



The *ddeg* Python library for point source quantification from remote sensing images (version 1.0)

Gerrit Kuhlmann¹, Erik Koene¹, Sandro Meier^{1,2}, Diego Santaren³, Grégoire Broquet³, Frédéric Chevallier³, Janne Hakkarainen⁴, Janne Nurmela⁴, Laia Amorós⁴, Johanna Tamminen⁴, and Dominik Brunner¹

¹Swiss Federal Laboratories for Materials Science and Technology (Empa), Dübendorf, Switzerland

²Remote Sensing Laboratories, University of Zurich, Zurich, Switzerland

³Laboratoire des Sciences du Climat et de l'Environnement, LSCE/IPSL, CEA-CNRS-UVSQ, Université Paris-Saclay, Gif-sur-Yvette, France

⁴Earth Observation Research, Finnish Meteorological Institute (FMI), Helsinki, Finland

Correspondence: Gerrit Kuhlmann (gerrit.kuhlmann@empa.ch)

Received: 6 December 2023 – Discussion started: 18 January 2024

Revised: 15 April 2024 – Accepted: 16 April 2024 – Published: 18 June 2024

Abstract. Atmospheric emissions from anthropogenic hotspots, i.e., cities, power plants and industrial facilities, can be determined from remote sensing images obtained from airborne and space-based imaging spectrometers. In this paper, we present a Python library for data-driven emission quantification (*ddeg*) that implements various computationally light methods such as the Gaussian plume inversion, cross-sectional flux method, integrated mass enhancement method and divergence method. The library provides a shared interface for data input and output and tools for pre- and post-processing of data. The shared interface makes it possible to easily compare and benchmark the different methods. The paper describes the theoretical basis of the different emission quantification methods and their implementation in the *ddeg* library. The application of the methods is demonstrated using Jupyter notebooks included in the library, for example, for NO₂ images from the Sentinel-5P/TROPOMI satellite and for synthetic CO₂ and NO₂ images from the Copernicus CO₂ Monitoring (CO2M) satellite constellation. The library can be easily extended for new datasets and methods, providing a powerful community tool for users and developers interested in emission monitoring using remote sensing images.

1 Introduction

The majority of anthropogenic emissions of air pollutants and greenhouse gases is confined to localized sources such as cities, power plants and industrial facilities (e.g., Crippa et al., 2022). The emissions of these hotspots can be determined from the atmospheric plumes of trace gas column densities in remote sensing images. Trace gases of interest are nitrogen dioxide (NO_x = NO₂ + NO), carbon monoxide (CO), carbon dioxide (CO₂), methane (CH₄) and others. Numerous methods have been developed in recent years to quantify the emissions from remote sensing images by matching observations to simulated plumes (e.g., Bovensmann et al., 2010; Nassar et al., 2017; Broquet et al., 2018; Ye et al., 2020; Lei et al., 2021; Kaminski et al., 2022), applying the principle of mass conservation to individual or temporally averaged images (e.g., Beirle et al., 2011; de Foy et al., 2015; Varon et al., 2018; Reuter et al., 2019; Zheng et al., 2020; Kuhlmann et al., 2021; Leguijt et al., 2023), or using machine-learning models trained with synthetic observations (e.g., Jongaramrungruang et al., 2022; Joyce et al., 2023; Dumont Le Brazidec et al., 2024).

Remotely sensed trace gas images are available from space-based imaging spectrometers such as the TROPOMI (TROPOMI; Veefkind et al., 2012) and the Orbiting Carbon Observatory-3 (OCO-3; Eldering et al., 2019) and from high-resolution point source imagers such as GHGSat, several multispectral land imag-

ing sensors (e.g., Sentinel-2) and the upcoming generation of missions in polar and geostationary orbits (i.e., Copernicus CO₂ Monitoring, CO2M; Geostationary Environment Monitoring Spectrometer, GEMS; The Global Observing SATellite for Greenhouse gases and Water cycle, GOSAT-GW; Sentinel-4 and Sentinel-5; Tropospheric Emissions: Monitoring of Pollution, TEMPO; and others). In addition, airborne imaging spectrometers can be used to map emission plumes at a high spatial resolution (e.g., Thorpe et al., 2017; Tack et al., 2019; Fujinawa et al., 2021). An example of the plumes that can be visible from a city or power plant is shown in Fig. 1. This example is from the synthetic SMARTCARB dataset (Kuhlmann et al., 2019) generated for the CO2M satellite constellation planned for launch in 2026 (ESA Earth and Mission Science Division, 2020).

Measurement-based emissions monitoring systems are currently being developed to support global efforts to reduce greenhouse gas emissions to achieve the goals of the Paris Agreement on climate change. One such system is the European CO₂ Monitoring and Verification Support (CO2MVS) capacity that will be implemented as part of the European Copernicus program (Janssens-Maenhout et al., 2020). A prototype system of the European CO2MVS capacity is currently developed as part of the CoCO2 project (<https://coco2-project.eu/>, last access: 24 May 2024). Since one goal of CO2MVS is the global monitoring of emission hotspots, several emission quantification methods were implemented and benchmarked in the CoCO2 project using synthetic CO2M CO₂ and NO₂ observations and Sentinel-5P/TROPOMI NO₂ observations (Hakkarainen et al., 2024; Santaren et al., 2024). As the new generation of satellites will provide a large number of hotspot images (e.g., Kuhlmann et al., 2021; Wang et al., 2020), it is foreseen that computationally lightweight methods will be needed to process such a large amount of data in the operational CO2MVS system. Such methods will have to make optimal use of the information contained in the images without requiring expensive plume simulations with a high-resolution atmospheric transport model.

This paper describes the Python library for data-driven emission quantification (*ddeg*; version 1.0), developed as part of the CoCO2 project as a shared library for the implementation of various lightweight approaches. Although the various methods differ in many aspects, they share many pre- and post-processing steps including data input and output. The common interface thus makes *ddeg* a powerful tool not only for comparing and benchmarking different methods, but also for implementing new approaches. The *ddeg* library was originally developed as part of the SMARTCARB project for detecting and quantifying CO₂ and NO_x emissions in synthetic CO₂ and NO₂ images of the CO2M mission (Kuhlmann et al., 2019, 2020a, 2021), but it has also been used for quantifying NO_x emissions with the airborne APEX imaging spectrometer (Kuhlmann et al., 2022). *ddeg* is designed for the lightweight emission quantification of hotspots

from remotely sensed images. A similar community-driven library exists for regional and global emission estimation with atmospheric model runs through the Community Inversion Framework (CIF; Berchet et al., 2021). The *ddeg* version presented here does not include machine-learning models, which were also considered in the CoCO2 project (Dumont Le Brazidec et al., 2023, 2024). *ddeg* has been used in the CoCO2 project for benchmarking the different methods using synthetic CO2M observations (Santaren et al., 2024) and TROPOMI NO₂ observations (Hakkarainen et al., 2024).

The paper has two parts. In the first part, we describe the general principles of lightweight emission quantification and describe the different methods. In the second part, we describe the common framework and interfaces provided by *ddeg* and details of the implementation of the different methods. The application of the *ddeg* library to synthetic CO2M observations for estimating CO₂ and NO_x emissions and to TROPOMI observations for estimating NO_x emissions is showcased in several Jupyter notebooks available in the Supplement.

2 Theoretical basis

Two families of lightweight approaches exist for emission quantification of hotspots. The first family quantifies emissions from “instantaneous plumes” obtained from single remote sensing images. The second family requires averaging over multiple images taken at different times before quantifying the emissions. The *ddeg* library currently implements four methods for emission estimation, partly in different flavors. The emission quantification methods are the (1) Gaussian plume inversion, (2) cross-sectional flux method, (3) integrated mass enhancement method and (4) divergence method. All methods can be applied to single images although the divergence method typically requires averaging over many images. In the following, the theoretical basis for plume detection, background estimation, effective wind speeds and the four quantification methods are summarized briefly. We also briefly cover the conversion of NO₂ to NO_x observations and the estimation of annual emissions from individual estimates.

2.1 Identification of the plume region

A critical first step required by most methods is the identification of a subregion within the image (described by a polygon) where the emission quantification method is applied (see Fig. 1). The subregion contains the plume, i.e., the pixels where trace gas columns are enhanced due to emissions from the source of interest and a fraction of the background field. This first step not only identifies the plume location, but also assigns the source location and computes a local coordinate system defining along- and across-plume distances.

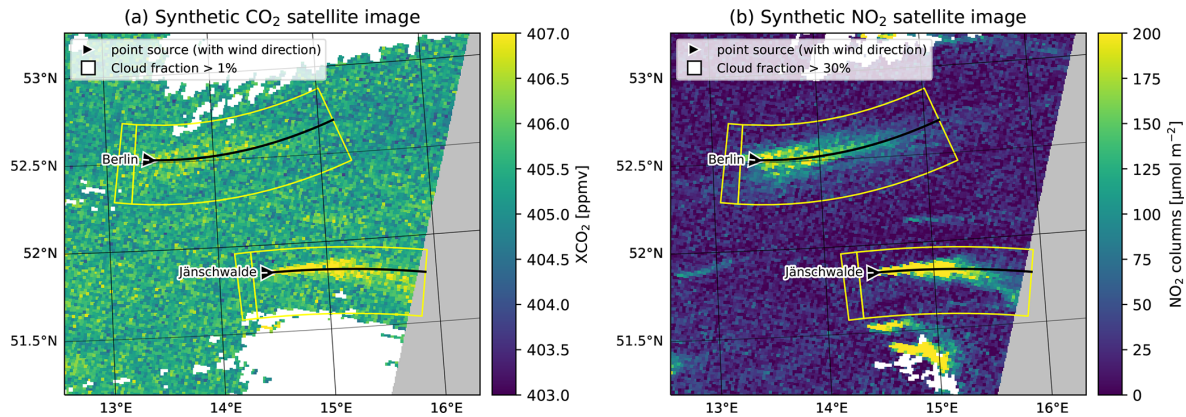


Figure 1. Example of synthetic CO₂M (a) CO₂ and (b) NO₂ satellite images from the SMARTCARB dataset. The images show the emission plumes of the city of Berlin and the coal-fired power plant near Jänschwalde at a 2 km resolution. Pixels with cloud fractions larger than 1 % for CO₂ and 30 % for NO₂ are shown in white, and regions outside the satellite swath are shown in gray. The triangular marker indicates the wind direction at the source. The yellow polygons delineate the subregions containing the plumes. The smaller polygons show the regions upstream of the source.

Broadly speaking, two different approaches to defining the subregion are available, and both have been implemented in *ddeq*. The first approach identifies the plumes inside the remote sensing images using an image segmentation algorithm. For example, if thresholding is used, pixels are assigned to the plume if their signal-to-noise ratio (SNR) exceeds a threshold, z_q :

$$\text{SNR} = \frac{V - V_{\text{bg}}}{\sigma_V} \geq z_q, \quad (1)$$

where V and V_{bg} are the total and background vertical column density and σ_V is the local noise in the image (Varon et al., 2018; Kuhlmann et al., 2019). The determination of the background, i.e., the vertical column density that would be expected without the presence of the source, is described in the next section. More complex segmentation algorithms apply feature detection using, for example, convolutional neural networks (Finch et al., 2022; Dumont Le Brazidec et al., 2023). The detected plumes can be assigned to one or more sources by checking their overlap with a known list of source locations, for example, from an emission inventory. The Boolean mask obtained from the image segmentation can be converted to the polygons shown in Fig. 1 by computing the bounding boxes in the local coordinate system.

The second approach determines the subregion based on the source location and the wind field available from, for example, a meteorological reanalysis product. This means that the remote sensing images are not used. In the simplest case, the wind vector is taken at the source location and the plume is assumed to be located downstream. It is then possible to draw a rectangular polygon with the along- and across-wind direction. The approach can be extended to simulate the plume location or an ensemble of plume locations with an atmospheric transport model, which can provide a better estimate of the plume location, especially further downstream

of the source. However, this can get computationally quite expensive and does not qualify as a lightweight method anymore.

For each detected plume, a natural coordinate system can be established with the along-plume coordinate, x , and the across-plume coordinate, y . The coordinates can be computed as distance either along and perpendicular to the wind vector. For a curved plume, the coordinates can be computed as the arc length along and distance from a two-dimensional curve fitted to the detected plume (see Fig. 1 and Kuhlmann et al., 2020a for details).

2.2 Background estimation

To estimate the emission rate of a source, we are interested in the enhancement above the background:

$$V_e(x, y) = V(x, y) - V_{\text{bg}}(x, y). \quad (2)$$

A common approach for estimating the background is applying a low-pass filter (e.g., a median filter) or a normalized convolution after masking enhancements, assuming a spatially smooth background field. Alternatively, the background can be estimated from the pixels upstream of the source. Another approach is explicitly fitting a background term in the emission quantification method, which is possible with Gaussian plume inversion and the cross-sectional flux method.

2.3 Effective wind speed

Wind speed and direction describe the transport of the trace gas in the atmosphere and are therefore important input parameters for all emission quantification methods. For methods that are applied to instantaneous images, the wind direction can be estimated from the plume direction. The

Table 1. Symbols used in equations.

Symbol	Description	Units
A	pixel area	m^2
D	decay function	1
E	emission	$\text{kg m}^{-2} \text{s}^{-1}$
F	gas flux	kg s^{-1}
G	Gaussian plume model	kg m^{-2}
H	Heaviside step function	1
J	cost function	–
K	eddy diffusion coefficient	$\text{m}^2 \text{s}^{-1}$
L	plume length	m
M	integrated mass enhancement	kg
Q	emission rate	kg s^{-1}
S	sink term	$\text{kg m}^{-2} \text{s}^{-1}$
V	vertical column density	kg m^{-2}
V_{bg}	background vertical column density	kg m^{-2}
b	offset of linear background	kg m^{-3}
c	decay correction factor	1
f	NO_2 -to- NO_x conversion factor	1
g	Gaussian curve	kg m^{-2}
m	slope of linear background	kg m^{-2}
p	two-dimensional Gaussian surface	$\text{kg m}^{-2} \text{s}^{-1}$
q	line density	kg m^{-1}
r	correlation	1
t	residence time	s
u	wind speed	m s^{-1}
x	along-plume coordinate	m
y	across-plume coordinate	m
z_q	detection threshold	1
κ	coefficient in Gaussian plume model	1
ρ	trace gas concentration	kg m^{-3}
μ	center shift of Gaussian curve	m
σ	standard width of Gaussian curve	m
σ_V	random noise of V	–
τ	decay time	s
\mathcal{P}	plume pixels	–

wind speed needs to be taken from another source such as ECMWF's ERA5 reanalysis product (Hersbach et al., 2023).

To obtain an unbiased estimate of the emissions, it is necessary to calculate the effective wind speed that corresponds to the mean transport speed of the plume. The effective wind speed is the vertically averaged wind speed weighted by the vertical profile of the trace gas concentration. It can be calculated as

$$u(x, y) = \frac{\int_0^{z_T} \rho_e(x, y, z) u(x, y, z) dz}{\int_0^{z_T} \rho_e(x, y, z) dz}, \quad (3)$$

where $\rho_e(x, y, z)$ is the concentration of the trace gas enhancement, $u(x, y, z)$ is the along-plume wind speed and z_T is a height above the plume.

Since the vertical profile, $\rho_e(z)$, is usually not known, a common approach to approximating the effective wind speed is by averaging the lowest layers of a reanalysis product (e.g.,

Fioletov et al., 2015, uses the mean of the three lowest ERA5 layers). Alternatively, plume rise calculations can be used to estimate the height and spread of the trace gas at the source location. The effective emission height can be significantly higher than the geometric height of a stack due to the momentum and buoyancy of the flue gas. Plume rise is generally influenced by stack geometry (height and diameter), flue gas properties (temperature, humidity and exit velocity) and meteorological conditions (wind speed and atmospheric stability) (Bieser et al., 2011; Brunner et al., 2019). In the case of a well-mixed atmospheric boundary layer, the exhaust plume becomes uniformly mixed throughout the depth of the boundary layer with increasing distance from the source. Therefore, a pressure-weighted mean wind speed in the boundary layer may be sufficient. This approach could also be viable for emissions from a city.

Lightweight approaches use a single value, such as the wind speed at the source location or averaged over the plume. It is important to note that the effective wind speed also has a temporal component, even in an instantaneous image, as the trace gas at the end of long plumes can be several hours old.

2.4 Emission quantification methods

Figure 2 illustrates the application of the Gaussian plume inversion, cross-sectional flux, integrated mass enhancement and divergence method. In each panel, the pseudo-color map shows the NO_2 column densities observed by an imaging spectrometer with a 5 km resolution for an emission plume of a source located at the origin. The plume was modeled with a Gaussian plume model with an emission rate of 1 kg s^{-1} ($\approx 32 \text{ kt NO}_2 \text{ a}^{-1}$), a chemical lifetime of NO_2 of 6 h, and a wind speed of 5 m s^{-1} . The white crosses mark pixels where the NO_2 column is larger than $50 \mu\text{mol m}^{-2}$. In the following, the theoretical basis for these methods is described for determining the emissions of a chemically inert gas (e.g., CO_2 and CH_4) and an exponentially decaying gas (e.g., NO_2). The conversion of NO_2 to NO_x is discussed in Sect. 2.5.

2.4.1 Gaussian plume inversion

In this method, a vertically integrated Gaussian plume model, G , is fitted to the observed column densities, V (e.g., Bovensmann et al., 2010; Nassar et al., 2017). The Gaussian plume model can be written as

$$G(x, y) = \frac{Q H(x)}{\sqrt{2\pi} u \sigma(x)} \exp\left(-\frac{y^2}{2\sigma(x)^2}\right) + V_{\text{bg}}(x, y), \quad (4)$$

with emission rate Q , wind speed u and background column $V_{\text{bg}}(x, y)$. $H(x)$ is the Heaviside step function, and x and y are the along- and across-plume coordinates. The dispersion in the across-plume direction is modeled by the standard width

$$\sigma(x) = \sqrt{\frac{2Kx^\kappa}{u}}, \quad (5)$$

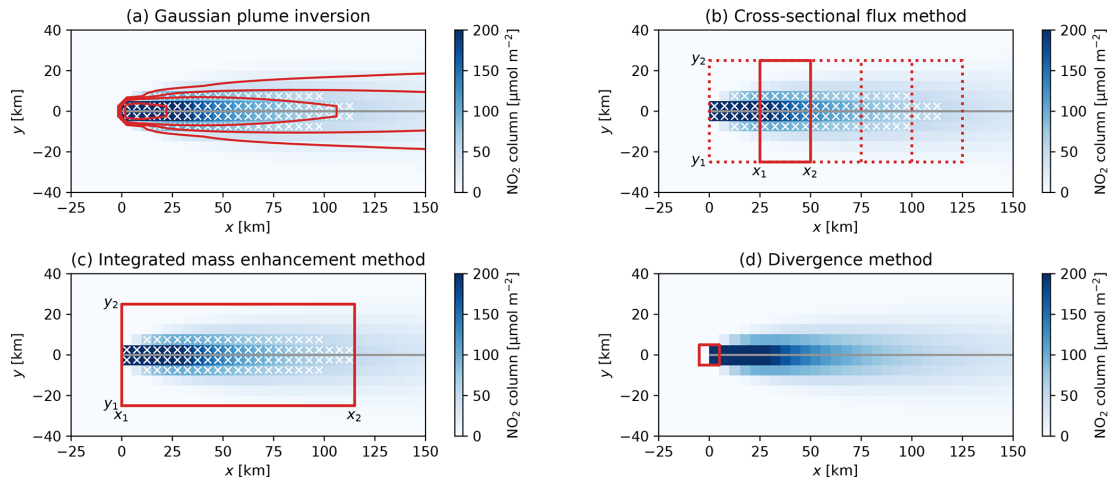


Figure 2. Sketch showing the application of the different lightweight methods to an emission plume. Each panel shows an NO₂ emission plume simulated by a Gaussian plume model ($Q = 1 \text{ kg s}^{-1}$; $u = 5 \text{ m s}^{-1}$), including exponential decay ($\tau = 6 \text{ h}$).

with the eddy diffusion coefficient, K (in $\text{m}^2 \text{ s}^{-1}$). The additional exponent κ accounts for possible changes in the dispersion rate along the plume depending on meteorological conditions. This makes it possible to modify the standard expression using $\kappa = 1$ with a power law of the form $\sigma(x) = ax^b$, which has also been used in the literature (e.g., Krings et al., 2013). An example of the Gaussian plume is shown by the contour lines in Fig. 2a.

A least-squares method can be used to obtain the optimal values for Q , K , V_{bg} and κ as well as their uncertainties by minimizing the following cost function:

$$J(Q, K, V_{\text{bg}}, \kappa) = \left\| V_{i,j} - G(x_i, y_i) \right\|_2^2, \quad (6)$$

where $V_{i,j}$ is the observed column density for the pixel with center coordinates (x_i, y_i) .

Equation (4) can be used to approximate the emission plumes for species with long lifetimes such CO₂, CO and CH₄. For species with short lifetimes, the Gaussian plume model needs to be multiplied with a decay term:

$$D(x, \tau) = H(x) \exp\left(-\frac{x}{u\tau}\right), \quad (7)$$

where the lifetime, τ , is an additional fitting parameter. In the case of NO₂, the lifetime is typically about 4 h.

The Gaussian plume model is valid for a point source, where the source area is smaller than the pixel size. For sources such as cities with dimensions larger than the pixel size, the flux slowly increases across the source area. An approach to accounting for this effect is to describe the emissions from the city by an emission map, $p(x, y)$, and the change in flux in the along-plume direction as the convolution of a map and a decay term.

$$G_a(x, y) = G(x, y) \int_{-\infty}^{+\infty} D(x', \tau) p(x - x', y) dx' \quad (8)$$

The emission map can take the form of a uniform surface within city boundaries or a two-dimensional Gaussian surface, $p(x, y)$, with the form

$$p(x, y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-r^2}} \cdot \exp\left(-\frac{(x-x_0)^2}{2\sigma_x^2(1-r^2)} - \frac{(y-y_0)^2}{2\sigma_y^2(1-r^2)} + \frac{r(x-x_0)(y-y_0)}{\sigma_x\sigma_y(1-r^2)}\right), \quad (9)$$

where (x_0, y_0) is the center position, σ_x and σ_y are the standard widths, and r is the correlation. These parameters can be included in the least-squares method as additional fitting parameters.

2.4.2 Cross-sectional flux method

The cross-sectional flux method applies mass conservation by computing the gas flux in the plume (F ; in kg s^{-1}) downwind of the source from wind speed, u , and line density, q (e.g., Varon et al., 2018; Reuter et al., 2019; Kuhlmann et al., 2021), i.e., as follows:

$$F = u \cdot q. \quad (10)$$

For a non-decaying gas, the flux is identical to the emission rate, Q , under the assumption of steady-state conditions and that turbulent mixing is negligible compared to advective transport in the along-plume direction. For a decaying gas, the flux decreases downstream of the source in the along-plume distance, x . In this case, the emission rate can be computed by compensating the flux for the along-plume decay:

$$Q = \frac{F(x)}{D(x, \tau)}. \quad (11)$$

The line density is obtained by integrating the column enhancements in the across-plume direction from y_1 to y_2 at distance x :

$$q(x) = \int_{y_1}^{y_2} (V(x, y) - V_{bg}) dy, \quad (12)$$

where the interval $[y_1, y_2]$ needs to be sufficiently large to contain the full plume extent.

The line density can simply be obtained by integrating over the enhancements for all pixels within a rectangle (a polygon in the case of a curved plume) delimited by $[x_1, x_2]$ and $[y_1, y_2]$ (see Fig. 2b). However, a disadvantage of this approach is that the background needs to be estimated first and subtracted from the observed vertical columns. Another disadvantage is that missing pixels (e.g., due to clouds) need to be interpolated to obtain the correct line density. An often-used alternative approach is therefore fitting a Gaussian curve to all observations within the rectangle:

$$g(y) = \frac{q}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right) + my + b, \quad (13)$$

with standard width σ and center position μ . The background is approximated here by a linear function with slope m and intercept b . The Gaussian curve has the advantage that it automatically interpolates missing values. Furthermore, assuming that different trace gases share the same distribution in the lateral direction, the method can be expanded to use the standard width and center position estimated for one trace gas directly when fitting the Gaussian function for another gas. This is particularly attractive for the combination of NO_2 and CO_2 observations from the future CO_2M mission. Since NO_2 can be measured with higher precision than CO_2 with current remote sensing instruments, images of NO_2 can provide a much stronger constraint on the width and position of the plume compared to the much noisier images of CO_2 (e.g., Reuter et al., 2019).

To increase the accuracy of the estimate, fluxes can be computed for multiple polygons downstream of the source. For a non-decaying gas, the estimated fluxes can simply be averaged. For a decaying gas, however, the emission, Q , can be obtained by additionally fitting the lifetime, τ , to the estimated fluxes. For point sources, where the pixel size is larger than the source area, a step function is assumed in the decay term; i.e., the flux increases stepwise from zero to the emission rate, Q , at the source location:

$$F_p(x, \tau) = Q \cdot D(x, \tau). \quad (14)$$

For sources such as cities, it is necessary to account for the effect of the source area by describing the emissions from the city, for example, as a Gaussian curve and the change in flux in the along-plume direction as the convolution of a Gaussian

curve and a decay term

$$F_d(x, \tau, \mu_a, \sigma_a) = Q \int_{-\infty}^{+\infty} D(x', \tau) g(x - x', \mu_a, \sigma_a) dx', \quad (15)$$

where μ_a and σ_a are the location and standard width of the Gaussian curve describing the extent of the area source. This is identical to the exponential modified Gaussian method but applied to a single image (Beirle et al., 2011; de Foy et al., 2015).

2.4.3 Integrated mass enhancement

The integrated mass enhancement approach computes the emission rate, Q , from the integrated total mass enhancement, M , of the detectable plume, \mathcal{P}_d , and a residence time, t (Frankenberg et al., 2016; Varon et al., 2018). The method can be derived by integrating the Gaussian plume model after subtracting the background (Eq. 4) over a large polygon up to distance x_2 (see Fig. 2c):

$$M = \int_{y_1}^{y_2} \int_{x_1}^{x_2} (G(x, y) - V_{bg}(x, y)) dx dy. \quad (16)$$

If the integration interval in the across-plume direction is sufficiently large to contain the full plume and $0 < x_1 < x_2$, we obtain

$$M = \int_{x_1}^{x_2} \frac{Q}{u} dx \quad (17)$$

or

$$Q = \frac{u}{L} M, \quad (18)$$

where u is the effective wind speed and $L = x_2 - x_1$ is the length of the detectable plume. Note that the derivation here is different from Varon et al. (2018), who only integrated over the detectable plume and computed L as the length scale defined as the square root of the plume area.

In practice, M can be computed as

$$M = \sum_{(i,j) \in \mathcal{P}_a} (V_{i,j} - V_{bg}) \cdot A_{i,j}, \quad (19)$$

where $A_{i,j}$ is the pixel area and \mathcal{P}_a is the integration area obtained by sufficiently expanding the detected plume in the across-wind direction to also include pixels with enhancements below the detection limit.

To apply the integrated mass enhancement method to a decaying gas, the decay term needs to be included in the integral:

$$M = \int_{y_1}^{y_2} \int_{x_1}^{x_2} G(x, y) \cdot D(x) dx dy. \quad (20)$$

As a result, the emission rate, Q , can be computed as

$$Q = \frac{1}{c} \frac{u}{L} M, \quad (21)$$

where the correction factor, c , corrects for the gas decay in the along-plume direction:

$$c = \frac{u\tau}{L} \left(\exp\left(-\frac{x_1}{u\tau}\right) - \exp\left(-\frac{x_2}{u\tau}\right) \right). \quad (22)$$

2.4.4 Divergence method

The divergence method was introduced by Beirle et al. (2011, 2019) for estimating NO_x emissions from TROPOMI NO_2 satellite observations. In the CoCO2 project, the method was adapted to estimate CO_2 emissions (Hakkarainen et al., 2022). The method is generally applied to a sequence of satellite images rather than a single image.

The divergence method is based on the continuity equation (Jacob, 1999; Koene et al., 2024) at steady state. According to this, the divergence of the flux field, \mathbf{F} , corresponds to the difference between emissions, E , and sinks, S :

$$\nabla \cdot \mathbf{F} = E - S. \quad (23)$$

The flux, \mathbf{F} , is defined as

$$\mathbf{F} = \begin{pmatrix} F_x \\ F_y \end{pmatrix} = \begin{pmatrix} V \cdot u \\ V \cdot v \end{pmatrix}, \quad (24)$$

where V is the vertical column density and u and v are the eastward and northward components of the plume transport speed, respectively, which correspond to the horizontal wind components weighted by the vertical distribution of the plume concentrations (Koene et al., 2024).

The NO_x sink can be calculated from the NO_2 columns as $S = \frac{fV}{\tau}$, where τ is the NO_x lifetime and f is the constant NO_x -to- NO_2 ratio. Assumptions about the lifetime and NO_x -to- NO_2 ratio are discussed in the next section.

In the case of gases like CO_2 with lifetimes much longer than the characteristic timescales of a plume (i.e., much longer than a few hours), the sink term can be neglected. For long-lived gases, however, it is critical to first subtract the atmospheric background before computing the divergence since the flux is not linear with column V due to the vertical change in wind speed (e.g., Hakkarainen et al., 2016).

To obtain the hotspot emissions, Q , from the emission map ($E = S + \nabla \cdot \mathbf{F}$), a peak-fitting algorithm can be applied that fits a two-dimensional Gaussian surface with a background at each source location (x_0, y_0) :

$$p(x, y) = \frac{Q}{2\pi\sigma_x\sigma_y\sqrt{1-r^2}} \cdot \exp\left(-\frac{(x-x_0)^2}{2\sigma_x^2(1-r^2)} - \frac{(y-y_0)^2}{2\sigma_y^2(1-r^2)} + \frac{r(x-x_0)(y-y_0)}{\sigma_x\sigma_y(1-r^2)}\right) + p_{\text{BG}}, \quad (25)$$

with standard widths (σ_x, σ_y) ; correlation r ; and a constant background in the divergence flux map, p_{BG} .

2.5 NO_2 to NO_x conversion

Many studies estimate emissions from NO_2 observations. However, NO_2 is emitted primarily as nitrogen monoxide (NO) and rapidly converted to NO_2 in the atmosphere. Emissions are therefore reported as nitrogen oxides ($\text{NO}_x = \text{NO}_2 + \text{NO}$) in NO_2 equivalents ($\text{kg NO}_2 \text{ s}^{-1}$). Since imaging remote sensing instruments only measure NO_2 column densities, it is necessary to convert NO_2 to NO_x using a NO_2 -to- NO_x ratio, f_V , representative of vertical columns

$$V_{\text{NO}_x}(x, y) = f_V(x, y) \cdot V_{\text{NO}_2}(x, y) \quad (26)$$

or a ratio, f_Q , for the estimated emissions

$$Q_{\text{NO}_x} = f_Q \cdot Q_{\text{NO}_2}. \quad (27)$$

If the conversion factor f_V is constant in space, f_V and f_Q are identical. However, the assumption of spatial (and temporal) homogeneity is generally not true for emission plumes, and more realistic models are currently being discussed (e.g., Hakkarainen et al., 2024; Meier et al., 2024).

2.6 Estimating annual emissions

Except for the divergence method, the methods described thus far allow us to quantify emissions from a single satellite image. To make statements about emissions over longer periods of time and to take advantage of the detection of a single source in multiple satellite images, one can compute a temporal average of the various computed emissions. Since the temporal coverage may be sparse and unevenly distributed over the year due to cloud cover and other factors, it may be useful to fill the gaps by making assumptions about the temporal variability. One possibility is to fit a seasonal cycle to the individual estimates using a low-order spline to approximate the time-varying emissions and to compute the annual mean emissions by integrating over the cycle (e.g., Kuhlmann et al., 2021). Extrapolating from a few single observations to an annual average is associated with significant uncertainties unless additional information on the true temporal variability is available (Hill and Nassar, 2019; Nassar et al., 2022). A further complication is the fact that satellite observations are often performed at the same time of day, providing almost no information on diurnal variability.

3 The *ddeg* Python library

In this paper, we describe version 1.0 of the library, which is provided in the Supplement. *ddeg* is an open-source library, whose latest release is available on the Python Package Index (PyPI; <https://pypi.org/project/ddeg/>). The issue

tracker and the development version of the library are available on the project's website at <https://gitlab.com/empa503/remote-sensing/ddeq> (last access: 24 May 2024). How to install *ddeq* is described in Appendix A. The documentation of the library is available in the Supplement and also published on Read the Docs (<https://ddeq.readthedocs.io/>, last access: 24 May 2024).

3.1 General framework

The library consists of four main components as shown in Fig. 3. (1) The data input component provides functions for reading remote sensing images (e.g., S5P/TROPOMI observations and synthetic CO₂M observations), hotspot locations (e.g., CoCO₂ point source database; Guevara et al., 2024) and (effective) wind fields (e.g., ERA5 reanalysis product). (2) The second component provides pre-processing of the data, which includes plume detection, conversion between coordinate systems, unit conversions, and estimation of the background field. (3) To quantify the emissions, five modules are provided implementing a Gaussian plume inversion (hereafter abbreviated as GP in `ddeq.gauss`), general and light cross-sectional flux method (hereafter CSF in `ddeq.csf` and LCSF in `ddeq.lcsf`), integrated mass enhancement (hereafter IME in `ddeq.ime`), and divergence method (hereafter DIV in `ddeq.div`). (4) Finally, the library provides functions for post-processing, which includes methods for estimating annual emissions from individual estimates, for converting NO₂ to NO_x emissions, for visualizing the results and for writing data output in standardized NetCDF format.

Listing 1 shows a simple code example demonstrating the individual steps required for estimating the CO₂ and NO_x emissions of the Janschwalde power plant in Germany. The Jupyter notebook with the full example is part of the *ddeq* library (notebooks/tutorial-introduction-to-ddeq.ipynb). First, the data input step corresponds to reading the location of sources, synthetic CO₂M data and ERA5 wind fields. ERA5 files were downloaded and prepared by the *ddeq* library (see Sect. 3.2). Second, a plume detection algorithm is used to locate the plume in the satellite image. A center curve is fitted to the data and natural coordinates are computed for the detected plume of each source. Third, the data are prepared for emission quantification by estimating and subtracting the background field and converting the CO₂ and NO₂ columns to kilograms per square meter. Fourth, CO₂ and NO_x emissions are estimated using the cross-sectional flux method as an example. The estimated NO₂ emissions are converted to NO_x using a conversion factor of 1.32 (in kg NO₂ s⁻¹). Finally, the results are saved as a NetCDF file and the data are visualized (see Fig. 4). In the following, the different components are described in more details. The implementation details are available in the documentation and the code itself in the Supplement.

3.2 Data input

ddeq requires that the location of sources used is known prior to estimating the emissions. The location and type of sources is therefore an important input for plume detection and emission quantification. *ddeq* makes extensive use of the *xarray* package (Hoyer and Hamman, 2017) for data handling to combine arrays with attributes. Sources are read from a comma-separated values (CSV) file into `xarray.Dataset`, which contains the source names (`source`), longitudes (`lon_o`), latitudes (`lat_o`), labels for visualization (`label`) and source types (`type`, which is currently either city or power plant). *ddeq* maintains a small list of sources as a CSV file that primarily contains cities and power plants used in previous studies by the developers. User-defined files containing other sources can be prepared in the same format. The file can be read with the `ddeq.misc.read_point_sources` function. In addition, *ddeq* can read the comprehensive CoCO₂ global emission point source database (Guevara et al., 2024) using `ddeq.coco2.read_ps_catalogue`. The catalogue is provided together with the library.

Remote sensing images are provided by airborne and space-based imaging spectrometers. *ddeq* handles images with `xarray.Dataset` with variables (i.e., `rs_data` in the example code), providing the trace gas columns and their uncertainties (e.g., CO₂ and CO_{2_precision}) that need to have a `units` attribute for automatic unit conversion and a `noise_level` attribute that is used as random uncertainty by the plume detection algorithm. In addition, the central longitude and latitude of the pixels need to be provided as `lon` and `lat`. If trace gases are provided as column-averaged mole fractions, surface pressure needs to be provided as `psurf` for unit conversion. Units are converted using the *ucat* Python library (<https://pypi.org/project/ucat/>, last access: 24 May 2024).

ddeq provides functions for automatically downloading and cropping TROPOMI NO₂ data for a given list of sources (`ddeq.download_S5P`). Furthermore, the library can read the synthetic CO₂M and Sentinel-5 data from the SMART-CARB dataset (Kuhlmann et al., 2020b) and the simulations from the library of plumes generated in the CoCO₂ project (Koene and Brunner, 2022). The synthetic datasets with known true emissions were used in the CoCO₂ project for method development and benchmarking (Santaren et al., 2024).

The final important input for emission quantification are wind fields, from which a representative transport speed of the trace gas within the plume is computed. *ddeq* provides functions for reading and downloading ERA5 reanalysis fields as well as reading wind fields from the SMARTCARB project (`ddeq.era5` and `ddeq.wind`). Wind speed and direction are provided either at the location of the source (`ddeq.wind.read_at_sources`) or as two-dimensional field that can also be spatially interpo-

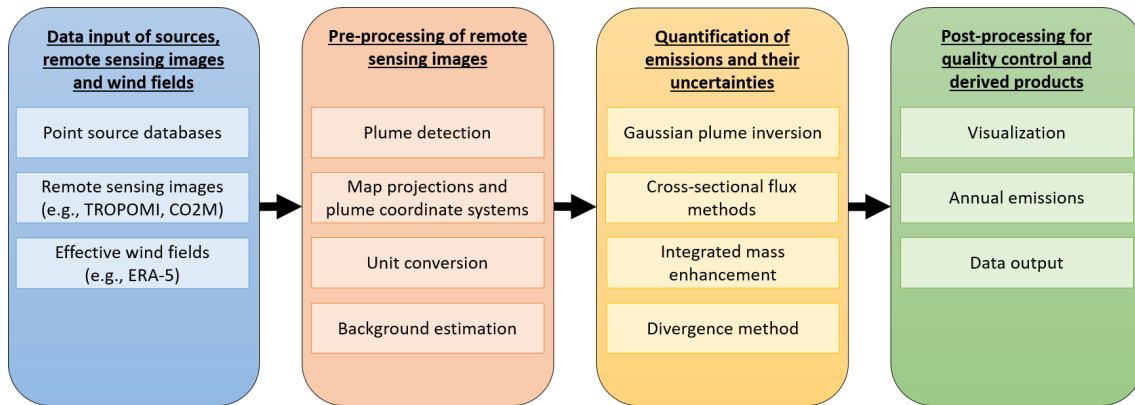


Figure 3. Data flow diagram illustrating the interactions between the different components in *ddeq*.

```

# (1) read hot spot locations, remote sensing data, winds
sources = ddeq.misc.read_point_sources()
rs_data = ddeq.smartcarb.read_level2('Sentinel_7_CO2_2015042311_o1670_10483.nc',
                                     co2_noise_level='low', no2_noise_level='high')
winds = ddeq.wind.read_wind_at_sources(rs_data.time, sources, product='ERA-5')

# (2a) plume detection from NO2 observations
rs_data = ddeq.plume.detect_plumes(rs_data, sources, variable='NO2', variable_std='NO2_std',
                                   filter_type='gaussian', filter_size='0.5')

# (2b) creation of plume-following coordinates
rs_data, curves = ddeq.plume_coords.compute_plume_line_and_coords(rs_data, radius=25e3)

# (3) background estimation and unit conversion
rs_data = ddeq.emissions.prepare_data(rs_data, 'CO2')
rs_data = ddeq.emissions.prepare_data(rs_data, 'NO2')

# (4) emission quantification (here: cross-sectional flux method)
results = ddeq.csf.estimate_emissions(rs_data, winds, sources, method='gauss',
                                       gases=['CO2', 'NO2'], f_model=1.32)

# (5) data output and visualization
results.to_netcdf(output_filename)
ddeq.vis.plot_csf_result(['CO2', 'NO2'], rs_data, winds, result, curves, 'Janschwalde')

```

Listing 1. Example of *ddeq* code for applying the CSF method to estimate CO₂ and NO_x emissions of the Janschwalde power plants in Germany. The full code is available in the library as a Jupyter notebook (notebooks/tutorial-introduction-to-ddeq.ipynb).

lated to the remote sensing image pixels, for example, for computing fluxes on Level-2 data (`ddeq.read_field`). Transport speeds can be computed by averaging the winds over a range of pressure levels, over the depth of the planet boundary layer or as weighted averages weighted by a vertical profile such as a typical emission profile of power plants (Brunner et al., 2019). The wind fields include the precision of the winds, currently hard-coded as 1 ms^{-1} , which is a rough estimate based on values used in previous studies (e.g., Varon et al., 2018; Reuter et al., 2019; Kuhlmann et al., 2021). Users are encouraged to replace the uncertainty with

a value suitable for their application and can also use their own wind data from other data sources.

3.3 Pre-processing

Data pre-processing includes a plume detection algorithm, conversion between coordinate systems, unit conversions and estimation of the background field. Which pre-processing steps are required or optional depends on the individual method.

One main pre-processing step is an algorithm for identifying the (a priori) location of the emission plume in the remote

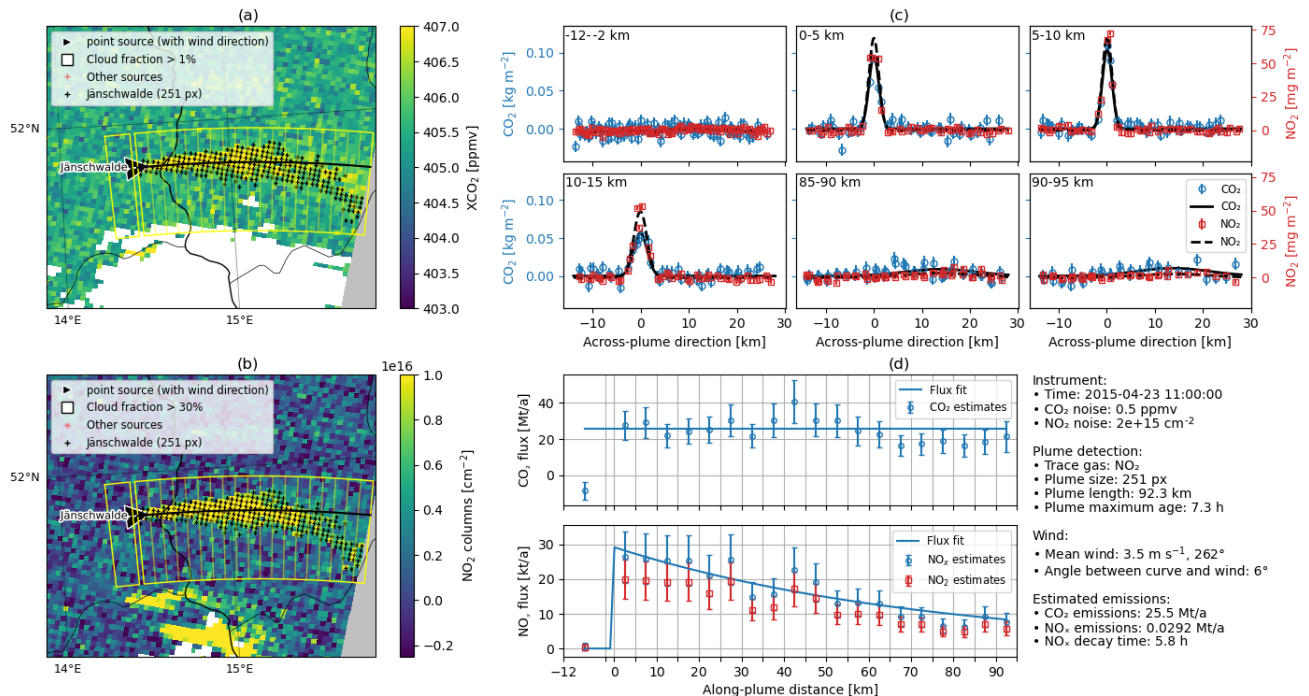


Figure 4. Plot created by the `ddeq.vis.plot_csf_result` command in the short example. Panels (a) and (b) show the CO₂ and NO₂ plumes from the Jämschwalde power plant in the synthetic CO2M images. Panel (c) shows the CO₂ and NO₂ columns in the across-plume direction for different along-plume distances with the two fitted Gaussian curves for computing the line densities. Panel (d) shows CO₂ and NO_x flux in the along-plume distance. The estimated CO₂ and NO_x emissions were 25.5 Mt a⁻¹ and 29.2 kt a⁻¹ for this example.

sensing image. *ddeq* implements an image segmentation algorithm for plume detection that is used as a pre-processing step by the GP, CSF and IME methods. As an alternative, the LCSF method determines the (a priori) plume location from the source location and the wind vector, which is currently part of the method's implementation. The divergence method does not require information about plume locations.

The image segmentation algorithm is described in detail in Kuhlmann et al. (2019, 2021). In short, the algorithm generates a Boolean mask, which is true where column densities are significantly enhanced above the background using Eq. (1). The signal is computed as the difference between the local mean and the background field. The noise (σ_V) is computed from the random and systematic uncertainty in the vertical column densities and uncertainties in the background. The local mean is computed by applying a uniform or a Gaussian filter to the image. The background field is computed using a median filter with a kernel that is large enough to contain areas outside the plume. The threshold (z_q) is computed from the probability (q) that the local mean is larger than the background given the uncertainty σ_V based on a statistical z test. In the Boolean mask, neighboring pixels are connected to regions and regions overlapping with known sources are labeled as plumes. The `ddeq.dplume.detected_plumes` function is used for applying the algorithm to the remote sensing images.

To create the natural coordinate system and compute the plume length, a center curve can be fitted to the plume mask using the `ddeq.plume_coords.compute_plume_line_and_coords` function. The center curve is described by two parabolic polynomials from which the along-plume coordinate x is computed as the arc length from the source and the across-plume coordinate y is computed as the distance from the center curve (see Kuhlmann et al., 2020a for details). The plume length is the arc length from the source to the most distant detectable pixel. Prior to fitting the center curve, longitude and latitude are converted to eastings and northings using a Cartesian coordinate reference system object from the *cartopy* library (Met Office, 2010–2015). If pixel corners are provided by the input dataset, the function also computes the pixel size (in m²), which is required, for example, by the integrated mass enhancement method.

Finally, `ddeq.emissions.prepare_data` can be used to estimate the background field and convert all trace gas fields to mass columns (in kg m⁻²) using the *ucat* Python library (Kuhlmann, 2022). *ddeq* implements a function for estimating the background field from pixels surrounding a detected plume (`ddeq.background.estimate`). The function masks all pixels where the signal-to-noise ratio is larger than the threshold z_q (using the results from the plume detection algorithm) and applies a normalized convolution to

estimate the background. Alternatively, the background can also be fitted directly by some emission quantification methods.

3.4 Emission quantification

3.4.1 Overview

Methods that are applied to single images all use the same order of parameters to estimate the emissions of sources:

```
results = ddeg.{method}.estimate_emissions(
    rs_data, winds, sources,
    [curves, gases, priors],
    variable='{gas}_minus_estimated_
    background_mass',
    [...])
```

where `{method}` can be `gauss`, `lcsf`, `csf` or `ime` for Gaussian plume inversion, light cross-sectional flux method, regular cross-sectional flux method and integrated mass enhancement method, respectively. Each method iterates over all sources provided by the sources dataset and estimates an emission if the source is inside the image. The method is applied to the variable in the remote sensing dataset (`rs_data`) given by the variable parameter (currently not implemented for LCSF and DIV). The default string is `{gas}_minus_estimated_background_mass`, which is the default variable name created by the pre-processing after subtracting the estimated background and converting to mass columns with units of kilogram per square meter. It is possible to provide a list of up to two gases for the CSF, GP and LCSF method (e.g. `gases = ['NO2', 'CO2']`). In this case, both gases are fitted either simultaneously (CSF) or after each other where the results from the first fit are used to constrain the second fit (GP and LCSF). The IME method can only be applied to a single gas. The `{gas}` placeholder in the variable parameter will be replaced with the names in `gases`.

The divergence method works on a series of remote sensing images and wind fields. The method therefore requires different inputs to read images and winds for a day on demand. The DIV method is called by

```
results = ddeg.div.estimate_emissions(
    datasets,
    wind_folder,
    sources, ...)
```

where `datasets` is a class with a `read_date` method that returns a list of remote sensing images for a given date and `wind_folder` is the path to a folder containing wind fields, for example, from the ERA5 reanalysis or the COSMO atmospheric model, for each day. Examples for the dataset are the `Level2Dataset` class in the `smartcarb` module or the `Level2TropomiDataset` in the `sats` module.

All methods return a results dataset, which is an `xarray.Dataset` with at least the variables `{gas}_estimated_emissions` and `{gas}_estimated_emissions_precision` with dimension source that can be saved using the `to_netcdf` method. The CSF method stores the results datasets using the `ddeg.misc.Result` class, which inherits from `dict` to handle dimensions of different size between sources. It has the methods `to_netcdf` and `from_file` to write and read the results. The results are saved as NetCDF files using a group for each source. The implementation of the `ddeg.misc.Result` class is necessary as NetCDF groups are currently not supported by Xarray.

The total uncertainty in the estimated emissions depends on the uncertainty in the trace gas columns, wind speed and background field. Additionally, all methods rely on simplifications and assumptions (e.g., steady-state conditions), which may result in non-negligible uncertainties that are currently not accounted for in the *ddeg* library (see Santaren et al., 2024, for details). To determine the total uncertainty, users are therefore encouraged to include such uncertainties although they might be difficult to assess and vary from case to case.

3.4.2 Gaussian plume inversion (GP)

The Gaussian plume inversion method implemented in *ddeg* fits Eq. (4) to the detected plume. The fit parameters are Q , V_{bg} , K , κ and coefficients of the center line. The wind speed is taken from the winds dataset, which, when using `ddeg.wind.read_at_sources`, is the effective wind speed at the source.

The plume center curve is described by a second-order Bézier curve which has three control points (one is centered at the known plume source location, and the other two are fitted parameters for the Gaussian curve), initialized along the curve, as it is already obtained in pre-processing. The reason for using a Bézier curve is that it behaves smoothly with respect to small changes in the control points, which is required for stabilizing the least-squares fit. In the case of a decaying gas, it is possible to fit a decay time, τ , by setting `fit_decay_times` to `true`.

The inversion consists of three simple Levenberg–Marquardt least-squares fitting steps. In the first fit, only the center curve, Q and κ are optimized. In the second fit, K and τ are optimized, and in the third fit, Q is optimized again. The implementation of three fits decreases the computation time and avoids overfitting.

The initial (prior) parameters for Q and τ need to be provided as a dictionary for each source and trace gas:

```
priors = {
    source: {
        'CO2': {'Q': 1000.0, 'tau': 1e10},
        'NO2': {'Q': 1.0, 'tau': 4.0*3600},
```

```

...
}

```

where Q is the source strength (in kg s^{-1}) and τ is the decay time in seconds. Other parameters are set to typical values in the `ddeq.gauss.generate_params` function. If two gases are provided, the values fitted for the first gas are used as initial conditions for the second gas (except for Q , which is reinitialized to the prior emission rate for that source). To constrain the fit for the second gas, we only allow for a small amount of deviation around the previously obtained Gaussian plume parameters (see `ddeq.gauss.gaussian_plume_estimates` for details).

The precision or random uncertainty in the Gaussian plume estimate is computed as

$$\sigma_Q = \sqrt{\sigma_{Q,\text{fit}}^2 + \sigma_u^2 \left(\frac{Q_{\text{fit}}}{u} \right)^2}, \quad (28)$$

where $\sigma_{Q,\text{fit}}$ is the estimated standard deviation of the fitted emission data and the second term accounts for the uncertainty in the wind speed as provided by the winds dataset (`wind_speed_precision`).

Estimates are rejected when no fit is found, when no standard deviation is estimated (i.e., if no good fit is possible), or when the emission rate is smaller than 0.1 times or larger than 1.9 times the prior expected emission rate (i.e., using $Q \pm 90\%$ uncertainty). These filters are currently fixed in the code. More flexible filters will be implemented in the future that are common to all methods.

3.4.3 Cross-sectional flux methods (CSF)

The CSF method is one of two different cross-sectional flux methods implemented in the library. The CSF method was originally developed as part of the SMARTCARB study (Kuhlmann et al., 2020a, 2021). It computes line densities in multiple polygons downstream of the source (see polygons in Fig. 4). Depending on the model parameter, line densities are obtained by either fitting a Gaussian curve (Eq. 13) using `model="gauss"` or integrating pixels inside the polygon using `model="sub_areas"`. The latter method divides the sub-polygons in the across-plume direction, computes the mean for each sub-polygon, and then integrates the mean values for each sub-polygon. This is done to account for missing values in the image. For NO_2 observations, the `f_model` parameter can be provided to convert NO_2 to NO_x line densities.

The line densities are converted to fluxes using the provided wind speeds at the sources. Emissions are obtained by fitting Eqs. (14) or (15) to the fluxes. For a non-decaying gas, the decay function is replaced by a Heaviside step function, which is 0 upstream and 1 downstream of the source. The uncertainty in the cross-sectional flux method is computed

by propagation of uncertainty from the single sounding precision that determines the uncertainty in each line density, which determines the uncertainty in the fitting parameters (Q and τ). Note that we assume that the wind speed uncertainty is independent of the number of pixels and the length of the plume.

To remove problematic cases, estimates are excluded if the angle between wind speed and center curve is larger than 45° , which often indicates erroneous plume detection. Estimates are also rejected if more than 5 pixels are detected upwind of the source.

3.4.4 Light cross-sectional flux method (LCSF)

The LCSF method is derived from the method originally developed by Zheng et al. (2020) to estimate the CO_2 emissions of Chinese cities and industrial areas from OCO-2 data. The method has then been adapted for routine and automatic estimation of isolated clusters of CO_2 emissions worldwide (Chevallier et al., 2020) and used to study the temporal variability in emissions using several years of OCO-2 and OCO-3 data (Chevallier et al., 2022).

Similarly to the CSF method, the LCSF method computes line densities by interpolating the pixels contained within polygons downstream of the source by a Gaussian function. The LCSF method uses the wind vector to construct a polygon that is 100 km wide in across-wind (perpendicular) direction and which extends downwind the source over a distance equal to the distance traveled by the wind in 1 h. A two-dimensional wind field read with `ddeq.wind.read_field` is used to determine the wind vector and for computing the wind speed used later for computing the flux. The `fit_backgrounds` parameter can be used to determine if a linearly changing background should be added to the Gaussian curve. Additional parameters are passed using a dictionary (`lcs_params`). For example, a NO_2 to NO_x conversion factor can be defined using the `f_NOx_NO2` key (3.5 being the default).

The uncertainty in the estimates provided by the LCSF method is computed by propagation of the uncertainty in the amplitude of the fitted Gaussian function. Several quality checks remove potentially unrealistic estimates: the fitting window should contain enough data pixels (50 pixels being the default) and the selected enhancements should have sufficient amplitude (i.e., larger than the standard deviation of the values in the polygon), the uncertainty (1σ) of the fitted width of the Gaussian function should be larger than 1 km and smaller than 5 km, and the estimated emissions should be larger than `min_est_emis` (0.0 being the default) and smaller than `max_est_emis` (infinity being the default), which, when calling `ddeq.lcsf.estimate_emissions`, are provided either as input parameters or as prior values similar to the implementation for the Gaussian plume inversion (i.e., using 0.1 and 1.9 Q).

3.4.5 Integrated mass enhancement (IME)

To first identify the location of the plume, the IME method uses the plume detection algorithm with the same settings as for the CSF and GP method. IME also requires the computation of the background field, the center line and the along- and across-plume distance for each pixel. In the across-plume direction, the integration interval is obtained by either computing the convex hull of the detected plume or applying a binary dilation to the Boolean mask of detected pixels. Both computations are carried out as part of the pre-processing inside the `ddeq.plume_coords.compute_plume_line_and_coords` function. In along-plume direction, the integration interval is defined by the `L_min` and `L_max` parameters. The default values are $L_{\min} = 0$ (i.e., the source location) and L_{\max} being set to the arc length of the most distant pixel in the integration area with 10 km subtracted from the value.

Missing values are interpolated using normalized convolution. The plume is discarded when more than 25 % of the pixels in the detected plume have been obtained through this gap filling. The effective wind speed is taken from the provided wind speeds at the source location. A decay time can be provided to compute the decay time correction term (Eq. 22).

The uncertainty in the emissions is calculated by propagation of uncertainty from the random uncertainty in the gas columns and the wind speed.

3.4.6 Divergence method (DIV)

In *ddeq*, the divergence method is applied to each source individually. The remote sensing images are provided as a dataset class as described above, where images are read for each date between `start_date` and `end_date`. To access the wind fields, the folder containing the wind files is provided along with a filename pattern (e.g., `ERA5-gnfra-%Y%m%d%H00.nc`).

The divergence method is performed in two steps. In the first step, the average divergence, $\nabla \cdot \mathbf{F}$, and sink, S , fields are calculated from the vertical column densities and wind fields. In the second step, the peak fitting is applied to derive the hotspot emissions.

In the first step, the trace gas fields can be denoised using a median filter, and depending on the choice of the user, either the local or the regional background is removed. Next, each pixel is associated with interpolated u and v wind components from the provided wind fields, and the vector field, \mathbf{F} , is derived using second-order accurate central differences as implemented in `numpy.gradient`. All the data are gridded to a regular kilometer grid defined by the user by the input parameters `lon_km`, `lat_km` and `grid_resol` given in kilometers. The divergence of the vector field is calculated before averaging over all available images. The computation of the divergence before averaging is preferred for remote sensing images with data gaps, for example, due to clouds

(Hakkarainen et al., 2022; Koene et al., 2024). For NO_x , the sink term is calculated from the averaged vertical column densities, assuming a lifetime of 4 h.

In the second step, the peak fitting is performed using a least-squares fit between the averaged emission field, $E = S + \nabla \cdot \mathbf{F}$, and the peak fitting function, Eq. (25). The optimization is first done using the Nelder–Mead method from the *scipy* library. The uncertainty in the estimated emissions is obtained from the mismatch between the emission field, E , and peak fitting function; i.e., we assume that the chi square of the fit is the number of degrees of freedom. *ddeq* also implements the adaptive Metropolis algorithm (Haario et al., 2001) for sampling the posterior distribution (assuming a non-informative prior) to obtain an optimized estimate of the fitting parameters and their uncertainty.

3.5 Post-processing

The post processing step provides functions for data visualization (`ddeq.vis`) and for estimating annual emissions from individual estimates (`ddeq.timeseries`). In addition, it is possible to apply scaling factors to estimated emissions to convert NO_2 to NO_x emissions.

To convert NO_2 to NO_x emissions, it is possible to scale all estimated NO_2 emissions in the results dataset, i.e., variables that start with `NO2` and have units of kg s^{-1} , to NO_x emissions using the scalar f . The function `convert_NO2_to_NOx_emissions(results, f=1.32)` is implemented in the `ddeq.emissions` module. A more complex NO_2 to NO_x conversion will be developed in the future to account for non-constant conversion factors.

ddeq provides functions for fitting a seasonal cycle using a cubic C spline with periodic boundary conditions to a time series of estimated emissions (`ddeq.timeseries`). Annual emissions and their uncertainties are obtained by integrating the seasonal cycle, as shown in the following code:

```
fit, model, _, _, _ = ddeq.timeseries.fit(
    times,
    estimates,
    estimates_precision
)

annual, annual_precision = model.integrate(
    fit['x'],
    fit['x_std']
)
```

A detailed example is available as a Jupyter notebook (`example_annual_emissions.ipynb`).

Finally, *ddeq* provides functions for visualizing the remote sensing images, the plume detection, and the results of the emission quantification using `ddeq.vis.plot_{method}_result` (e.g., Fig. 4). The library includes several Jupyter notebooks, with demonstrations for the different methods.

4 Conclusions

There have been many studies quantifying the emissions of hotspots in remote sensing images in the past and we expect significantly more studies in the future with the new generation of imaging satellites. The ability to monitor emissions of air pollutants and greenhouse gases is an important capability of remote sensing instruments. As information on emissions from hotspots can be politically sensitive, it is essential that the emission estimates are reliable, i.e., that the methodology used can be verified, for which an open-source library will be an invaluable tool.

ddeq is a community library that is open to new users and developers. The Jupyter notebooks with tutorials and examples make it easy for new users to learn how to apply the library to different datasets. *ddeq* is hosted as an open-source project on GitLab, making it possible for users and developers to submit bug reports and feature request. Finally, developers can further expand existing methods in *ddeq* or implement new methods. *ddeq* makes it possible to compare these additions with existing methods in a reproducible way, for example, using the same input data, increasing the reliability of estimates of anthropogenic emissions of air pollutants and greenhouse gases.

The *ddeq* library provides a general framework where new methods and options can be easily implemented. It will be further developed in the future, for example, in the CO2MVS Research on Supplementary Observations (CORSO) project, where *ddeq* will be used for quantifying CO and NO_x emissions from Sentinel-5P/TROPOMI and other satellite instruments. Furthermore, *ddeq* will be used for quantifying CH₄ emissions from observations of the newest Airborne Visible/Infrared Imaging Spectrometer (AVIRIS-4) operated by a consortium of Swiss research institutes led by the University of Zurich (Green et al., 2022). Future updates will further harmonize the different implementations, allowing for more flexibility and even better comparability between the different methods. It is also planned for it to provide support for machine-learning models such as the plume segmentation algorithm developed by Dumont Le Brazidec et al. (2023).

One main goal of the CoCO₂ project is the development of prototype systems for anthropogenic emission monitoring. The *ddeq* Python library for data-driven emission quantification presents such a prototype for emission monitoring of hotspots using lightweight approaches. It was developed and used in the CoCO₂ project to benchmark different emission quantification methods with the aim to identify the most suitable and accurate approaches to be implemented in the prototype of the European CO₂ monitoring system (Hakkarainen et al., 2024; Santaren et al., 2024).

Appendix A: Installation and interactive computing environment

The latest version of the *ddeq* library can be installed using the Python package manager, pip:

```
python -m pip install ddeq
```

The development version of *ddeq* can be cloned using

```
git clone https://gitlab.com/empa503/remote-sensing/ddeq.git
```

and then installed using pip. For the development version, it is useful to install it as editable (-e option):

```
pip install -e ddeq/
```

The version described in this paper is available in the Supplement. The version can be installed using conda and pip with the following steps:

```
# create Python environment
conda create -n ddeq-test python=3.9
conda activate ddeq-test

# install additional packages
conda install jupyterlab
conda install pycurl

# unzip tar.gz file in the Supplement
and install using pip
tar -xzf ddeq-1.0.tar.gz
python -m pip install ddeq-1.0/

# start JupyterLab
jupyter lab --notebook-dir
```

The tutorial and examples are included in the notebooks folder.

Code and data availability. The *ddeq* version 1.0 described in this document is available in the Supplement. The code repository is available on GitLab (<https://gitlab.com/empa503/remote-sensing/ddeq>, Kuhlmann, 2024a). The synthetic satellite observations and wind fields from the SMARTCARB dataset are available on Zenodo at <https://doi.org/10.5281/zenodo.4048228> (Kuhlmann et al., 2020b) and <https://doi.org/10.5281/zenodo.10684753> (Kuhlmann, 2024b).

Supplement. The supplement related to this article is available online at: <https://doi.org/10.5194/gmd-17-4773-2024-supplement>.

Author contributions. The paper was written by GK with input from all the co-authors. GK developed the original *ddeq* library as part of the SMARTCARB project and merged all implementations from the CoCO₂ and CORSO projects; EK developed the Gaussian

plume inversion method; SM implemented the code for downloading and pre-processing the ERA5 and Sentinel-5P/TROPOMI data; DS, GB and FC developed the light cross-sectional flux method; and JH, JN and LA implemented the divergence method. The project was coordinated by DB, GB, GK and JT.

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

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims made in the text, published maps, institutional affiliations, or any other geographical representation in this paper. While Copernicus Publications makes every effort to include appropriate place names, the final responsibility lies with the authors.

Acknowledgements. We would like to thank the ICOS Carbon Portal for providing access to their JupyterLab servers, which were used for code development and data sharing.

Financial support. This research has been supported by the European Space Agency (contract no. 4000119599/16/ NL/FF/mg); the EU Horizon 2020 program (grant nos. 776186 and 958927); the Horizon Europe Digital, Industry and Space program (grant no. 101082194); the Swiss State Secretariat for Education, Research and Innovation (grant no. 22.00422); and Research Council of Finland projects (grant nos. 353082 and 359196).

Review statement. This paper was edited by Yilong Wang and reviewed by Robert Roland Nelson and one anonymous referee.

References

- Beirle, S., Boersma, K. F., Platt, U., Lawrence, M. G., and Wagner, T.: Megacity Emissions and Lifetimes of Nitrogen Oxides Probed from Space, *Science*, 333, 1737–1739, <https://doi.org/10.1126/science.1207824>, 2011.
- Beirle, S., Borger, C., Dörner, S., Li, A., Hu, Z., Liu, F., Wang, Y., and Wagner, T.: Pinpointing nitrogen oxide emissions from space, *Sci. Adv.*, 5, eaax9800, <https://doi.org/10.1126/sciadv.aax9800>, 2019.
- Berchet, A., Sollum, E., Thompson, R. L., Pison, I., Thanwerdas, J., Broquet, G., Chevallier, F., Aalto, T., Berchet, A., Bergamaschi, P., Brunner, D., Engelen, R., Fortems-Cheiney, A., Gerbig, C., Groot Zwaftink, C. D., Haussaire, J.-M., Henne, S., Houweling, S., Karstens, U., Kutsch, W. L., Luijkx, I. T., Monteil, G., Palmer, P. I., van Peet, J. C. A., Peters, W., Peylin, P., Potier, E., Rödenbeck, C., Saunio, M., Scholze, M., Tsuruta, A., and Zhao, Y.: The Community Inversion Framework v1.0: a unified system for atmospheric inversion studies, *Geosci. Model Dev.*, 14, 5331–5354, <https://doi.org/10.5194/gmd-14-5331-2021>, 2021.
- Bieser, J., Aulinger, A., Matthias, V., Quante, M., and Denier van der Gon, H.: Vertical emission profiles for Europe based on plume rise calculations, *Environ. Pollut.*, 159, 2935–2946, <https://doi.org/10.1016/j.envpol.2011.04.030>, 2011.
- Bovensmann, H., Buchwitz, M., Burrows, J. P., Reuter, M., Krings, T., Gerilowski, K., Schneising, O., Heymann, J., Tretner, A., and Erzinger, J.: A remote sensing technique for global monitoring of power plant CO₂ emissions from space and related applications, *Atmos. Meas. Tech.*, 3, 781–811, <https://doi.org/10.5194/amt-3-781-2010>, 2010.
- Broquet, G., Bréon, F.-M., Renault, E., Buchwitz, M., Reuter, M., Bovensmann, H., Chevallier, F., Wu, L., and Ciais, P.: The potential of satellite spectro-imagery for monitoring CO₂ emissions from large cities, *Atmos. Meas. Tech.*, 11, 681–708, <https://doi.org/10.5194/amt-11-681-2018>, 2018.
- Brunner, D., Kuhlmann, G., Marshall, J., Clément, V., Fuhrer, O., Broquet, G., Löscher, A., and Meijer, Y.: Accounting for the vertical distribution of emissions in atmospheric CO₂ simulations, *Atmos. Chem. Phys.*, 19, 4541–4559, <https://doi.org/10.5194/acp-19-4541-2019>, 2019.
- Chevallier, F., Zheng, B., Broquet, G., Ciais, P., Liu, Z., Davis, S. J., Deng, Z., Wang, Y., Bréon, F.-M., and O'Dell, C. W.: Local anomalies in the column-averaged dry air mole fractions of carbon dioxide across the globe during the first months of the coronavirus recession, *Geophys. Res. Lett.*, 47, e2020GL090244, <https://doi.org/10.1029/2020GL090244>, 2020.
- Chevallier, F., Broquet, G., Zheng, B., Ciais, P., and Eldering, A.: Large CO₂ Emitters as Seen From Satellite: Comparison to a Gridded Global Emission Inventory, *Geophys. Res. Lett.*, 49, e2021GL097540, <https://doi.org/10.1029/2021GL097540>, 2022.
- Crippa, M., Guizzardi, D., Banja, M., Solazzo, E., Muntean, M., Schaaf, E., Pagani, F., Monforti-Ferrario, F., Olivier, J., Quadrelli, R., Risquez Martin, A., Taghavi-Moharamli, P., G., G., Rossi, S., Oom, D., Branco, A., San-Miguel, J., and Vignati, E.: CO₂ emissions of all world countries – JRC/IEA/PBL 2022, Tech. Rep. JRC130363, Luxembourg (Luxembourg), <https://doi.org/10.2760/07904>, 2022.
- de Foy, B., Lu, Z., Streets, D. G., Lamsal, L. N., and Duncan, B. N.: Estimates of power plant NO_x emissions and lifetimes from OMI NO₂ satellite retrievals, *Atmos. Environ.*, 116, 1–11, <https://doi.org/10.1016/j.atmosenv.2015.05.056>, 2015.
- Dumont Le Brazidec, J., Vanderbecken, P., Farchi, A., Bocquet, M., Lian, J., Broquet, G., Kuhlmann, G., Danjou, A., and Lauvaux, T.: Segmentation of XCO₂ images with deep learning: application to synthetic plumes from cities and power plants, *Geosci. Model Dev.*, 16, 3997–4016, <https://doi.org/10.5194/gmd-16-3997-2023>, 2023.
- Dumont Le Brazidec, J., Vanderbecken, P., Farchi, A., Broquet, G., Kuhlmann, G., and Bocquet, M.: Deep learning applied to CO₂ power plant emissions quantification using simulated satellite images, *Geosci. Model Dev.*, 17, 1995–2014, <https://doi.org/10.5194/gmd-17-1995-2024>, 2024.
- Eldering, A., Taylor, T. E., O'Dell, C. W., and Pavlick, R.: The OCO-3 mission: measurement objectives and expected performance based on 1 year of simulated data, *Atmos. Meas. Tech.*, 12, 2341–2370, <https://doi.org/10.5194/amt-12-2341-2019>, 2019.
- ESA Earth and Mission Science Division: Copernicus CO₂ Monitoring Mission Requirements Document (MRD), Tech. rep.,

- version 3.0, 1 October 2020, https://esamultimedia.esa.int/docs/EarthObservation/CO2M_MRD_v3.0_20201001_Issued.pdf (last access: 24 May 2024), 2020.
- Finch, D. P., Palmer, P. I., and Zhang, T.: Automated detection of atmospheric NO₂ plumes from satellite data: a tool to help infer anthropogenic combustion emissions, *Atmos. Meas. Tech.*, 15, 721–733, <https://doi.org/10.5194/amt-15-721-2022>, 2022.
- Fioletov, V. E., McLinden, C. A., Krotkov, N., and Li, C.: Lifetimes and emissions of SO₂ from point sources estimated from OMI, *Geophys. Res. Lett.*, 42, 1969–1976, <https://doi.org/10.1002/2015GL063148>, 2015.
- Frankenberg, C., Thorpe, A. K., Thompson, D. R., Hulley, G., Kort, E. A., Vance, N., Borchardt, J., Krings, T., Gerilowski, K., Sweeney, C., Conley, S., Bue, B. D., Aubrey, A. D., Hook, S., and Green, R. O.: Airborne methane remote measurements reveal heavy-tail flux distribution in Four Corners region, *P. Natl. Acad. Sci. USA*, 113, 9734–9739, <https://doi.org/10.1073/pnas.1605617113>, 2016.
- Fujinawa, T., Kuze, A., Suto, H., Shiomi, K., Kanaya, Y., Kawashima, T., Kataoka, F., Mori, S., Eskes, H., and Tanimoto, H.: First Concurrent Observations of NO₂ and CO₂ From Power Plant Plumes by Airborne Remote Sensing, *Geophys. Res. Lett.*, 48, e2021GL092685, <https://doi.org/10.1029/2021GL092685>, 2021.
- Green, R. O., Schaepman, M. E., Mouroulis, P., Geier, S., Shaw, L., Hueini, A., Bernas, M., McKinley, I., Smith, C., Wehbe, R., Eastwood, M., Vinckier, Q., Liggett, E., Zandbergen, S., Thompson, D., Sullivan, P., Sarture, C., Van Gorp, B., and Helmlinger, M.: Airborne Visible/Infrared Imaging Spectrometer 3 (AVIRIS-3), in: 2022 IEEE Aerospace Conference (AERO), 1–10, <https://doi.org/10.1109/AERO53065.2022.9843565>, 2022.
- Guevara, M., Enciso, S., Tena, C., Jorba, O., Dellaert, S., Denier van der Gon, H., and Pérez García-Pando, C.: A global catalogue of CO₂ emissions and co-emitted species from power plants, including high-resolution vertical and temporal profiles, *Earth Syst. Sci. Data*, 16, 337–373, <https://doi.org/10.5194/essd-16-337-2024>, 2024.
- Haario, H., Saksman, E., and Tamminen, J.: An adaptive Metropolis algorithm, *Bernoulli*, 7, 223–242, <https://projecteuclid.org/443/euclid.bj/1080222083> (last access: 24 May 2024), 2001.
- Hakkarainen, J., Ialongo, I., and Tamminen, J.: Direct space-based observations of anthropogenic CO₂ emission areas from OCO-2, *Geophys. Res. Lett.*, 43, 11400–11406, <https://doi.org/10.1002/2016GL070885>, 2016.
- Hakkarainen, J., Ialongo, I., Koene, E., Szelag, M., Tamminen, J., Kuhlmann, G., and Brunner, D.: Analyzing local carbon dioxide and nitrogen oxide emissions from space using the divergence method: An application to the synthetic SMARTCARB dataset, *Front. Remote Sens.*, 3, 878731, <https://doi.org/10.3389/frsen.2022.878731>, 2022.
- Hakkarainen, J., Kuhlmann, G., Koene, E., Santaren, D., Meier, S., Krol, M. C., Ialongo, I., Chevallier, F., Tamminen, J., Brunner, D., and Broquet, G.: Analyzing nitrogen dioxide to nitrogen oxide scaling factors for data-driven satellite-based emission estimation methods: A case study of Matimba/Medupi power stations in South Africa, *Atmos. Pollut. Res.*, 15, 102171, <https://doi.org/10.1016/j.apr.2024.102171>, 2024.
- Hersbach, H., Bell, B., Berrisford, P., Biavati, G., Horányi, A., Muñoz Sabater, J., Nicolas, J., Peubey, C., Radu, R., Rozum, I., Schepers, D., Simmons, A., Soci, C., Dee, D., and Thépaut, J.-N.: ERA5 hourly data on pressure levels from 1940 to present, Copernicus Climate Change Service (C3S) Climate Data Store (CDS) [data set], <https://doi.org/10.24381/cds.bd0915c6>, 2023.
- Hill, T. and Nassar, R.: Pixel Size and Revisit Rate Requirements for Monitoring Power Plant CO₂ Emissions from Space, *Remote Sens.*, 11, 1608, <https://doi.org/10.3390/rs11131608>, 2019.
- Hoyer, S. and Hamman, J.: xarray: N-D labeled arrays and datasets in Python, *J. Open Res. Softw.*, 5, 10, <https://doi.org/10.5334/jors.148>, 2017.
- Jacob, D. J.: Introduction to atmospheric chemistry, Princeton university press, ISBN 0691001855, 9780691001852, 1999.
- Janssens-Maenhout, G., Pinty, B., Dowell, M., Zunker, H., Andersson, E., Balsamo, G., Bézy, J.-L., Brunhes, T., Bösch, H., Bockov, B., Brunner, D., Buchwitz, M., Crisp, D., Ciais, P., Counet, P., Dee, D., Denier van der Gon, H., Dolman, H., Drinkwater, M., Dubovik, O., Engelen, R., Fehr, T., Fernandez, V., Heimann, M., Holmlund, K., Houweling, S., Husband, R., Juvyns, O., Kentarchos, A., Landgraf, J., Lang, R., Löscher, A., Marshall, J., Meijer, Y., Nakajima, M., Palmer, P., Peylin, P., Rayner, P., Scholze, M., Sierk, B., Tamminen, J., and Veefkind, P.: Toward an operational anthropogenic CO₂ emissions monitoring and verification support capacity, *B. Am. Meteorol. Soc.*, 101, E1439–E1451, <https://doi.org/10.1175/BAMS-D-19-0017.1>, 2020.
- Jongaramrungruang, S., Thorpe, A. K., Matheou, G., and Frankenberg, C.: MethaNet – An AI-driven approach to quantifying methane point-source emission from high-resolution 2-D plume imagery, *Remote Sens. Environ.*, 269, 112809, <https://doi.org/10.1016/j.rse.2021.112809>, 2022.
- Joyce, P., Ruiz Villena, C., Huang, Y., Webb, A., Gloor, M., Wagner, F. H., Chipperfield, M. P., Barrio Guilló, R., Wilson, C., and Boesch, H.: Using a deep neural network to detect methane point sources and quantify emissions from PRISMA hyperspectral satellite images, *Atmos. Meas. Tech.*, 16, 2627–2640, <https://doi.org/10.5194/amt-16-2627-2023>, 2023.
- Kaminski, T., Scholze, M., Rayner, P., Houweling, S., Voßbeck, M., Silver, J., Lama, S., Buchwitz, M., Reuter, M., Knorr, W., Chen, H. W., Kuhlmann, G., Brunner, D., Dellaert, S., Denier van der Gon, H., Super, I., Löscher, A., and Meijer, Y.: Assessing the Impact of Atmospheric CO₂ and NO₂ Measurements From Space on Estimating City-Scale Fossil Fuel CO₂ Emissions in a Data Assimilation System, *Front. Remote Sens.*, 3, 887456, <https://doi.org/10.3389/frsen.2022.887456>, 2022.
- Koene, E. and Brunner, D.: CoCO2 WP4.1 Library of Plumes, Zenodo [data set], <https://doi.org/10.5281/zenodo.7448144>, 2022.
- Koene, E. F. M., Brunner, D., and Kuhlmann, G.: On the theory of the divergence method for quantifying source emissions from satellite observations, *J. Geophys. Res.-Atmos.*, 129, e2023JD039904, <https://doi.org/10.1029/2023JD039904>, 2024.
- Krings, T., Gerilowski, K., Buchwitz, M., Hartmann, J., Sachs, T., Erzinger, J., Burrows, J. P., and Bovensmann, H.: Quantification of methane emission rates from coal mine ventilation shafts using airborne remote sensing data, *Atmos. Meas. Tech.*, 6, 151–166, <https://doi.org/10.5194/amt-6-151-2013>, 2013.
- Kuhlmann, G.: u-cat: Unit conversion for atmospheric trace gases, Gitlab [code], <https://gitlab.com/empa503/general-tools/u-cat> (last access: 28 November 2023), 2022.
- Kuhlmann, G.: *ddeq*, Gitlab [code], <https://gitlab.com/empa503/remote-sensing/ddeq>, last access: 24 May 2024a.

- Kuhlmann, G.: SMARTCARB wind fields, Zenodo [data set], <https://doi.org/10.5281/zenodo.10684753>, 2024b.
- Kuhlmann, G., Broquet, G., Marshall, J., Clément, V., Löscher, A., Meijer, Y., and Brunner, D.: Detectability of CO₂ emission plumes of cities and power plants with the Copernicus Anthropogenic CO₂ Monitoring (CO2M) mission, *Atmos. Meas. Tech.*, 12, 6695–6719, <https://doi.org/10.5194/amt-12-6695-2019>, 2019.
- Kuhlmann, G., Brunner, D., Broquet, G., and Meijer, Y.: Quantifying CO₂ emissions of a city with the Copernicus Anthropogenic CO₂ Monitoring satellite mission, *Atmos. Meas. Tech.*, 13, 6733–6754, <https://doi.org/10.5194/amt-13-6733-2020>, 2020a.
- Kuhlmann, G., Clément, V., Marshall, J., Fuhrer, O., Broquet, G., Schnadt-Poberaj, C., Löscher, A., Meijer, Y., and Brunner, D.: Synthetic XCO₂, CO and NO₂ observations for the CO2M and Sentinel-5 satellites, Zenodo [data set], <https://doi.org/10.5281/zenodo.4048228>, 2020b.
- Kuhlmann, G., Henne, S., Meijer, Y., and Brunner, D.: Quantifying CO₂ Emissions of Power Plants With CO₂ and NO₂ Imaging Satellites, *Front. Remote Sens.*, 2, 689838, <https://doi.org/10.3389/frsen.2021.689838>, 2021.
- Kuhlmann, G., Chan, K. L., Donner, S., Zhu, Y., Schwaerzel, M., Dörner, S., Chen, J., Hueni, A., Nguyen, D. H., Damm, A., Schütt, A., Dietrich, F., Brunner, D., Liu, C., Buchmann, B., Wagner, T., and Wenig, M.: Mapping the spatial distribution of NO₂ with in situ and remote sensing instruments during the Munich NO₂ imaging campaign, *Atmos. Meas. Tech.*, 15, 1609–1629, <https://doi.org/10.5194/amt-15-1609-2022>, 2022.
- Leguijt, G., Maasackers, J. D., Denier van der Gon, H. A. C., Segers, A. J., Borsdorff, T., and Aben, I.: Quantification of carbon monoxide emissions from African cities using TROPOMI, *Atmos. Chem. Phys.*, 23, 8899–8919, <https://doi.org/10.5194/acp-23-8899-2023>, 2023.
- Lei, R., Feng, S., Danjou, A., Broquet, G., Wu, D., Lin, J. C., O'Dell, C. W., and Lauvaux, T.: Fossil fuel CO₂ emissions over metropolitan areas from space: A multi-model analysis of OCO-2 data over Lahore, Pakistan, *Remote Sens. Environ.*, 264, 112625, <https://doi.org/10.1016/j.rse.2021.112625>, 2021.
- Meier, S., Koene, E., Krol, M., Brunner, D., Damm, A., and Kuhlmann, G.: A light-weight NO₂ to NO_x conversion model for quantifying NO_x emissions of point sources from NO₂ satellite observations, *EGU sphere* [preprint], <https://doi.org/10.5194/egusphere-2024-159>, 2024.
- Met Office: Cartopy: a cartographic python library with a Matplotlib interface, Exeter, Devon, <https://scitools.org.uk/cartopy> (last access: 24 May 2024), 2010–2015.
- Nassar, R., Hill, T. G., McLinden, C. A., Wunch, D., Jones, D. B. A., and Crisp, D.: Quantifying CO₂ Emissions From Individual Power Plants From Space, *Geophys. Res. Lett.*, 44, 10045–10053, <https://doi.org/10.1002/2017GL074702>, 2017.
- Nassar, R., Moeini, O., Mastrogiacomo, J.-P., O'Dell, C. W., Nelson, R. R., Kiel, M., Chatterjee, A., Eldering, A., and Crisp, D.: Tracking CO₂ emission reductions from space: A case study at Europe's largest fossil fuel power plant, *Front. Remote Sens.*, 3, <https://doi.org/10.3389/frsen.2022.1028240>, 2022.
- Reuter, M., Buchwitz, M., Schneising, O., Krautwurst, S., O'Dell, C. W., Richter, A., Bovensmann, H., and Burrows, J. P.: Towards monitoring localized CO₂ emissions from space: co-located regional CO₂ and NO₂ enhancements observed by the OCO-2 and S5P satellites, *Atmos. Chem. Phys.*, 19, 9371–9383, <https://doi.org/10.5194/acp-19-9371-2019>, 2019.
- Santaren, D., Hakkarainen, J., Kuhlmann, G., Koene, E., Chevallier, F., Ialongo, I., Lindqvist, H., Nurmela, J., Tamminen, J., Amoros, L., Brunner, D., and Broquet, G.: Benchmarking data-driven inversion methods for the estimation of local CO₂ emissions from XCO₂ and NO₂ satellite images, *Atmos. Meas. Tech. Discuss.* [preprint], <https://doi.org/10.5194/amt-2023-241>, in review, 2024.
- Tack, F., Merlaud, A., Meier, A. C., Vlemmix, T., Ruhtz, T., Iordache, M.-D., Ge, X., van der Wal, L., Schuettmeyer, D., Ardelean, M., Calcan, A., Constantin, D., Schönhardt, A., Meuleman, K., Richter, A., and Van Roozendaal, M.: Intercomparison of four airborne imaging DOAS systems for tropospheric NO₂ mapping – the AROMAPEX campaign, *Atmos. Meas. Tech.*, 12, 211–236, <https://doi.org/10.5194/amt-12-211-2019>, 2019.
- Thorpe, A. K., Frankenberg, C., Thompson, D. R., Duren, R. M., Aubrey, A. D., Bue, B. D., Green, R. O., Gerilowski, K., Krings, T., Borchardt, J., Kort, E. A., Sweeney, C., Conley, S., Roberts, D. A., and Dennison, P. E.: Airborne DOAS retrievals of methane, carbon dioxide, and water vapor concentrations at high spatial resolution: application to AVIRIS-NG, *Atmos. Meas. Tech.*, 10, 3833–3850, <https://doi.org/10.5194/amt-10-3833-2017>, 2017.
- Varon, D. J., Jacob, D. J., McKeever, J., Jervis, D., Durak, B. O. A., Xia, Y., and Huang, Y.: Quantifying methane point sources from fine-scale satellite observations of atmospheric methane plumes, *Atmos. Meas. Tech.*, 11, 5673–5686, <https://doi.org/10.5194/amt-11-5673-2018>, 2018.
- Veefkind, J., Aben, I., McMullan, K., Förster, H., de Vries, J., Otter, G., Claas, J., Eskes, H., de Haan, J., Kleipool, Q., van Weele, M., Hasekamp, O., Hoogeveen, R., Landgraf, J., Snel, R., Tol, P., Ingmann, P., Voors, R., Kruizinga, B., Vink, R., Visser, H., and Levelt, P.: TROPOMI on the ESA Sentinel-5 Precursor: A GMES mission for global observations of the atmospheric composition for climate, air quality and ozone layer applications, *Remote Sens. Environ.*, 120, 70–83, <https://doi.org/10.1016/j.rse.2011.09.027>, 2012.
- Wang, Y., Broquet, G., Bréon, F.-M., Lespinas, F., Buchwitz, M., Reuter, M., Meijer, Y., Loescher, A., Janssens-Maenhout, G., Zheng, B., and Ciais, P.: PMIF v1.0: assessing the potential of satellite observations to constrain CO₂ emissions from large cities and point sources over the globe using synthetic data, *Geosci. Model Dev.*, 13, 5813–5831, <https://doi.org/10.5194/gmd-13-5813-2020>, 2020.
- Ye, X., Lauvaux, T., Kort, E. A., Oda, T., Feng, S., Lin, J. C., Yang, E. G., and Wu, D.: Constraining Fossil Fuel CO₂ Emissions From Urban Area Using OCO-2 Observations of Total Column CO₂, *J. Geophys. Res.-Atmos.*, 125, e2019JD030528, <https://doi.org/10.1029/2019JD030528>, 2020.
- Zheng, B., Chevallier, F., Ciais, P., Broquet, G., Wang, Y., Lian, J., and Zhao, Y.: Observing carbon dioxide emissions over China's cities and industrial areas with the Orbiting Carbon Observatory-2, *Atmos. Chem. Phys.*, 20, 8501–8510, <https://doi.org/10.5194/acp-20-8501-2020>, 2020.