable. For example, the following model statement occurs in the forward-model code:

```
ueff            = max(0.01, np.sqrt(u**2. + v**2. + wstar**2.))
```

Now if *v* and *wstar* are 0 and *u* equals 0.01, the derivative $\frac{dueff}{du}$ is undefined, since the left derivative and right derivative are not equal, i.e. the left derivative (moving to lower values of u) equals zero and the right derivative (moving to higher values of *u*) equals 1. We have written the following tangent linear code belonging to this model statement (only *du* term shown here):

```
if np.sqrt(u**2. + v**2. + wstar**2.) < 0.01:
    dueff = 0
else:
    dueff = 0.5 * (u**2. + v**2. + wstar**2.)**(-0.5) * (2 * u * du)
```

Which means that at the point $u = 0.01\ \mathrm{ms}^{-1}$, we have chosen to implement the right derivative instead of the left derivative. This is however an arbitrary choice, the right derivative is the correct one to use in case we (or the optimisation algorithm) are interested in the behaviour of *ueff* for increasing *u*, in the other case the left derivative is the correct one. This is a potential source of error for the written adjoint code, however, we have to keep in mind that (in this specific example) the calculated derivative is only (possibly) wrong at the exact value of $u = 0.01$. Furthermore, the model is very non-linear anyway, which makes the calculated gradients only valid locally.

4.5 Loops

For-loops in the forwardmodel result in for-loops in the tangent linear and adjoint models. As an example, the following forward-model code:

```
for i in range(nr_of_surf_lay_its):
    run_surface_layer(call_from_init=True,iterationnumber=i)
```

results in the following tangent linear code:

```
for i in range(nr_of_surf_lay_its):
    tl_run_surface_layer(model,checkpoint_init[i])
```

and the corresponding adjoint code:

```
for i in range(nr_of_surf_lay_its-1,-1,-1):
    adj_run_surface_layer(forcing,checkpoint_init[i],model)
```

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022


Geoscientific Model Development Discussions

Note that if the for-loop would contain multiple statements, the order would be reversed in the adjoint code. The forward model also contains a while-loop. This loop can be transformed into a for-loop in the tangent linear and adjoint code (and dealt with similarly as above), when the number of performed iterations for this loop is known. Therefore the value of the variable counting the number of iterations performed in the while-loop is stored during a forward-model run. As mentioned earlier, the forward model is run before calling the adjoint, so we have the necessary information.

5 Error statistics

5.1 Prior and observations

From a Bayesian point of view, ICLASS combines information, both from observations and from prior knowledge about the state vector, to come to a solution with a reduced uncertainty in the state parameters. In the derivation of Eq. (2), it is assumed that both prior and observational errors follow a (multivariate) normal distribution (Tarantola, 2005). However, some prior input parameters are bounded (Sect. 3.6), e.g. the albedo cannot be negative. Dealing with bounds is a known challenge in inverse modelling, several approaches can be followed (Miller et al., 2014). As an example, Bergamaschi et al. (2009) had to deal with methane emissions in the state vector becoming negative. Their solution was to introduce an emission parameter that can be optimised, instead of the emissions themselves. In our case such an approach is more difficult, as we have a diverse set of bounded parameters.

Our approach is to enforce hard bounds for state values via the *tnc* algorithm, it however means that the normality assumption will be violated to some extent, as the normal distribution (for which the user specifies the variance) for prior parameter values then becomes a truncated normal distribution.

Our system also allows for specifying covariances between state elements in $\mathbf{S_A}$. We assume observational errors to be uncorrelated, see Sect. 3.1. Equation 5 states that the observational error consists of an instrument error, a model error and a representation error. The instrument and representation error are taken from user input, the model error can either be specified by the user or estimated from a sensitivity analysis. In the latter case, an ensemble of forward-model runs is performed. For each of these runs, a perturbed set of parameters, not belonging to the state vector, is used as model input (so parameters part of vector $\boldsymbol{p}$, not from $\boldsymbol{x}$, see Sect. 3.1). The user should specify which parameters should be perturbed, and the distribution from which random numbers will be sampled to add to to the default (prior) model parameters. For the latter, there is the choice between a "normal", "bounded normal", "uniform" or "triangular" distribution. The model error for each observation stream at each observation time is then obtained from the spread in the ensemble of model output for the observation stream at that time.

5.2 Ensemble mode to estimate posterior errors

When ICLASS is run in ensemble mode, it allows for an estimation of the error on the posterior state. The principle for obtaining the posterior errors bears strong similarity to what has been done in Chevallier et al. (2007), it will be shortly explained here. We start with the prior state specified by the user. Then, for each ensemble member and for every state parameter, a

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022


Geoscientific Model Development Discussions

random number is drawn from a normal distribution with variance of the respective parameter specified by the user, and mean 0. When also covariances are specified for the prior parameters, a multivariate normal distribution can be used to sample from. The sampled numbers are then added to the state vector. In case of non-zero covariances, if the new state vector falls out of bounds (if specified), it is discarded and the procedure repeated for that ensemble member. In case no covariance structure is present, only the parameter which falls out of bounds has to be replaced. It has to be noted that this effectively leads to sampling from a truncated (multivariate) normal distribution.

Similarly, for every ensemble member a random number drawn from a normal distribution (without bounds) is added to the (scaled) observations. This modifies the cost function of an individual ensemble member (Eq. 6) as follows:

$$J(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{x_A})^T \mathbf{S_A}^{-1} (\boldsymbol{x} - \boldsymbol{x_A}) + \sum_{i=1}^{m} w_i \frac{(H(\boldsymbol{x_m}, \boldsymbol{p})_i - s_i\, y_i + \varepsilon_i)^2}{\sigma_{O,i}^2} \tag{20}$$

The random numbers ε_i are sampled from normal distributions with mean zero and standard deviation $\sigma_{O,i}$. This can be seen as either a perturbation of the scaled observations or a perturbation of the model-data mismatch. As a consequence of the change in the cost function equation, the equation for the forcing vector (Eq. 12) is updated as well:

$$F_k = w_k \frac{(H(\boldsymbol{x_m}, \boldsymbol{p})_k - s_k\, y_k + \varepsilon_k)}{\sigma_{O,k}^2} \tag{21}$$

For simplicity, the vector $\boldsymbol{\varepsilon}_{\text{eb}}$ (Eq. 8) is not perturbed and kept identical for all members. For every ensemble member an optimisation is performed. Each optimisation is classified as either successful or not successful. Here, successful is defined as having a final chi-squared (Eq. 22) smaller or equal than a user-specified number. The posterior state parameters for the successful members are saved in a matrix. A covariance and a correlation matrix is then constructed using the matrix of posterior state parameters. These covariance and correlation matrices are used as an estimate of the true posterior state covariance and correlation matrices, respectively. If the user prefers so, also parameters not belonging to the state can be perturbed in the ensemble, in a similar way as outlined in Sect. 5.1.

Note that, next to the ensemble, there will also be a run with an unperturbed prior, just as is the case when no ensemble is used. We refer to this run as member 0. This member is however not included in the calculation of ensemble-based error statistics (e.g. correlation matrix), as the choice of prior values is not random for this member.

6 Output

ICLASS can write several output files. The output includes the obtained parameters, as well as the posterior chi-squared statistic and the posterior and prior cost function. The chi-squared goodness-of-fit statistic is defined in ICLASS as:

$$\chi^2 = \frac{J}{\sum_{i=1}^{m}(w_i) + n} \tag{22}$$

In this equation, m is the number of observations and n the number of state parameters (see Sect. 3 for the other symbols). Note that if all the weights are set to 1, the denominator simplifies to m+n. The "n" will only be included in the denominator when the user chooses to include a background part in the cost function (default). The χ^2 statistic for the posterior solution

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022


Geoscientific Model Development Discussions

should be around 1 when the optimisation converges well and errors are properly specified (Brasseur and Jacob, 2017, their Sect. 11.5.4). ICLASS also calculates a prior and a posterior partial cost function and a posterior χ^2 statistic for individual observation streams and for the correspondence to prior information (background). The partial cost function for observation stream j is calculated as:

$$J_j = \sum_{i=1}^{m_j} w_i \frac{(H(\boldsymbol{x_m}, \boldsymbol{p})_i - s_j\, y_{j,i} + \varepsilon_i)^2}{\sigma_{O,i}^2} \tag{23}$$

where m_j is the number of obs of stream j, $y_{j,i}$ is the i$^{\text{th}}$ observation of stream j, and $H(\boldsymbol{x_m}, \boldsymbol{p})_i$ is the model output corresponding with $y_{j,i}$. In case scaled observations are perturbed in the ensemble (Sect. 5.2), ε_i is a random number, otherwise it is zero. The χ^2 statistic for observation stream j is defined here as:

$$\chi_j^2 = \frac{J_j}{\sum_{i=1}^{m_j}(w_i)} \tag{24}$$

Which resembles Eq. (13) of Meirink et al. (2008), but note that $\frac{\chi^2}{n_s}$ in their paper is a similar variable as χ_j^2 in our paper. The χ^2 statistic for the background part is calculated as:

$$\chi_b^2 = \frac{J_b}{n} \tag{25}$$

Where J_b is the background part of the cost function, i.e. term 1 from Eq. (6). The mean bias error, root mean squared error and the ratio of model and observation variance for every observation stream is also calculated, both for the prior and posterior state. The mean bias error for the j$^{\text{th}}$ observation stream is defined here as:

$$\varepsilon_{\text{bias},j} = \frac{1}{m_j} \sum_{i=1}^{m_j} (H(\boldsymbol{x_m}, \boldsymbol{p})_{j,i} - s_j\, y_{j,i}) \tag{26}$$

In this equation, m_j is the number of obs of type (stream) j, $H(\boldsymbol{x_m}, \boldsymbol{p})_{j,i}$ is the model output for observation stream j at time index i, $y_{j,i}$ is an observation of stream j at time index i and s_j is an observation scaling factor for obs of stream j. Note that even when the scaled observations are perturbed, we do not include those perturbations here (compare Eq. 23). In case the Frac_H parameter is used, the energy balance corrected observations (equations 9 and 10) will be used in the equation above. For the root mean squared error and the ratio of model and observation variance, the observation scales and energy balance corrected observations are used as well. ICLASS also calculates the normalised deviation of the posterior from the prior for every state parameter i:

$$\delta_{\text{nor},i} = \frac{x_{\text{post},i} - x_{A,i}}{\sigma_{A,i}} \tag{27}$$

where $\sigma_{A,i}$ is the square root of the prior variance of parameter i, i.e. the square root of diagonal element i from the matrix in Eq. (3). Note that, also in case of an ensemble, we use the unperturbed prior, i.e. the prior of member 0.

If run in ensemble mode, we additionally store estimates of the posterior error covariance matrix (Sect. 5), the posterior error correlation matrix, the posterior/prior variance ratio for each of the state parameters, the mean posterior state, and the

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022

optimised and prior states for every single ensemble member. When non-state parameters are perturbed in the ensemble, these parameters are part of the output, also the correlation of the posterior state parameters with these non-state parameters can be written as output. When the model error is estimated by ICLASS (Sect. 5.1), there is a separate file containing statistics on the estimated errors. Finally, there are additional output files with information about the optimisation process. For every model simulation (and for every ensemble member) in the iterative optimisation process, one can find the parameter values used in this simulation as well as the value of the cost function, split up into a data and a background part. For the gradient calculations, one can find the parameter values used, as well as the the derivatives of the cost function with respect to every state parameter. The derivatives of the background part are also provided separately. More details on the output of ICLASS can be found in a separate section of the manual.

7 Technical details of the code

We have written the entire code in Python 3 (https://www.python.org). Using the framework requires a Python 3 installation, including the NumPy (van der Walt et al. (2011); http://www.numpy.org), SciPy (https://scipy.org/) and Matplotlib (Hunter (2007); https://matplotlib.org) libraries. The code is operating-system independent and consists of three files. "forward-model.py", contains the adapted CLASS model, "inverse_modelling.py", contains most of the inverse modelling framework, such as the function to be minimised in the optimisation and the model adjoint. Finally, the file "optimisation.py" is the file to be run by the user for performing an actual optimisation, this file reads in observations, defines the state vector etc. The input paragraphs in this file should be adapted by the user to the optimisation performed. There are three additional files, the first is ‘optimisation_OSSE.py’, and is similar to optimisation.py, but is meant specifically for observation system simulation experiments (also in this file, the user should adapt the input paragraphs to the optimisation to be performed). The second file is "testing.py", containing tests for the code as described in Sect. 8. The file "postprocessing.py" is a script that can be run after the optimisations are finished, for post-processing output data, e.g. to plot a colored matrix of correlations. Using the Pickle module, the optimisation can store variables on the disk near the end of the optimisation, and these variables can be read in again in the "postprocessing.py" script. This has the advantage that e.g. the formatting of plots can easily be adapted without redoing the optimisation. This script should be adapted by the user to the optimisation performed and the output desired.

8 Adjoint model validation

When using the adjoint modelling technique, an extensive testing system is required to make sure that the analytical gradient of the cost function computed by the adjoint model is correct. There are two tests that are essential, which are described below.

The gradient test (for the tangent linear model) is a test to determine whether the derivatives with respect to the model input parameters in the tangent linear model are constructed correctly. The construction of the adjoint is based on the tangent linear model, so errors in the tangent linear propagate to the adjoint model. In the gradient test, model input state variables are perturbed, which leads to a change in model output. The change in model output when employing the tangent linear model is

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022

Geoscientific Model Development Discussions

compared to the change in model output when a numerical finite difference approximation is employed. Mathematically, the test can be written as (similar to Honnorat et al., 2007):

$$\frac{dH(\boldsymbol{x_m},\boldsymbol{p})}{d\boldsymbol{x_m}} \cdot \boldsymbol{\Delta x_m} \approx \frac{H(\boldsymbol{x_m}+\alpha\boldsymbol{\Delta x_m},\boldsymbol{p}) - H(\boldsymbol{x_m},\boldsymbol{p})}{\alpha} \tag{28}$$

where $\boldsymbol{\Delta x_m}$ is a vector of ones, with the same length as vector $\boldsymbol{x_m}$. α is a small positive number, H represents the forward-model operator, $\boldsymbol{x_m}$ is the vector of model input variables to be tested (state), "$\cdot$" is the Euclidian inner product, and vector $\boldsymbol{p}$ is the set of non-state parameters used by the model. Several increasingly smaller values are tested for α. When the tangent linear model is correct, the right-hand side of the equation will converge to the left-hand side when α gets progressively smaller, although for too small α numerical errors start to arise (Elizondo et al., 2000). Instead of using the full tangent linear model, individual model statements can also be checked. In this case $\boldsymbol{x_m}$ and $H(\boldsymbol{x_m},\boldsymbol{p})$ can contain intermediate (not part of model input or output) model variables. The gradient test is considered successful if the ratio of the left- and right-hand sides of the equation lies in the interval [0.999 - 1.001].

The dot-product (adjoint) test checks whether the adjoint model code is correct for the given tangent linear code. It tests whether the identity

$$\langle \mathbf{H_L}\boldsymbol{x_m}, \boldsymbol{y}\rangle = \langle \boldsymbol{x_m}, \mathbf{H_L^T}\boldsymbol{y}\rangle \tag{29}$$

holds (Krol et al., 2008; Meirink et al., 2008; Honnorat et al., 2007; Claerbout, 1992). The equation can also be written as

$$\frac{\langle \mathbf{H_L}\boldsymbol{x_m}, \boldsymbol{y}\rangle - \langle \boldsymbol{x_m}, \mathbf{H_L^T}\boldsymbol{y}\rangle}{\langle \mathbf{H_L}\boldsymbol{x_m}, \boldsymbol{y}\rangle} = 0 \tag{30}$$

In this equation, $\mathbf{H_L} = \boldsymbol{\nabla}_{\boldsymbol{x_m}} H(\boldsymbol{x_m},\boldsymbol{p})$ represents the tangent linear model operator, i.e. a matrix (the local Jacobian of H) with the element on the i$^{\text{th}}$ row and j$^{\text{th}}$ column given by $\frac{dH(\boldsymbol{x_m},\boldsymbol{p})_i}{dx_{m,j}}$. $\boldsymbol{x_m}$ is in this equation a vector with perturbations to the model state, $\boldsymbol{y}$ is here a vector containing perturbations to observations for $H(\boldsymbol{x_m},\boldsymbol{p})$, $\mathbf{H_L^T}$ represents the adjoint model operator and "$\langle\rangle$" is the Euclidian inner product. A very small deviation from 0 is acceptable due to machine rounding errors (Claerbout, 1992). Our criterion for passing the test, when using 64-bit floating-point calculations, is that the absolute value of the left-hand side of the equation should be $<= 5 \times 10^{-13}$. This test can also be applied to individual statements or a block of statements in the adjoint model code. In this case the definitions of the variables in Eq. 30 slightly change, e.g. $\mathbf{H_L^T}$ then represents a part of the adjoint model. The adjoint test will be illustrated by a (slightly simplified) example from the model code, the well-known Stefan–Boltzmann's law for outgoing longwave radiation. The forward-model code reads:

```
Lwout = bolz * Ts ** 4.
```

The tangent linear model code for this statement reads:

```
dLwout = bolz * 4 * Ts ** 3. * dTs
```

And the corresponding adjoint code:

```
adTs += bolz * 4 * Ts ** 3. * adLwout
adLwout = 0
```

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022


Geoscientific Model Development Discussions

In this case $\boldsymbol{x_m}$ corresponds to *dTs*, the $\mathbf{H_L}\boldsymbol{x_m}$ variable is *dLwout*. The vector $\boldsymbol{y}$ is *adLwout* and $\mathbf{H_L^T}\boldsymbol{y}$ is *adTs*. In the test $\boldsymbol{x_m}$ and $\boldsymbol{y}$ are assigned random numbers. When we evaluated Eq. (30) on this part of the code, the result was less than 1×10^{-15}, meaning that the test passes.

We have constructed a separate script that performs a vast amount of gradient tests and adjoint tests and informs the user whenever a test fails. This file (testing.py) is included with the code files. The number of time steps we specified for testing is small, as the computational burden increases with the amount of time steps. The model passes the vast majority of the tests that are executed in this file. Closer inspection of the tests that fail reveal that numerical noise is a likely explanation for the small fraction of tests that are labeled as failing. Further validation of the inverse modelling framework follows in the next section.

9 Inverse modelling validation: Observation System Simulation Experiments

To test the ability of the system to properly estimate model parameters, we show in this section the results of observation system simulation experiments with artificial data. Similar tests have been performed by Henze et al. (2007), for their constructed adjoint of a chemical transport model. In our experiments we simulate a growing convective boundary layer for a location at mid-latitudes from 10–14 h. In the chosen setup, the land surface is coupled to the boundary layer. The land surface provides heat and moisture, and exchanges CO_2 with the ABL. We perform a total of 5 main experiments of varying complexity, the things that differ among experiments, are the choice of observations and prior state vectors. The procedure for the first four experiments is as follows. We first run the model with chosen values of a set of parameters we want to optimise. A set of model output data from this simulation then serve as the observations, while the parameters used to create these observations are referred to as the "true" parameters. Then we perform an optimisation using these observations, starting from a perturbed prior state vector. In the cost function, we do not include the background part, as that would send us away from finding the "true" parameters. In the ideal case, the framework should be able to find back the parameter values that were used for creating the observations, starting from any other prior state vector.

Since real-life observations usually contain noise, the last experiment follows the same procedure, except that the observations are also perturbed. This is done by adding white Gaussian noise to the observations. For each observation, a random number drawn from a normal distribution with mean zero and standard deviation equal to a specified measurement error for the respective observation, is added. Table 1 list for every experiment the prior, the "true" and the optimised state variables. The complexity of the experimental setup increases from the top to the bottom of the table.

For the first experiment we perturbed the initial boundary-layer height and albedo. The rest of the parameters are unchanged compared to the true parameters. The observation streams we used are specific humidity and boundary-layer height. The specific humidity is expected to be influenced by the albedo, as the amount of available net radiation is relevant for the amount of evapotranspiration, which influences the specific humidity. The size of the boundary layer is relevant for specific humidity as well, as it determines the size of the mixing volume in which evaporated water is distributed. The state parameters are thus expected to have a profound influence on the cost function.

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022

Geoscientific Model Development Discussions

Table 1. Parameter values for the observation system simulation experiments. The observation streams used in the 2-obs-streams experiments are q and h, for the 6 obs streams experiment H, LE, T_2 and F_{CO2} are added, for the 7-obs-streams experiments additionally $CO2_{20}$ is added. The value of e.g. parameter z_{0m} (only a state element in the 10-parameter-state experiment) is in the 2-parameter-state and 5-parameter-state experiments equal to the true value of the 10-parameter state experiment. See Table A1 for a description of the parameters, and Table A2 for a description of the observation streams.

Parameter	True	Prior	Optimised	
	2-parameter state		2 obs streams	
h (m)	350.0	650.0	350.0	
α_{rad} $(-)$	0.200	0.450	0.200	
	5-parameter state		2 obs streams	6 obs streams
h (m)	350.0	650.0	350.0	350.0
α_{rad} $(-)$	0.200	0.450	0.200	0.200
α_{sto} $(-)$	1.00	0.50	0.96	1.00
w_g $(-)$	0.270	0.140	0.306	0.270
γ_θ (K m^{-1})	0.0030	0.0050	0.0030	0.0030
	10-parameter state		7 obs streams	
			no noise	noise
h (m)	350.0	650.0	350.0	344.4
α_{rad} $(-)$	0.200	0.450	0.200	0.207
α_{sto} $(-)$	1.00	0.50	1.00	1.01
w_g $(-)$	0.270	0.140	0.270	0.238
γ_θ (K m^{-1})	3.0×10^{-3}	5.0×10^{-3}	3.0×10^{-3}	3.1×10^{-3}
γ_q $(\text{kg kg}^{-1}\text{m}^{-1})$	-1.0×10^{-6}	-3.0×10^{-6}	-1.0×10^{-6}	-1.1×10^{-6}
CO_2 (ppm)	422.0	380.0	422.0	423.2
adv_{CO2} (ppm s^{-1})	0.0	6.0×10^{-3}	9.8×10^{-8}	-1.4×10^{-4}
z_{0m} (m)	0.020	0.100	0.020	0.048
z_{0h} (m)	0.020	0.010	0.020	0.013

The optimised model shows a very good fit to the observations (Fig. 3), and the original parameter values are found back with a precision of at least 5 decimal places (Table 1, no 5 decimal places shown). This result proves the optimisations work in such simple situations.

To test the system for a more complex problem, we change from two parameters in the state to five (Table 1, second block), while keeping the same observations. Again, based on physical reasoning, the added parameters are expected to influence the cost function. However, one might expect that the observations might not be able to uniquely constrain the state, i.e. parameter interdependency issues might arise for this setup. As an example, a reduction in soil moisture might influence the observations

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022


Geoscientific Model Development Discussions

in a similar way as a reduction in stomatal conductance. Mathematically, this means that multiple nearly equivalent local minima might be present in the cost function.

Also for this more complex case, we manage to get a very good fit (Fig. 3 c and d). However, inspecting the obtained parameter values (Table 1, second block, column "2 obs streams") we see that the optimization does not fully converge to the "true" state. This is likely caused by parameter interdependency, as described earlier. To solve this, extra observation streams need to be added that allow to disentangle the effects of the different parameters in the state. We have tested this in the third experiment, by adding the sensible and latent heat flux, the temperature at 2 m height and the surface CO_2 flux to the observations. With these more complete set of observations, the true parameter values are found back with a precision of at least 4 decimal places (column "6 obs streams" in Table 1). As a side experiment, not included in Table 1, we have run the same experiment as before, but using the numerical derivative described in Sect. 3.5. In this case, convergence is notably slower, with e.g. more than six times as many iterations needed to reduce the cost function to less than 0.1 % of its prior value. This indicates that, at least for this more complex non-linear optimisation case, an analytical gradient outperforms our simple numerical gradient calculation.

In the fourth experiment, we increase our number of state parameters to 10 (Table 1). To further constrain the parameters, we expand our set of observation streams with the CO_2 mixing ratio measured at 20 m height. Also for this complex setup the framework managed to find the true values back, with a precision of one decimal place for initial boundary-layer height, and more for the other parameters.

These tests show that (at least for the range of tests performed) the framework is able to find the minimum of the cost function well. Finally, we performed another observation system simulation experiment to test the framework's usefulness in a more realistic situation with noise on the observations. The result can be seen in Figure 4, the framework finds a good fit to the observations, even though it has now become impossible to fit every single (perturbed) observation. The χ^2 statistic of the optimisation has become close to 1 (0.83), indicating no strong over- or under-fitting of the observations. As our system performs well in the tests with artificial data, we will show an application to a measured dataset in the next section.

10 Application example: CO_2 and boundary layer meteorology at a Dutch grassland

For this example of how the framework can be used, we used data obtained at Cabauw, the Netherlands from 25 September 2003. A 213 m high measurement tower is present at this location (Bosveld et al., 2020). This day is chosen here since it has been used in several studies before (Vilà-Guerau De Arellano et al., 2012; Casso-Torralba et al., 2008) and we consider it as a "golden day" (Sect. 1), for which the CLASS model is expected to perform at its best.

The set of observation streams we used is given in Table 2, and a description of these streams is given in Table A2. Some of the observations we used are not directly part of the Cabauw dataset, we have derived them from other observations in the same dataset. The parameters we optimise (the state) are given in Table 3, and a description of these parameters can be found in Table A1. The model settings, prior state and non-state parameters are to a large extent based on Vilà-Guerau De Arellano et al. (2012). The modelled period is from 9–15 UTC. We ran ICLASS in ensemble mode with 175 members, an ensemble

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022

Geoscientific Model Development Discussions
Open Access
EGU

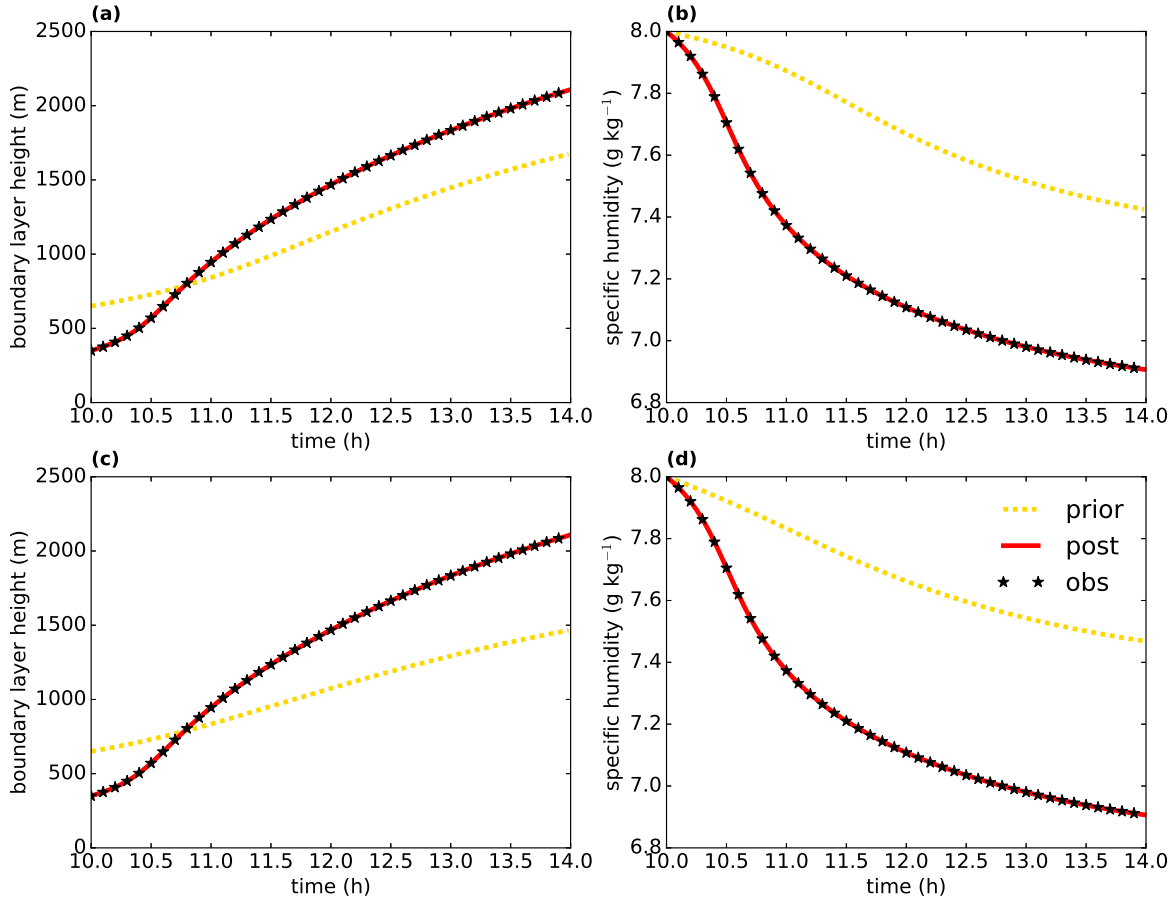


Figure 3. Model output from two observation system simulation experiments: 2-parameter state experiment with albedo and initial boundary-layer height (top row), and 5-parameter state experiment with two observation streams (bottom row). "post" means posterior, i.e. the optimised simulation. Even though the posterior lines look very similar for the bottom and top rows, the model parameters are not identical.

member was considered successful when having a final $\chi^2 <= 2.0$ (Eq. 22). The detailed settings on chosen model errors etc. can be found via the Zenodo link in the "Code and data availability" section.

Figures 5 and 6 show that the optimised run shows a much better fit than the prior run to the subset of observations present in these figures. For temperature, a total of 7 observation heights are included, but we only show 3 for brevity. The cost function is reduced from a value of 4702 to 126. The χ^2 goodness of fit statistic has a value of 0.80 for the optimised run, indicating a slight overfitting (or non-optimal error specifications). For all observation streams, the root mean squared error (Sect. 6, Table 2) and the mean bias error (Eq. 26, not shown) are reduced after optimisation. The ratio of model and observation variance

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022

Geoscientific Model Development Discussions

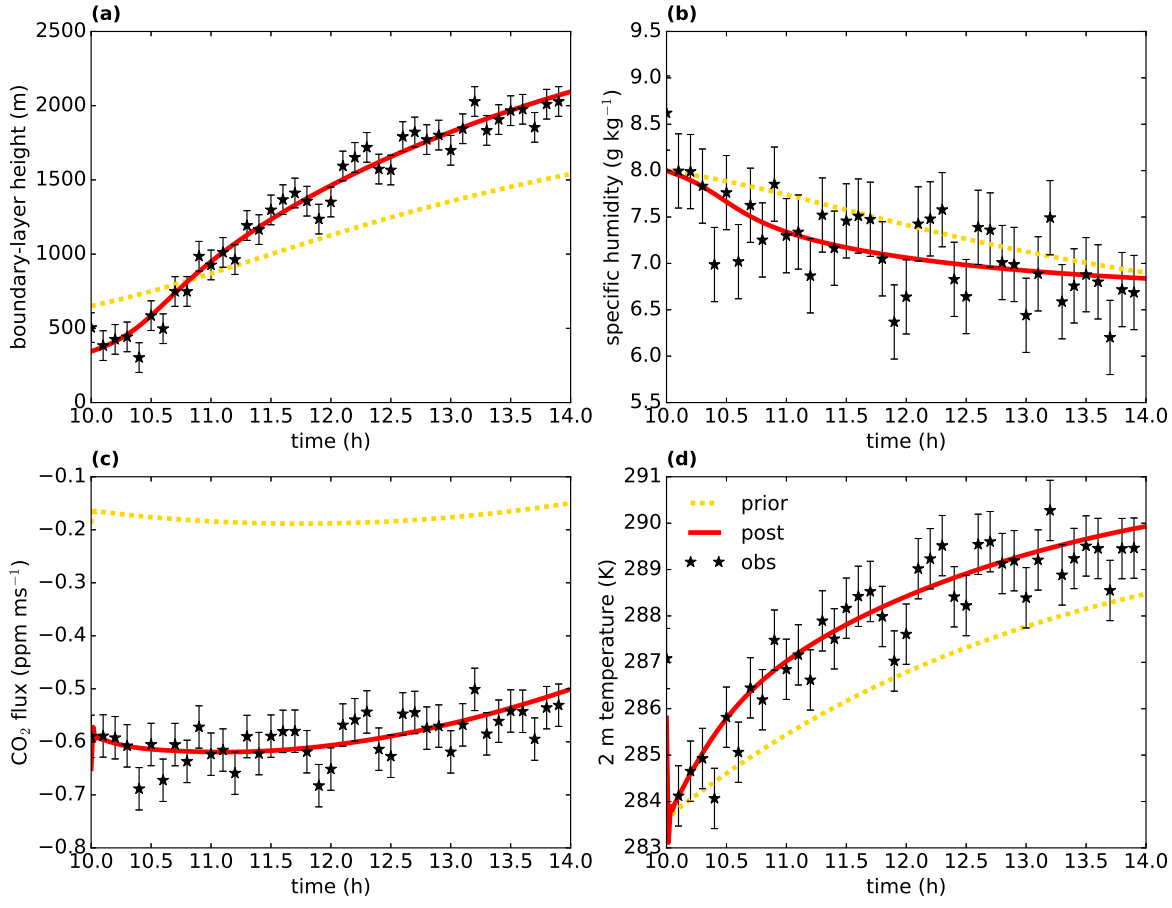


Figure 4. Observation system simulation experiment with perturbed observations. The specific humidity is from the mixed layer.

becomes closer to 1 for the vast majority of observation streams, except for three of the specific humidity streams and the sensible heat flux.

Let us now turn to the energy balance (Fig. 7). As described in Sect. 3.3, we forced the energy balance in the observations to close, by partitioning the energy balance gap partly to H, and partly to LE. As the model also has a closed energy balance, the extra energy could make it easier for the model to obtain a good fit. We notice that most of the additional energy is partitioned to LE (Table 3 and Fig. 7). The energy balance closure residual (Eq. 8) is important to account for in this case, the mean value of the residuals is only 11 % smaller as the mean of the measured (without applying Eq. 9) sensible heat flux. Note that for some of the data points around noon, the energy-balance correction tends to decrease the heat fluxes. This is likely caused by

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022

Geoscientific Model Development Discussions

Table 2. Observation fit statistics for the Cabauw optimisation. the left columns show the root mean squared error (RMSE) for the prior and posterior state. The two right columns show the ratio between model and observation variance for the prior and posterior state. Only model data points for times at which we have an observation available are taken into account. The sensible and latent heat flux observations used are corrected for the energy balance gap, see Eq. (9) and Eq. (10). The observation streams are described in Table A2.

Observation stream	RMSE prior	RMSE post.	Var. ratio prior	Var. ratio post.
T_{200} (K)	0.828	0.286	1.456	0.763
T_{140} (K)	0.868	0.296	1.449	0.756
T_{80} (K)	0.751	0.259	1.422	0.753
T_{40} (K)	0.741	0.242	1.426	0.773
T_{20} (K)	0.827	0.294	1.440	0.786
T_{10} (K)	0.781	0.282	1.434	0.789
T_2 (K)	0.815	0.307	1.505	0.842
q_{200} (kg kg^{-1})	9.22×10^{-4}	1.24×10^{-4}	0.353	0.753
q_{140} (kg kg^{-1})	9.59×10^{-4}	1.24×10^{-4}	0.332	0.708
q_{80} (kg kg^{-1})	9.49×10^{-4}	1.33×10^{-4}	0.328	0.623
q_{40} (kg kg^{-1})	8.70×10^{-4}	1.71×10^{-4}	0.351	0.488
q_{20} (kg kg^{-1})	8.87×10^{-4}	1.74×10^{-4}	0.434	0.406
q_{10} (kg kg^{-1})	9.16×10^{-4}	1.67×10^{-4}	0.523	0.383
q_2 (kg kg^{-1})	8.90×10^{-4}	2.55×10^{-4}	0.514	0.219
$CO2_{207}$ (ppm)	19.693	0.712	5.728	0.866
$CO2_{127}$ (ppm)	21.528	2.778	3.754	0.568
$CO2_{67}$ (ppm)	20.774	2.088	3.840	0.578
$CO2_{27}$ (ppm)	19.506	1.342	4.548	0.660
h (m)	270.5	131.5	0.524	1.107
LE (W m^{-2})	22.28	18.47	0.691	0.986
H (W m^{-2})	33.41	13.05	0.997	1.316
F_{CO2} (mg CO$_2$ m^{-2} s^{-1})	0.494	0.102	0.342	0.514
S_{out} (W m^{-2})	18.83	10.21	2.507	2.095

the presence of cumulus clouds, causing a fast drop in net radiation. The measured heat fluxes tend to react slower however, leading to a negative value for ε_{eb} in Eq. (8).

As is the case for the surface energy balance closure problem, there can also be biases in the CO_2 flux eddy-covariance observations (Liu et al., 2006). Here we have neglected this for simplicity. Such a bias can be accounted for in the framework, by adding a scaling factor for the surface CO_2 flux observations to the state. This however implies the assumption that the bias takes the form of a fixed fraction of the observed surface CO_2 flux.

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022

Geoscientific Model Development Discussions

Table 3. State parameters for the Cabauw optimisation. δ_{nor} is the normalised deviation, i.e. the posterior minus the prior, normalised with the square root of the prior variance (Eq. 27). "Post. st. dev." is the posterior standard deviation of the state parameters.

Parameter	Prior	Posterior	δ_{nor}	Post. st. dev.
adv_θ ($\mathrm{K\,s^{-1}}$)	0.000	-3.684×10^{-5}	-6.632×10^{-2}	1.731×10^{-5}
adv_q ($\mathrm{kg\,kg^{-1}s^{-1}}$)	0.000	1.086×10^{-8}	1.955×10^{-2}	4.415×10^{-8}
adv_{CO2} ($\mathrm{ppm\,s^{-1}}$)	0.000	1.602×10^{-3}	0.3845	5.20×10^{-4}
Δ_θ (K)	4.200	3.121	-0.7191	0.486
γ_θ ($\mathrm{K\,m^{-1}}$)	3.600×10^{-3}	2.849×10^{-3}	-0.2502	6.10×10^{-4}
Δ_q ($\mathrm{kg\,kg^{-1}}$)	-8.000×10^{-4}	-2.138×10^{-3}	-0.6692	5.59×10^{-4}
γ_q ($\mathrm{kg\,kg^{-1}\,m^{-1}}$)	-1.200×10^{-6}	-1.207×10^{-7}	0.5397	1.8409×10^{-6}
Δ_{CO2} (ppm)	-44.00	-22.42	0.8632	7.00
γ_{CO2} ($\mathrm{ppm\,m^{-1}}$)	0.000	-6.389×10^{-2}	-2.130	2.040×10^{-2}
α_{sto} (−)	1.000	0.8990	-0.4038	0.0640
α_{rad} (−)	0.2500	0.2285	-0.2146	0.0014
Frac_H (−)	0.6000	0.3474	-0.8420	0.0391
w_g (−)	0.4800	0.5153	0.2355	0.0667
R_{10} ($\mathrm{mg\,CO_2\,m^{-2}\,s^{-1}}$)	0.2300	0.6393	1.364	0.0346

The optimised state is shown in Table 3. Advection of heat remains relatively close to zero after optimisation (– 0.13 ± 0.06 $\mathrm{K\,h^{-1}}$). This is slightly outside the range of 0.1–0.3 $\mathrm{K\,h^{-1}}$ found by Casso-Torralba et al. (2008), who analysed the same day (using a longer time period). The result can be considered in agreement with Bosveld et al. (2004), who concluded large scale advection to be negligible for this day. For CO_2 however, we find an advection of 5.8 ± 1.9 $\mathrm{ppm\,h^{-1}}$, which is higher than what was found by Casso-Torralba et al. (2008, their Fig. 9). We shortly return to this later in this section. The two parameters that deviate the most from their prior values (normalised with the prior standard deviation, Table 3) are γ_{CO2} and R_{10}. Both parameters are linked to the CO_2 budget, γ_{CO2} influences the amount of entrained CO_2 from the free troposphere, while R_{10} determines the amount of CO_2 entering the mixed layer via respiration. From Fig. 5c, it is clear that the prior run has a way too strong net surface CO_2 flux compared to the observations, which explains why R_{10} is strongly increased in the optimised state. Two main pathways in which the CO_2 flux can decrease (in magnitude) is either by increasing R_{10} or by decreasing the conductance (via α_{sto}). However, the latter also impacts the model fit of the latent heat flux. Unfortunately, separate measurements of GPP and respiration (e.g. derived from carbonyl sulfide observations, Whelan et al., 2018) are not available in our dataset. These could help to further constrain the CO_2-related parameters.

The discussion above illustrates that not all of the parameters in the state may be assumed to be fully independent. To analyse the posterior correlations, we have constructed a correlation matrix (Sect. 5.2), shown in Fig. 8. From the 174 perturbed ensemble members, 150 were successful, and were used for the calculation of statistics. As expected, the correlation between α_{sto} and R_{10} is very strongly positive (0.94) and this can likely be explained by their opposite effects on the CO_2 flux, as

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022

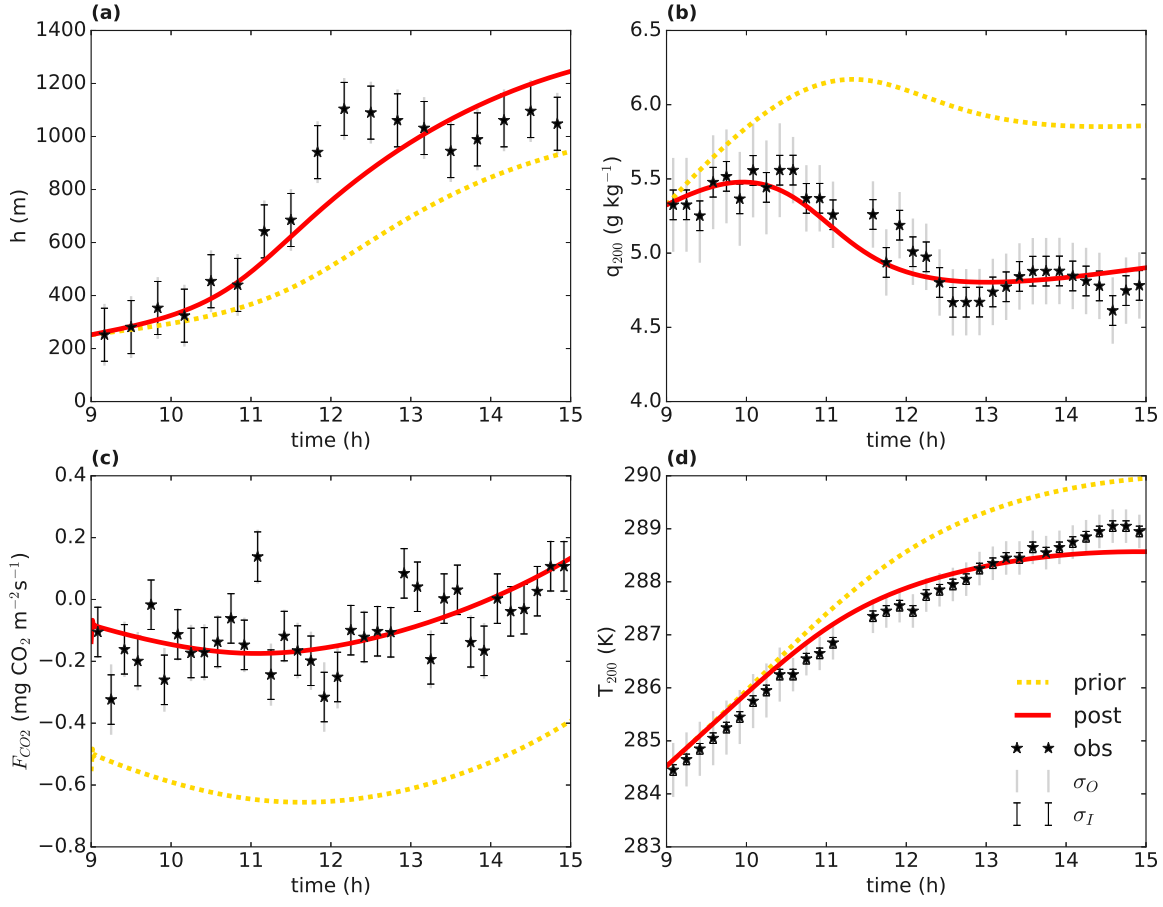


Figure 5. Optimisation for the Cabauw data. The full red line shows the optimised model ("post" meaning posterior), the dotted yellow line is the prior model. The observations are shown by a black star, the error bars show the measurement error σ_I and total error σ_O on the observations.

explained in the previous paragraph. Some of the correlations between parameters can be relatively complex, e.g. a correlation between two posterior parameters might involve a third parameter that correlates with both.

Coming back to the discrepancy in CO_2 advection between our analysis and what was found by Casso-Torralba et al. (2008), it can be noted that the adv_{CO2} parameter is relatively strongly correlated with both the Δ_{CO2} and γ_{CO2} parameters (Fig. 8). This can indicate that entrainment from the free troposphere is hard to disentangle from advection in shaping the CO_2 budget, with the current set of observations we incorporate. Differences in how entrainment is handled by Casso-Torralba et al. (2008) might explain part of the difference in estimated CO_2 advection.

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022

Geoscientific Model Development Discussions

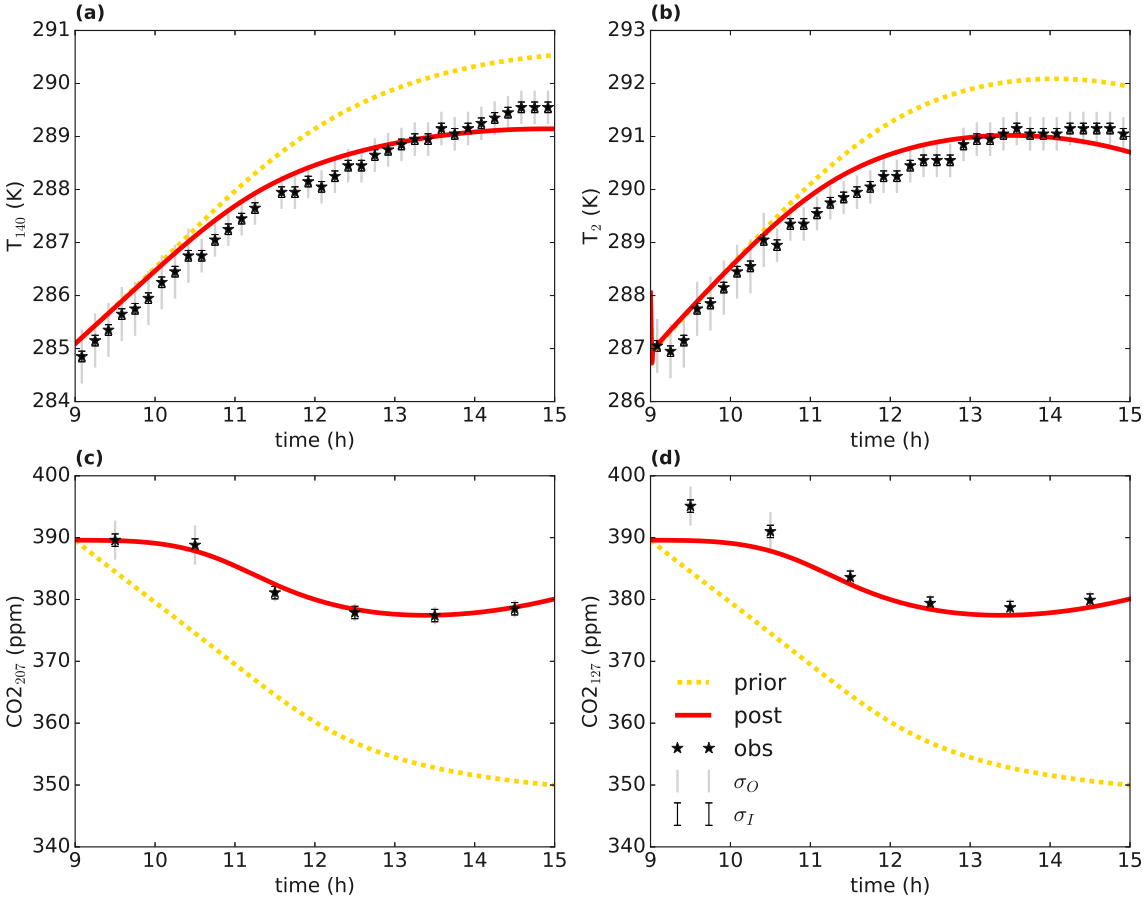


Figure 6. As in Fig. 5, but for a different subset of observations.

To check to what extent the obtained correlations are robust and independent of the selected number of ensemble members, we reconstructed the correlation matrix using only half of the successful perturbed members (75 of the 150). The average absolute value of difference between the non-diagonal matrix entries when using the subsample and the non-diagonal matrix entries when using the full successful perturbed ensemble amounts to 0.05, with a maximum of 0.23 for one entry. This suggests that using 150 members in the correlation analysis leads to a reasonably robust estimate of the posterior correlations.

Another option we explore is to analyse the posterior probability density functions (pdfs) for the successful perturbed members in the ensemble. As an example, we show the pdfs for two parameters, adv_θ and γ_q (Fig. 9). For adv_θ, the posterior uncertainty is clearly reduced compared to the prior, as the posterior pdf is markedly narrower. For γ_q (the free tropospheric specific humidity lapse rate) however, there is no clear reduction in uncertainty. The wide posterior pdf implies that similar

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022

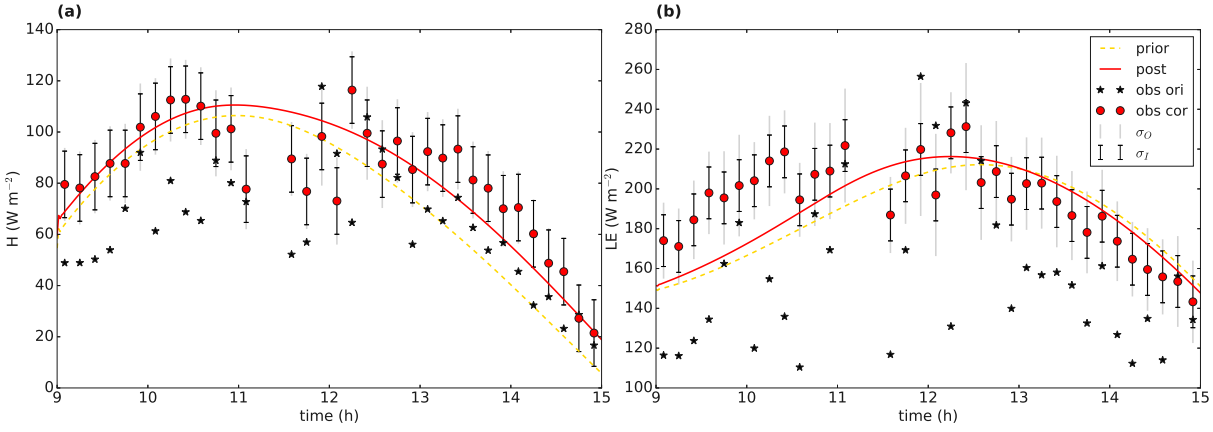


Figure 7. Energy balance data (left: sensible heat flux, right: latent heat flux) for the Cabauw case. The black stars are the original observations, the red dots the observations after forced closure of the energy balance. The error bars indicate the measurement error σ_I and the total observational error σ_O. The full red line shows the optimised model ("post" meaning posterior), the dashed yellow line is the prior model.

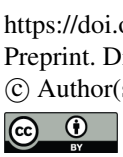
results can be obtained over a relatively wide range of γ_q, possibly by perturbing other parameters with a similar effect. To further constrain this parameter, more specific observation streams would need to be added, possibly from radiosondes. The use of ICLASS in e.g. the planning of observational campaigns can therefore help to determine beforehand what type of observations are needed to better constrain the processes represented in the model.

To give an idea of the computational costs involved, we added a timer to one optimisation. An optimisation like the one described in this section, but without the use of an ensemble, took in our specific case about 1000 times the time of an individual model simulation using the original CLASS model. However, in our test an individual simulation with this model took less than 1 second using a single CPU, so the computational cost is still relatively small. This is an advantage of using a relatively simple model. The total time it takes to perform an optimisation is dependent on how well the optimisation converges and on the configuration. Using an ensemble increases the computational costs, but multiple ensemble members can be run in parallel on multiprocessor computing systems.

11 Concluding discussion

We have presented here a description of ICLASS, a variational Inverse modelling framework for the Chemistry Land-surface Atmosphere Soil Slab model. This framework serves as a tool to study the atmospheric boundary layer and/or land–atmosphere exchange. It avoids the need of manual trial-and-error in setting parameter values, thereby providing more objectivity. Some extent of subjectivity however remains, as the proper specification of errors is not always simple (e.g. Rödenbeck et al., 2003). The use of more advanced error estimation methods can mitigate this. The very non-linear model around which the framework is built, makes the optimisation challenging. In this study, the two main ways in which this challenge is tackled are:

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022



Geoscientific Model Development Discussions

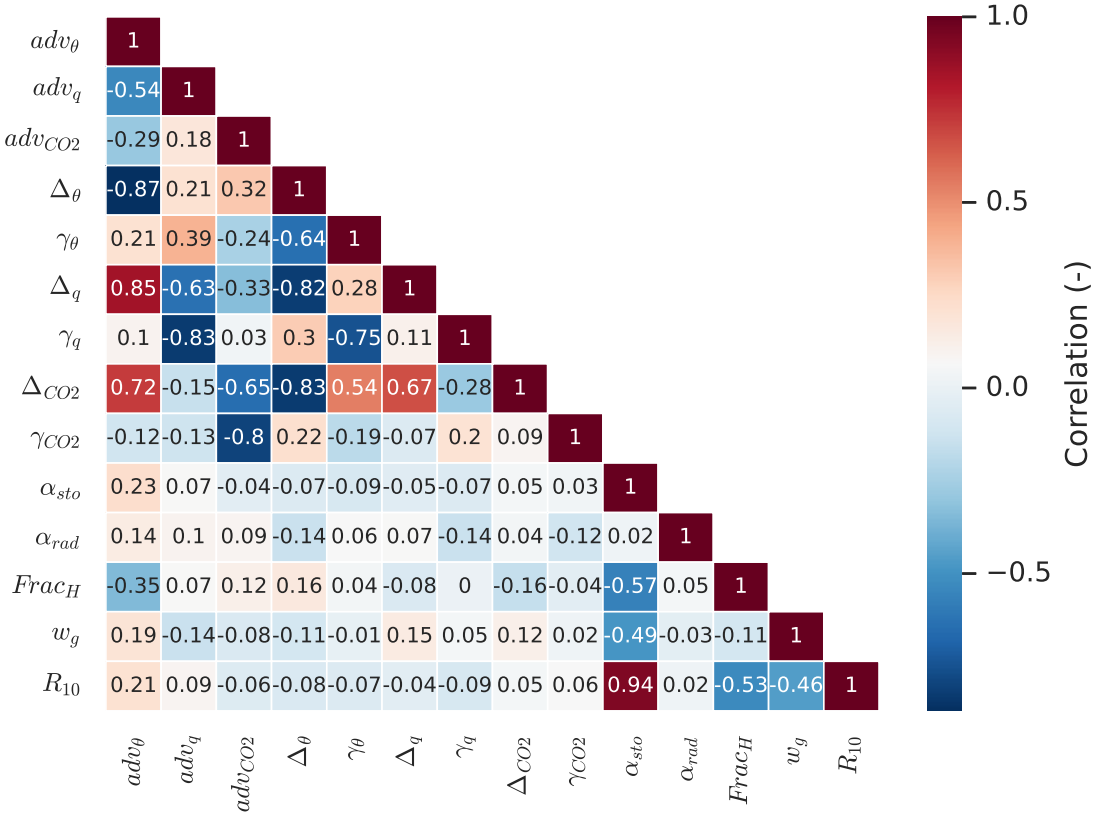


Figure 8. Correlation matrix for the optimised state parameters for the Cabauw case.

- The use of an analytical gradient of the cost function, involving the model adjoint, allowing for more precise gradient calculations
- The possibility of running ICLASS in a Monte-Carlo way. This involves perturbing the prior state vector and the (scaled) observations. When a single optimisation does not converge, the use of an ensemble can provide a solution.

The latter way of running ICLASS also has the advantage that posterior error statistics can be obtained, which is of paramount importance in inverse modelling.

The model is relatively simple, yet contains the physics to model the essentials of the convective boundary layer and of land surface–atmosphere exchange. Its simplicity however means that the model does not perform well in every situation, the best

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022

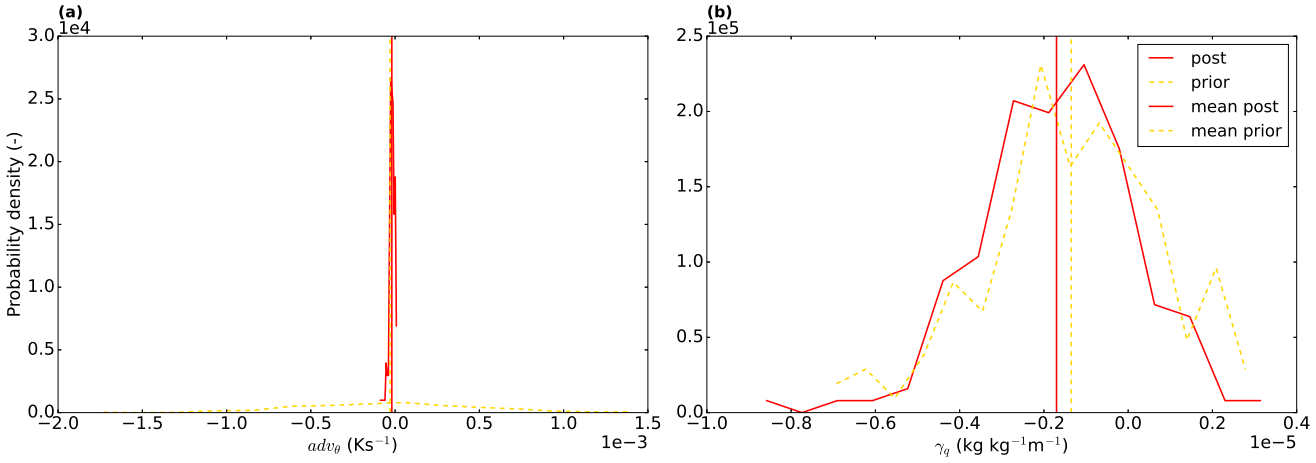


Figure 9. Probabilty density functions for the adv_θ (left) and γ_q (right) parameters, the red full line is for the posterior and the yellow dashed line for the prior distribution. The vertical lines represent the mean of the distributions. 15 bins were used. Note that the prior distribution is determined from the sample of priors, which has a component of randomness.

performance can be expected for days when the boundary layer resembles a prototypical convective boundary layer. We have shown the usefulness of ICLASS by applying it to a "golden day" at a Dutch grassland site where extensive data is available. The fit to several streams of observations simultaneously was greatly improved in the posterior compared to the prior. We have to keep in mind, however, that we cannot expect a relatively simple model to capture all small-scale processes playing a role in the atmospheric boundary layer and in land surface–atmosphere exchange.

Key strengths of this framework are that observations from several information streams can be used simultaneously, and surface layer profiles can be taken into account. The framework allows to integrate knowledge from ecosystem-level studies (fluxes) and global studies (mixing ratios). It can be seen as a tool that maximizes the use of available observational data at ABL/ecosystem level (e.g. along tall towers like the Cabauw tower). Another feature of the framework is the capacity of correcting observations for biases. A specific bias-correction system for the energy balance closure problem is implemented. The energy balance residual was shown to be substantial at the Dutch grassland site in our application example. Correcting for biases is critical in inverse modelling, to prevent bias errors to propagate in parameter estimates.

ICLASS is computationally relatively cheap to run and can be extended in the future by e.g. incorporating a more detailed representation of the vegetation. This extension can further improve the capabilities to fit sets of observations at locations with a more complex vegetation structure.

Code and data availability. The code is hosted at GitHub, the current version of ICLASS is available from https://github.com/PBosmanatm/ICLASS under a GNU General Public License, v3.0 or later. The adapted forward model is also included via the GitHub link, as well as the ICLASS manual. The GitHub repository is linked to Zenodo, which provides DOIs for released software. The release on which this reference pa-

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022


Geoscientific Model Development Discussions

per is based can be found at https://doi.org/10.5281/zenodo.6408051 (Bosman and Krol, 2022). The code and the data used for creating the plots with optimisation results in this paper are part of the downloadable material. The data used in this paper can somewhat differ from the most recent version of these data. Boundary layer height data were provided by Henk Klein Baltink (KNMI). This data (and all other used data) can be found via the Zenodo link. The newest version of the Cabauw data (except ABL heights) can be found at the following locations: The CO_2 mixing ratios can be found at the ICOS (https://www.icos-cp.eu/data-products/ERE9-9D85) and ObsPack (https://gml.noaa.gov/ccgg/obspack/) websites. Temperature, heat fluxes etc. can be found at https://dataplatform.knmi.nl/dataset/?tags=Insitu&tags=CESAR.

Appendix A: Description of used parameters and observation streams

Table A1. Used parameters in this paper.

Name	Name in code	Description	Units
adv_{CO2}	advCO2	Advection of CO_2	$\mathrm{ppm\,s^{-1}}$
adv_q	advq	Advection of moisture	$\mathrm{kg\,kg^{-1}s^{-1}}$
adv_θ	advtheta	Advection of heat	$\mathrm{K\,s^{-1}}$
α_{sto}	sca_sto	Scaling factor for stomatal conductance	–
α_{rad}	alpha	Surface albedo	–
Δ_{CO2}	deltaCO2	Initial CO_2 jump at h	ppm
Δ_q	deltaq	Initial specific humidity jump at h	$\mathrm{kg\,kg^{-1}}$
Δ_θ	deltatheta	Initial temperature jump at h	K
Frac_H	FracH	Fraction of energy balance gap partitioned to H obs	–
γ_{CO2}	gammaCO2	Free atmosphere CO_2 lapse rate	$\mathrm{ppm\,m^{-1}}$
γ_q	gammaq	Free atmosphere specific humidity lapse rate	$\mathrm{kg\,kg^{-1}\,m^{-1}}$
γ_θ	gammatheta	Free atmosphere potential temperature lapse rate	$\mathrm{K\,m^{-1}}$
h	h	Initial atmospheric boundary-layer height	m
R_{10}	R10	Respiration at 10 °C	$\mathrm{mg\,CO_2\,m^{-2}\,s^{-1}}$
w_g	wg	Volumetric water content top soil layer	–
CO_2	CO2	Mixed-layer CO_2	ppm
z_{0m}	z0m	Roughness length for momentum	m
z_{0h}	z0h	Roughness length for scalars	m

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022

Geoscientific Model Development Discussions

Table A2. Used observation streams in this paper. Note that variables such as T_{200} and T_{140} are called "Tmh" and "Tmh2" respectively in the model code for Sect. 10, which represents temperatures at the user-specified temperature measuring heights 1 and 2 respectively. For the CO_2 mixing ratio, we do not make a distinction in this paper between moist-air and dry-air mixing ratio.

Name	Name in code	Description	Units
T_{200}	Tmh (Sect. 10)	Temperature measured at 200 m height	K
T_{140}	Tmh2 (Sect. 10)	Temperature measured at 140 m height	K
	⋮		
T_2	Tmh7 (Sect. 10), Tmh (Sect. 9)	Temperature measured at 2 m height	K
q_{200}	qmh (Sect. 10)	Specific humidity measured at 200 m height	$\mathrm{kg\,kg^{-1}}$
q_{140}	qmh2 (Sect. 10)	Specific humidity measured at 140 m height	$\mathrm{kg\,kg^{-1}}$
	⋮		
q_2	qmh7 (Sect. 10)	Specific humidity measured at 2 m height	$\mathrm{kg\,kg^{-1}}$
q	q	Mixed-layer specific humidity	$\mathrm{kg\,kg^{-1}}$
$CO2_{207}$	CO2mh (Sect. 10)	CO_2 mixing ratio measured at 207 m height	ppm
	⋮		
$CO2_{27}$	CO2mh4 (Sect. 10)	CO_2 mixing ratio measured at 27 m height	ppm
$CO2_{20}$	CO2mh (Sect. 9)	CO_2 mixing ratio measured at 20 m height	ppm
h	h	Boundary-layer height	m
H	H	Surface sensible heat flux	$\mathrm{W\,m^{-2}}$
LE	LE	Surface latent heat flux	$\mathrm{W\,m^{-2}}$
F_{CO2}	wCO2	Surface CO_2 flux	$\mathrm{mg\,CO_2\,m^{-2}\,s^{-1}}$
S_{out}	Swout	Outgoing shortwave radiation	$\mathrm{W\,m^{-2}}$

Author contributions. MCK and PJMB designed the study. PJMK performed the majority of the coding and adjoint model construction, with help from MCK. PJMK performed the numerical optimisations, and wrote the manuscript with help from MCK.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. This work is part of the COS-OCS project (http://cos-ocs.eu/), a project that received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No 742798.

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022

Part of this work was carried out on the Dutch national e-infrastructure with the support of SURF Cooperative. We are grateful to Jordi Vilà-Guerau De Arellano, Fred Bosveld and Arnoud Frumau for helping us with and providing the Cabauw data. We also like to thank the developers of the CLASS model. We furthermore thank Chiel van Heerwaarden and Jordi Vilà-Guerau De Arellano for providing comments on the manuscript.

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022


Geoscientific Model Development Discussions

References

Barbaro, E., Vilà-Guerau de Arellano, J., Ouwersloot, H. G., Schröter, J. S., Donovan, D. P., and Krol, M. C.: Aerosols in the convective boundary layer: Shortwave radiation effects on the coupled land-atmosphere system, Journal of Geophysical Research: Atmospheres, 119, 5845–5863, https://doi.org/10.1002/2013JD021237, 2014.

Bergamaschi, P., Frankenberg, C., Meirink, J. F., Krol, M., Villani, M. G., Houweling, S., Dentener, F., Dlugokencky, E. J., Miller, J. B., Gatti, L. V., Engel, A., and Levin, I.: Inverse modeling of global and regional CH4 emissions using SCIAMACHY satellite retrievals, Journal of Geophysical Research Atmospheres, 114, 1–28, https://doi.org/10.1029/2009JD012287, 2009.

Bosman, P. and Krol, M.: PBosmanatm/ICLASS: ICLASS v1.0, Zenodo, https://doi.org/10.5281/zenodo.6408051, 2022.

Bosveld, F., Van Meijgaard, E., Moors, E., and Werner, C.: Interpretation of flux observations along the Cabauw 200 m meteorological tower, in: 16th Symposium on Boundary Layers and Turbulence 6.18, pp. 1–4, Portland, USA, 2004.

Bosveld, F. C., Baas, P., Beljaars, A. C. M., Holtslag, A. A. M., de Arellano, J. V.-G., and van de Wiel, B. J. H.: Fifty Years of Atmospheric Boundary-Layer Research at Cabauw Serving Weather, Air Quality and Climate, Boundary-Layer Meteorology, 177, 583–612, https://doi.org/10.1007/s10546-020-00541-w, 2020.

Brasseur, G. and Jacob, D.: Inverse Modeling for Atmospheric Chemistry, in: Modeling of Atmospheric Chemistry, pp. 487–537, Cambridge University Press, Cambridge, https://doi.org/10.1017/9781316544754.012, 2017.

Casso-Torralba, P., de Arellano, J. V. G., Bosveld, F., Soler, M. R., Vermeulen, A., Werner, C., and Moors, E.: Diurnal and vertical variability of the sensible heat and carbon dioxide budgets in the atmospheric surface layer, Journal of Geophysical Research Atmospheres, 113, https://doi.org/10.1029/2007JD009583, 2008.

Charpentier, I.: Checkpointing Schemes for Adjoint Codes: Application to the Meteorological Model Meso-NH, SIAM Journal on Scientific Computing, 22, 2135–2151, https://doi.org/10.1137/S1064827598343735, 2001.

Chevallier, F., Fisher, M., Peylin, P., Serrar, S., Bousquet, P., Bréon, F.-M., Chédin, A., and Ciais, P.: Inferring CO2 sources and sinks from satellite observations: Method and application to TOVS data, Journal of Geophysical Research, 110, D24 309, https://doi.org/10.1029/2005JD006390, 2005.

Chevallier, F., Bréon, F.-M., and Rayner, P. J.: Contribution of the Orbiting Carbon Observatory to the estimation of CO2 sources and sinks: Theoretical study in a variational data assimilation framework, Journal of Geophysical Research, 112, D09 307, https://doi.org/10.1029/2006JD007375, 2007.

Claerbout, J. F.: Earth soundings analysis: processing versus inversion, Blackwell Scientific Publications, Cambridge, 1992.

Commane, R., Meredith, L. K., Baker, I. T., Berry, J. A., Munger, J. W., Montzka, S. A., Templer, P. H., Juice, S. M., Zahniser, M. S., and Wofsy, S. C.: Seasonal fluxes of carbonyl sulfide in a midlatitude forest, Proceedings of the National Academy of Sciences of the United States of America, 112, 14 162–14 167, https://www.jstor.org/stable/26466420, 2015.

Elizondo, D., Faure, C., and Cappelaere, B.: Automatic- versus Manual- differentiation for non-linear inverse modeling, Tech. rep., INRIA (Institut National de Recherche en Informatique et en Automatique), https://hal.inria.fr/inria-00072666/document, 2000.

Errico, R. M.: What Is an Adjoint Model?, Bulletin of the American Meteorological Society, 78, 2577–2591, https://doi.org/10.1175/1520-0477(1997)078<2577:WIAAM>2.0.CO;2, 1997.

Foken, T.: The Energy Balance Closure Problem : An Overview, Ecological Applications, 18, 1351–1367, http://www.jstor.org/stable/40062260, 2008.

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022

Giering, R. and Kaminski, T.: Recipes for adjoint code construction, ACM Transactions on Mathematical Software, 24, 437–474, https://doi.org/10.1145/293686.293695, 1998.

Henze, D. K., Hakami, A., and Seinfeld, J. H.: Atmospheric Chemistry and Physics Development of the adjoint of GEOS-Chem, Atmos. Chem. Phys, 7, 2413–2433, www.atmos-chem-phys.net/7/2413/2007/, 2007.

Honnorat, M., Marin, J., Monnier, J., and Lai, X.: Dassflow v1.0: a variational data assimilation software for 2D river flows, Tech. rep., INRIA (Institut National de Recherche en Informatique et en Automatique), http://hal.inria.fr/inria-00137447, 2007.

Hunter, J.: Matplotlib: A 2D Graphics Environment, Computing in Science & Engineering, 9, 90–95, https://doi.org/10.1109/MCSE.2007.55, 2007.

Jacobs, C.: Direct impact of atmospheric CO2 enrichment on regional transpiration, Ph.D. thesis, Wageningen University, 1994.

Jarvis, P.: The interpretation of the variations in leaf water potential and stomatal conductance found in canopies in the field, Philosophical Transactions of the Royal Society of London. B, Biological Sciences, 273, 593–610, https://doi.org/10.1098/rstb.1976.0035, 1976.

Krol, M. C., Meirink, J. F., Bergamaschi, P., Mak, J. E., Lowe, D., Jöckel, P., Houweling, S., and Röckmann, T.: What can 14CO measurements tell us about OH?, Atmospheric Chemistry and Physics, 8, 5033–5044, https://doi.org/10.5194/acp-8-5033-2008, 2008.

Liu, H., Randerson, J. T., Lindfors, J., Massman, W. J., and Foken, T.: Consequences of incomplete surface energy balance closure for CO2 fluxes from open-path CO2/H2O infrared gas analysers, Boundary-Layer Meteorology, 120, 65–85, https://doi.org/10.1007/s10546-005-9047-z, 2006.

Ma, J., Kooijmans, L. M., Cho, A., Montzka, S. A., Glatthor, N., Worden, J. R., Kuai, L., Atlas, E. L., and Krol, M. C.: Inverse modelling of carbonyl sulfide: Implementation, evaluation and implications for the global budget, Atmospheric Chemistry and Physics, 21, 3507–3529, https://doi.org/10.5194/acp-21-3507-2021, 2021.

Margulis, S. A. and Entekhabi, D.: A Coupled Land Surface–Boundary Layer Model and Its Adjoint, Journal of Hydrometeorology, 2, 274–296, https://doi.org/10.1175/1525-7541(2001)002<0274:ACLSBL>2.0.CO;2, 2001a.

Margulis, S. A. and Entekhabi, D.: Feedback between the land surface energy balance and atmospheric boundary layer diagnosed through a model and its adjoint, Journal of Hydrometeorology, 2, 599–620, https://doi.org/10.1175/1525-7541(2001)002<0599:FBTLSE>2.0.CO;2, 2001b.

McNorton, J., Wilson, C., Gloor, M., Parker, R. J., Boesch, H., Feng, W., Hossaini, R., and Chipperfield, M. P.: Attribution of recent increases in atmospheric methane through 3-D inverse modelling, Atmospheric Chemistry and Physics, 18, 18 149–18 168, https://doi.org/10.5194/acp-18-18149-2018, 2018.

Meirink, J. F., Bergamaschi, P., and Krol, M. C.: Four-dimensional variational data assimilation for inverse modelling of atmospheric methane emissions: Method and comparison with synthesis inversion, Atmospheric Chemistry and Physics, 8, 6341–6353, https://doi.org/10.5194/acp-8-6341-2008, 2008.

Miller, S. M., Michalak, A. M., and Levi, P. J.: Geoscientific Model Development Atmospheric inverse modeling with known physical bounds: an example from trace gas emissions, Geosci. Model Dev, 7, 303–315, https://doi.org/10.5194/gmd-7-303-2014, 2014.

Monin, A. and Obukhov, A.: Basic laws of turbulent mixing in the atmosphere near the ground, Tr. Akad. Nauk SSSR Geophiz. Inst., 24, 1963–1987, 1954.

Nash, S. G.: A survey of truncated-Newton methods, Journal of Computational and Applied Mathematics, 124, 45–59, https://doi.org/10.1016/S0377-0427(00)00426-X, 2000.

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022

Oncley, S. P., Foken, T., Vogt, R., Kohsiek, W., DeBruin, H. A. R., Bernhofer, C., Christen, A., van Gorsel, E., Grantz, D., Feigenwinter, C., Lehner, I., Liebethal, C., Liu, H., Mauder, M., Pitacco, A., Ribeiro, L., and Weidinger, T.: The Energy Balance Experiment EBEX-2000. Part I: overview and energy balance, Boundary-Layer Meteorology, 123, 1–28, https://doi.org/10.1007/s10546-007-9161-1, 2007.

Ouwersloot, H. G., Vilà-Guerau De Arellano, J., Nàlscher, A. C., Krol, M. C., Ganzeveld, L. N., Breitenberger, C., Mammarella, I., Williams, J., and Lelieveld, J.: Characterization of a boreal convective boundary layer and its impact on atmospheric chemistry during HUMPPA-COPEC-2010, Atmospheric Chemistry and Physics, 12, 9335–9353, https://doi.org/10.5194/acp-12-9335-2012, 2012.

Raoult, N. M., Jupp, T. E., Cox, P. M., and Luke, C. M.: Land-surface parameter optimisation using data assimilation techniques: The adJULES system V1.0, Geoscientific Model Development, 9, 2833–2852, https://doi.org/10.5194/gmd-9-2833-2016, 2016.

Renner, M., Brenner, C., Mallick, K., Wizemann, H. D., Conte, L., Trebs, I., Wei, J., Wulfmeyer, V., Schulz, K., and Kleidon, A.: Using phase lags to evaluate model biases in simulating the diurnal cycle of evapotranspiration: A case study in Luxembourg, Hydrology and Earth System Sciences, 23, 515–535, https://doi.org/10.5194/hess-23-515-2019, 2019.

Rödenbeck, C., Houweling, S., Gloor, M., and Heimann, M.: CO2 flux history 1982-2001 inferred from atmospheric data using a global inversion of atmospheric transport, Atmospheric Chemistry and Physics, 3, 1919–1964, https://doi.org/https://doi.org/10.5194/acp-3-1919-2003, 2003.

Ronda, R. . J. ., de Bruin, H. . A. . R., and Holtslag, A.: Representation of the Canopy Conductance in Modeling the Surface Energy Budget for Low Vegetation, American Meteorological Society, 40, 1431–1444, https://www.jstor.org/stable/10.2307/26184869, 2001.

Schulte, R., van Zanten, M., Rutledge-Jonker, S., Swart, D., Wichink Kruit, R., Krol, M., van Pul, W., and Vilà-Guerau de Arellano, J.: Unraveling the diurnal atmospheric ammonia budget of a prototypical convective boundary layer, Atmospheric Environment, 249, 118 153, https://doi.org/10.1016/J.ATMOSENV.2020.118153, 2021.

Stewart, J.: Modelling surface conductance of pine forest, Agricultural and Forest Meteorology, 43, 19–35, https://doi.org/10.1016/0168-1923(88)90003-2, 1988.

Stull, R. B.: An introduction to boundary layer meteorology, Kluwer Academic Publishers, Dordrecht, 1988.

Tarantola, A.: Inverse problem theory and methods for model parameter estimation, Society for Industrial and Applied Mathematics (siam), Philadelphia, USA, https://doi.org/10.1137/1.9780898717921, 2005.

The SciPy community: scipy.optimize.fmin_tnc, https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.fmin_tnc.html.

van der Walt, S., Colbert, S., and Varoquaux, G.: The NumPy Array: A Structure for Efficient Numerical Computation, Computing in Science & Engineering, 13, 22–30, https://doi.org/10.1109/MCSE.2011.37, 2011.

van Heerwaarden, C. C., Vilà-Guerau de Arellano, J., Moene, A. F., and Holtslag, A. A. M.: Interactions between dry-air entrainment, surface evaporation and convective boundary-layer development, Quarterly Journal of the Royal Meteorological Society, 135, 1277–1291, https://doi.org/10.1002/qj.431, 2009.

van Heerwaarden, C. C., Vilà-Guerau de Arellano, J., Gounou, A., Guichard, F., and Couvreux, F.: Understanding the daily cycle of evapotranspiration: A method to quantify the influence of forcings and feedbacks, Journal of Hydrometeorology, 11, 1405–1422, https://doi.org/10.1175/2010JHM1272.1, 2010.

Vermeulen, A. T., Hensen, A., Popa, M. E., van den Bulk, W. C. M., and Jongejan, P. A. C.: Greenhouse gas observations from Cabauw Tall Tower (1992–2010), Atmospheric Measurement Techniques, 4, 617–644, https://doi.org/10.5194/amt-4-617-2011, 2011.

Vesala, T., Suni, T., Rannik, Ü., Keronen, P., Markkanen, T., Sevanto, S., Grönholm, T., Smolander, S., Kulmala, M., Ilvesniemi, H., Ojansuu, R., Uotila, A., Levula, J., Mäkelä, A., Pumpanen, J., Kolari, P., Kulmala, L., Altimir, N., Berninger, F., Nikinmaa, E., and Hari, P.: Effect of thinning on surface fluxes in a boreal forest, Global Biogeochem. Cycles, https://doi.org/10.1029/2004GB002316, 2005.

https://doi.org/10.5194/gmd-2022-63
Preprint. Discussion started: 7 April 2022

Vilà-Guerau De Arellano, J., Van Heerwaarden, C. C., and Lelieveld, J.: Modelled suppression of boundary-layer clouds by plants in a CO2 -rich atmosphere, Nature Geoscience, 5, 701–704, https://doi.org/10.1038/ngeo1554, 2012.

Vilà-Guerau De Arellano, J., Van Heerwaarden, C. C., Van Stratum, B. J., and Van Den Dries, K.: Atmospheric boundary layer: Integrating air chemistry and land interactions, Cambridge University Press, 2015.

Whelan, M. E., Lennartz, S. T., Gimeno, T. E., Wehr, R., Wohlfahrt, G., Wang, Y., Kooijmans, L. M., Hilton, T. W., Belviso, S., Peylin, P., Commane, R., Sun, W., Chen, H., Kuai, L., Mammarella, I., Maseyk, K., Berkelhammer, M., Li, K. F., Yakir, D., Zumkehr, A., Katayama, Y., Oge, J., Spielmann, F. M., Kitz, F., Rastogi, B., Kesselmeier, J., Marshall, J., Erkkila, K. M., Wingate, L., Meredith, L. K., He, W., Bunk, R., Launois, T., Vesala, T., Schmidt, J. A., Fichot, C. G., Seibt, U., Saleska, S., Saltzman, E. S., Montzka, S. A., Berry, J. A., and Elliott Campbell, J.: Reviews and syntheses: Carbonyl sulfide as a multi-scale tracer for carbon and water cycles, Biogeosciences, 15, 3625–3657, https://doi.org/10.5194/bg-15-3625-2018, 2018.

Wouters, H., Petrova, I. Y., van Heerwaarden, C. C., Vilà-Guerau de Arellano, J., Teuling, A. J., Meulenberg, V., Santanello, J. A., and Miralles, D. G.: Atmospheric boundary layer dynamics from balloon soundings worldwide: CLASS4GL v1.0, Geoscientific Model Development, 12, 2139–2153, https://doi.org/10.5194/gmd-12-2139-2019, 2019.