A multiple-gas inverse model for estimating sector-level greenhouse gas emissions using observations of secondary secondary gases or isotopes

Alice E. Ramsden

This document describes the setup of the multiple-gas inverse model. This includes the equations used to produce mole fraction and isotope observations and details of the Markov Chain Monte Carlo (MCMC) process used to produce final emissions estimates.

The model can be used with a range of different inputs. The following are discussed in more detail in this document:

- Mole fraction and/or isotope secondary trace gas observations and the different forward models used for the different observation types.
- Filtering of observations.
- Solving for emissions scaling factors for the primary gas, from any number of sectors.
- Solving for boundary conditions scaling factors for the the primary and secondary mole fraction gases. Or, an option not to include boundary conditions and only solve for local emissions.
- Using fixed or optimised emission ratios (or delta values) for the secondary gases (or isotopes) from each sector.
- Using optimised boundary condition delta values for secondary isotopes.
- Using fixed or optimised model error values.
- Specifying different prior PDFs for all of the variables discussed above.
- Using real data or a pseudo-data setup, which produces synthetic observations from a known emissions field.
- Using either ACRG or OpenGHG code to produce the merged input datasets from netCDF files or an OpenGHG object store.
- Option to reload the merged input data from a `.pickle` file or rerun this process.

Running the model

The model was tested and developed using a conda environment. The model code repository should contain a conda environment file, listing all required packages. The `multiple_gas_inverse_model` repository should be installed into this conda environment.

The `.ini` file can be modified to specify which inputs and setup you want to use. As described in `run_multi_gas_model.py`, the model can then be run using the following line:

```
python run_multi_gas_model.py ’start_date’ ’end_date’ -c /path/to/.ini/file
```

I recommend running the model on the compute nodes of a HPC system if possible as, depending on the number of observations and MCMC iterations, this can be a memory and time intensive process.

The rest of this document runs through the model step by step: the required data formats, stacking and producing matrices and arrays in the correct shapes for the MCMC process, modelling both mole fraction and isotope observations, the MCMC process itself and how these outputs are processed into posteriors.

1 Setting data directories, checking array sizes and creating filenames

Directories for all input data files can be set manually (to allow for the use of, for example, the shared file area for footprints and local file area for novel emissions priors).

The code then fails out if certain variables or inputs are incorrect. For example, if a model error prior mean has not be specified for each gas and site, or if the country mask file doesn't exist. This saves time when testing new model inputs as the code will fail instantly, rather than at the end or middle of the run.

Unique merged data and post-MCMC output names are then created, based on the type of inputs and model run chosen in the `.ini` file.

2 Producing or reloading the merged data object and filtering observations

Data is supplied to the model as a dictionary of `xarray` datasets. Each dataset contains the observations, observational uncertainty, transport footprints and emissions and boundary conditions sensitivity matrices for each site. A priori emissions and boundary conditions are also included in each dictionary.

This dictionary of datasets can be produced in one of two ways: using functions from the ACRG code repository and netCDF files (as written in `data_functions.merge_fp_data_flux_bc()`); or by using functions in the OpenGHG `openghg_inversions` repository with an OpenGHG object store. Both of these processes produce the same `fp_data_H` object.

This process is repeated for each gas (the primary and all secondary gases) and these objects are stacked into a dictionary (with species as the keys) of dictionaries (with sites as their keys) of datasets.

This object can then be saved as a `.pickle` file, and read in again, rather than rerunning this process. This can speed things up when testing, for example, different model uncertainty setups, variable PDFs, filtering observations or MCMC convergence.

If using prior fluxes option:

For synthetic data tests, `flux_name_prior` can be used to specify different a priori flux fields, than those used to produce the synthetic observations (this process of producing the observations is discussed in more detail below). If this option is used, the merged data object is created again for each gas, this time using the prior flux fields.

Observations can then be filtered (removed from the datasets) using the filtering options given in `data_functions.filtering()`. This process removes the values of all variables at each filtered timestamp, hence why this process has to occur after the `fp_data_H` object is created.

3 Creating the MCMC inputs

Observations (and their corresponding measurement errors and timestamps), sensitivity matrices and basis function grids are extracted from these merged datasets and stacked into their own dictionaries based on gas, site and sector.

Other parameters, which allow for creation of the modelled observations and uncertainty matrices, are also created at this point. These include, for example, a dictionary of lists of the number of observations per site; a dictionary of arrays containing the site for each timestamp; and a dictionary of indexes used to expand the model error prior out from one value per site, to one value per observation.

Diagonal emissions and boundary condition uncertainty matrices are created, using squares of the emissions and boundary conditions prior uncertainties (`x_mu`, `xbc_mu`) to fill each diagonal.

Model-measurement uncertainty matrices are created from the root square mean of the a-priori model error and the measurement error at each timestamp. These values squared form the diagonal of the model-measurement uncertainty matrix for each site. There are options to include off-diagonal uncertainty terms in the model-measurement uncertainty matrices, such as the covariance between two concurrent observations or the time-correlated uncertainty. However, these have not been fully tested and would require changes to the MCMC process. The inverse of the uncertainty matrix has to be recalculated every time any uncertainty value is changed, which is a very slow process for large matrices containing off-diagonal terms. The current code setup assumes that there are no off-diagonal terms, so a quicker method for calculating the inverse can be used (discussed in more detail in Section 4).

Step sizes for each variable parameter are created. These values are hard-coded here, but can be modified in the code.

If using pseudo data option:

In the pseudo data tests, no background mole fractions are included, so the `use_bc` option should be set to `False`.

For mole fraction gases, the synthetic observations are taken directly from the datasets of merged footprints and fluxes (the emissions sensitivity matrix H). See Appendix A for more detail on this. Optional `x_true` scaling factors (other than 1.0) can be specified by sector here.

For delta value observations, the synthetic observations are created from the modelled primary gas mole fractions and the a-priori mean delta value for each sector. See Appendix B for more detail and the equations used in this process.

Random noise is added to each observation, by sampling from a Gaussian distribution with a mean of zero and a standard deviation equal to the chosen model error mean (`y_sig_mu`). There's also options to add systematic noise (where all synthetic observations are scaled or down by a value) or correlated noise (where noise from the primary gas is added to observations of the secondary gas at the same timestamp). However, these last two types of noise have not recently been tested with this version of the model.

If using delta value secondary gas(es):

For every delta value observation, there needs to be an observation of the primary gas at the same timestamp. As modelled delta values are created by taking the ratio of the mole fraction contributions of each isotopologue, a total mole fraction estimate is needed first. This is discussed in more detail in Appendix B.

Therefore, any delta value observations without a concurrent primary gas observation are filtered out at this point. Corresponding measurement errors, timestamps etc. are also filtered out.

If using prior fluxes option:

In this case, the emissions sensitivity matrix (H) for the rest of the model is taken from the merged dataset containing the a-priori fluxes, rather than the merged dataset containing the flux fields used to produce the synthetic observations.

4 Running the MCMC process

All the inputs discussed above are then fed into the MCMC function. Before looping through this process, empty arrays are formed to hold the MCMC trace of each varied parameter and for other values (such as the modelled mole fraction) which are output by the code at the end of the model run.

The inverse and log determinant of uncertainty matrices for the emissions, boundary conditions (when used), model-measurement error and model error uncertainty (when used) are calculated once before the MCMC loop, to reduce computational time. The inverse of these matrices is only recalculated during the MCMC loop when the uncertainty parameters change, for example when a model error uncertainty hyperparameter is used.

Figure 1 outlines the overall MCMC process for a generic variable. Each step in this process is discussed in more detail below, including a step by step description of how this differs for each type of variable.

During each iteration of the loop, a new set of values for each parameter is tested in turn, so all variables are 'optimised' together.

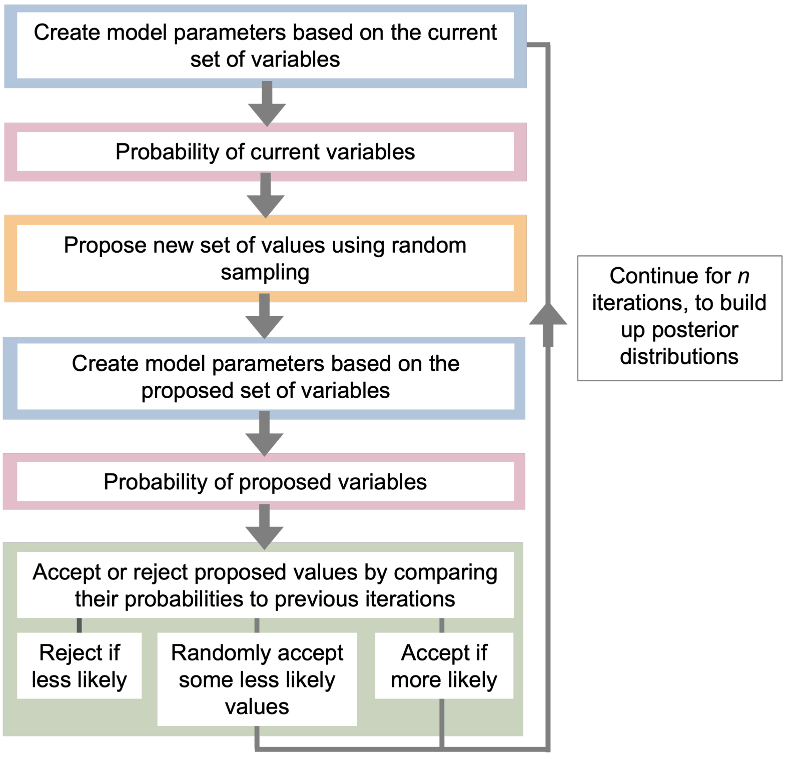
Create model parameters based on the current set of variables
Probability of current variables
Propose new set of values using random sampling
Create model parameters based on the proposed set of variables
Probability of proposed variables
Accept or reject proposed values by comparing their probabilities to previous iterations
Reject if less likely
Randomly accept some less likely values
Accept if more likely
Continue for *n* iterations, to build up posterior distributions

Figure 1: Flowchart showing a general outline of the MCMC process for a generic variable.

4.1 Emission scaling factors x

1. Current probability is found by taking the sum of the likelihood of all modelled mole fractions relative to the observed mode fractions (Equation 33 and the sum of the probability of the current emissions scaling factors relative to the prior emissions scaling factors (Equations 34, 35 or 36 depending on the shape of the emissions scaling factors prior PDF.
2. A set of proposed emissions scaling factors are generated using Equation 37.
3. Proposed modelled mole fraction or delta values of all gases are created using this set of proposed emissions scaling factors, using equations in Section A for the mole fraction gases or Section B for delta value gases.
4. The probability of the current and proposed emissions scaling factors is compared using Equation 38. Accepted values then become the 'current' emissions scaling factor values.
5. Current modelled mole fractions or delta values are recreated using the new set of 'current' emissions scaling factors, using equations as discussed in step 3.

4.2 Boundary condition scaling factors x_{bc}

The MCMC loop for this parameter is the same as that for the emissions scaling factors, except that the boundary conditions scaling factors are updated during step 2.

This loop is carried out for each mole fraction gas, as the boundary conditions for these gases are optimised directly. Boundary conditions for the delta values gases are optimised indirectly through the R_{bc} parameter, so are updated during the x_{bc} part of the loop.

4.3 Emission ratio or source signature scaling factors R

The MCMC loop for this parameter follows the same steps as the loop for the emissions scaling factors, other than the following changes during steps 3 and 5, before the current/proposed modelled mole fractions or delta values are updated:

- If single emission ratio/source signature scaling factors are being used, instead of spatial scaling factors on the same scale as the emissions scaling factors, these single scaling factors for each sector are extrapolated out to cover the whole study domain. These full arrays are required to produce modelled mole fractions or delta values.
- Following this, the sensitivity matrix corresponding to each mole fraction secondary gas is recreated using the full version of the emission ratio scaling factor arrays.

This loop is carried out in turn for each sector for each secondary gas.

4.4 Boundary condition source signature scaling factors R_{bc}

The MCMC loop for this parameter is only carried out for delta value secondary trace gases when boundary conditions are in use. This loops runs in the same way as that for the boundary condition scaling factors ($\mathbf{x}_{bc}$), except that only the likelihood of the modelled mole fractions of the primary gas and the current delta value secondary gas that is being updated/tested is included.

4.5 Model error uncertainty σ_y

Unlike previous variables, updating the model error for each gas only impacts the likelihood of that gas, so each is tested entirely separately. However, there are more computationally expensive steps involved in this MCMC loop:

1. Current probability is found by taking the sum of the likelihood of the modelled mole fractions relative to the observed mode fractions (Equation 33) and the sum of the probability of the current model error values relative to the prior model error values (Equations 34, 35 or 36 depending on the shape of the model error uncertainty prior PDF).
2. A set of proposed model error values are generated using Equation 37.
3. These values are extrapolated out (often from one value per site) to cover the whole array of observations for that gas.
4. The model-measurement uncertainty matrix for the gas is recreated using the new set of proposed model error values. The inverse and log determinant of this matrix is then recalculated (using Section C.5).
5. The probability of the current and proposed model error values is compared using Equation 38. Accepted values then become the 'current' model error values.
6. The 'current' model-measurement uncertainty matrix is recreated using the accepted model error values. And the inverse and log determinant of this matrix is recalculated.

This process is carried out for each gas separately.

4.6 Additional steps

At the end of each process described in Sections 4.1 to 4.5, the step size associated with each parameter is updated using Equation 40 to optimise the ratio of the number of times each proposed set of parameters is accepted or rejected.

To aid with convergence checking, the difference and percentage difference between the average of each value of each parameter along the MCMC trace is printed out to the command line. This is discussed in more detail in Section C.7.

5 Processing outputs

Once the MCMC process has been run, the traces of each variable are sub-sampled, to produce posterior distributions. First, the last 50% of values are retained (the percentage of values to keep can be changed using the `post_av` variable), effectively treating the first 50% of MCMC iterations as a 'burn-in' period. Then every 100^{th} value of the remaining trace is retained (this can be changed using the `n_trace_samples` variable, e.g. setting this to 50 would keep every 50^{th} value of the posterior trace). This step reduces the size of the posterior arrays (and therefore reduces the size of the output netCDF file) without impacting the posterior PDFs, if the traces have converged well.

Posterior means and 2.5 and 97.5 percentiles are then calculated, using the assumption that all posterior PDFs are Gaussian.

`produce_model_outputs` in `post_mcmc_functions.py` includes calculations for the posterior sector level fluxes, country flux totals and posterior emissions and emission ratio scaling factors, extrapolated out to the full spatial domain.

As emissions scaling factors are solved for using a basis function grid, total country fluxes are calculated per basis function cell, then summed together to produce totals in terms of teragrams per year.

The trace of modelled mole fractions and/or delta values was retained from the MCMC trace, so is also sub-sampled using the process described above, to produce posterior PDFs for the modelled observations. However, as modelled boundary condition mole fractions and delta values are not separated out during the MCMC process, posterior traces of modelled boundary condition mole fractions and delta values are recalculated here.

All output variables are stacked into a large `xarray` dataset, with variables indexed by species, sector and site where applicable. This dataset is saved as a netCDF file, in the specified `output_directory`.

A Modelled mole fraction values

Modelled mole fraction observations **y** are created using the forward model:

$$\mathbf{y} = \mathbf{H} \cdot \mathbf{x} + \epsilon_y \tag{1}$$

where **H** is the emissions sensitivity matrix containing the merged footprints and fluxes, **x** are the emissions scaling factors and ϵ_y is the model-measurement uncertainty.

For real data runs, where background (boundary condition) mole fractions are included, this forward model is expanded to include the boundary conditions sensitivity matrix ($\mathbf{H}_{bc}$) and boundary conditions scaling factors ($\mathbf{x}_{bc}$):

$$\mathbf{y} = \mathbf{H} \cdot \mathbf{x} + \mathbf{H}_{bc} \cdot \mathbf{x}_{bc} + \epsilon_y \tag{2}$$

As this model is built to estimate emissions from any number of sectors, each sector requires its own sensitivity matrix and scaling factors. The forward model for producing modelled mole fraction observations from multiple sectors (n) can be shown as:

$$\mathbf{y} = \begin{bmatrix} \mathbf{H}_1 & \dots & \mathbf{H}_n & \mathbf{H}_{bc} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_n \\ \mathbf{x}_{bc} \end{bmatrix} + \epsilon_y \tag{3}$$

Modelled mole fractions for secondary gases ($\mathbf{y}_S$) can be produced, relative to emissions (and emissions scaling factors) of the primary gas by using the secondary gas's emission ratios **R** for each sector:

$$\mathbf{y}_S = \begin{bmatrix} \mathbf{H}_1 & \dots & \mathbf{H}_n & \mathbf{H}_{bc,S} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_1\mathbf{R}_1 \\ \vdots \\ \mathbf{x}_n\mathbf{R}_n \\ \mathbf{x}_{bc,S} \end{bmatrix} + \epsilon_{y,S} \tag{4}$$

B Modelled delta values

Currently, the code only contains functions to produce modelled delta values for methane, with the assumption that $^{12}CH_4$, $^{13}CH_4$ and $^{12}CH_3D$ are the only isotopologues present. In general, this method takes a methane mole fraction measurement and a sample of delta values and uses these to calculate the mole fraction of each of methane's isotopologues (see Section B.1 for the derivation of these equations). The ratios of these isotopologues are then converted into a modelled delta value. So, any time a new set of modelled methane mole fractions is created (when new emissions scaling factors are sampled) or when a new set of delta values is sampled, the impact on modelled delta values can be calculated. This method runs as follows:

1. Convert sampled emissions source signatures and boundary condition source signatures ($\delta_{i,source}$) into sampled isotopic ratios ($\mathrm{R}_{i,source}$) using:

$$R_{i,source} = R_{i,std}\left(\frac{\delta_{i,source}}{1000} + 1\right) \tag{5}$$

where i is the isotopologue, *source* is the emissions source or boundary condition and $R_{i,std}$ is the isotopic standard. For ^{13}CH$_4$, $R_{13,std}$ is the Vienna PeeDee Belemnite (VPDB) standard with a value of 0.0112372 and for ^{12}CH$_3$D, $R_{2,std}$ Vienna Standard Mean Ocean Water (VSMOW) with a value of 0.00015575.

2. Calculate the absolute fraction of each isotopologue ($^{12}\mathbf{f}_{source}$, $^{13}\mathbf{f}_{source}$ and $^{2}\mathbf{f}_{source}$) using the sampled isotopic ratios and Equations 17, 19 and 21 below.
3. Produce modelled mole fractions of each isotopologue using the forward model for mole fractions (Equation 3) and the absolute fractions for each isotopologue. For example, for the modelled mole fraction of ^{13}CH$_4$ from two sectors (FF and nonFF):

$$\mathbf{y}_{^{13}CH_4} = \begin{bmatrix}\mathbf{H}_{FF} & \mathbf{H}_{nonFF} & \mathbf{H}_{bc,CH_4}\end{bmatrix} \cdot \begin{bmatrix} ^{13}\mathbf{f}_{FF}\mathbf{x}_{FF} \\ ^{13}\mathbf{f}_{nonFF}\mathbf{x}_{nonFF} \\ ^{13}\mathbf{f}_{bc,CH_4}\mathbf{x}_{bc,CH_4} \end{bmatrix} \tag{6}$$

4. Calculate the isotopic ratio from the ratio of the isotopologue mole fractions:

$$R_{^{13}CH_4} = \frac{\mathbf{y}_{^{13}CH_4}}{\mathbf{y}_{^{12}CH_4}} \tag{7}$$

$$R_{^{2}CH_3D} = \frac{\mathbf{y}_{^{2}CH_4}}{\mathbf{y}_{^{12}CH_4}} \tag{8}$$

5. Then convert this ratio into a delta values using the isotopic delta value notation:

$$\delta_i = \left(\frac{R_i}{R_{i,std}} - 1\right) \times 1000 \tag{9}$$

6. These modelled delta values are based on the typically more-frequency primary gas observations, so need to be resampled down, to remove any timestamps without concurrent delta value observations.

B.1 Isotopic concentration

This method assumes that only three methane isotopologues are present in the atmosphere.

The total methane mole fraction can be found by summing the mole fraction contribution from each isotopologue:

$$CH_4 = {}^{12}\mathrm{CH}_4 + {}^{13}\mathrm{CH}_4 + {}^{12}\mathrm{CH}_3\mathrm{D} \tag{10}$$

$$\delta^{13} = \left(\frac{R_{13}}{R_{std_{13}}} - 1\right) \times 1000 \tag{11}$$

Where:

$$R_{13} = \frac{^{13}CH_4}{^{12}CH_4} \tag{12}$$

and $R_{std_{13}}$ is the Vienna PeeDee Belemnite (VPDB) standard with a value of 0.0112372.

$$\delta^2 = \left(\frac{R_2}{R_{std_2}} - 1 \right) \times 1000 \tag{13}$$

Where:

$$R_2 = \frac{^{12}CH_3D}{^{12}CH_4} \tag{14}$$

and R_{std_2} is the Vienna Standard Mean Ocean Water (VSMOW) with a value of 0.00015575.

The concentration of ^{12}CH$_4$ can be found as follows. First, Equations 12 and 14 are rearranged, then Equations 11 and 13 are substituted in, to give these isotopologue concentrations in terms of delta values:

$$\begin{aligned}
^{13}CH_4 &= {}^{12}\text{CH}_4 R_{13} \\
^{12}CH_3D &= {}^{12}\text{CH}_4 R_2 \\
^{13}CH_4 &= {}^{12}\text{CH}_4 R_{std_{13}} \left(\frac{\delta^{13}}{1000} + 1 \right) \\
^{12}CH_3D &= {}^{12}\text{CH}_4 R_{std_2} \left(\frac{\delta^{2}}{1000} + 1 \right)
\end{aligned} \tag{15}$$

Then these equations are substituted in to Equation 10 and rearranged:

$$\begin{aligned}
CH_4 &= {}^{12}\text{CH}_4 + {}^{12}\text{CH}_4 R_{std_{13}} \left(\frac{\delta^{13}}{1000} + 1 \right) + {}^{12}\text{CH}_4 R_{std_2} \left(\frac{\delta^{2}}{1000} + 1 \right) \\
CH_4 &= {}^{12}\text{CH}_4 \left[1 + R_{std_{13}} \left(\frac{\delta^{13}}{1000} + 1 \right) + R_{std_2} \left(\frac{\delta^{2}}{1000} + 1 \right) \right]
\end{aligned} \tag{16}$$

$$^{12}CH_4 = \frac{CH_4}{\left[1 + R_{std_{13}} \left(\frac{\delta^{13}}{1000} + 1 \right) + R_{std_2} \left(\frac{\delta^{2}}{1000} + 1 \right) \right]} \tag{17}$$

Similarly, the concentration of ^{13}CH$_4$ can be found:

$$\begin{aligned}
{}^{12}CH_4 &= \frac{{}^{13}CH_4}{R_{std_{13}}\left(\frac{\delta^{13}}{1000}+1\right)} \\
{}^{12}CH_3D &= {}^{12}\mathrm{CH_4}R_{std_2}\left(\frac{\delta^{2}}{1000}+1\right) \\
CH_4 &= {}^{12}\mathrm{CH_4} + {}^{13}\mathrm{CH_4} + {}^{12}\mathrm{CH_3D} \\
CH_4 &= \frac{{}^{13}CH_4}{R_{std_{13}}\left(\frac{\delta^{13}}{1000}+1\right)} + {}^{13}\mathrm{CH_4} + {}^{12}\mathrm{CH_4}R_{std_2}\left(\frac{\delta^{2}}{1000}+1\right) \\
CH_4 &= \frac{{}^{13}CH_4}{R_{std_{13}}\left(\frac{\delta^{13}}{1000}+1\right)} + {}^{13}\mathrm{CH_4} + \frac{{}^{13}CH_4}{R_{std_{13}}\left(\frac{\delta^{13}}{1000}+1\right)}R_{std_2}\left(\frac{\delta^{2}}{1000}+1\right) \\
CH_4 &= {}^{13}\mathrm{CH_4}\left[\frac{1}{R_{std_{13}}\left(\frac{\delta^{13}}{1000}+1\right)} + 1 + \frac{R_{std_2}\left(\frac{\delta^{2}}{1000}+1\right)}{R_{std_{13}}\left(\frac{\delta^{13}}{1000}+1\right)}\right]
\end{aligned} \tag{18}$$

$${}^{13}CH_4 = \frac{CH_4}{\left[\frac{1}{R_{std_{13}}\left(\frac{\delta^{13}}{1000}+1\right)} + 1 + \frac{R_{std_2}\left(\frac{\delta^{2}}{1000}+1\right)}{R_{std_{13}}\left(\frac{\delta^{13}}{1000}+1\right)}\right]} \tag{19}$$

The concentration of ${}^{12}\mathrm{CH_3D}$ can be found:

$$\begin{aligned}
{}^{13}CH_4 &= {}^{12}\mathrm{CH_4}R_{std_{13}}\left(\frac{\delta^{13}}{1000}+1\right) \\
{}^{12}CH_4 &= \frac{{}^{12}CH_3D}{R_{std_2}\left(\frac{\delta^{2}}{1000}+1\right)} \\
CH_4 &= {}^{12}\mathrm{CH_4} + {}^{13}\mathrm{CH_4} + {}^{12}\mathrm{CH_3D} \\
CH_4 &= \frac{{}^{12}CH_3D}{R_{std_2}\left(\frac{\delta^{2}}{1000}+1\right)} + {}^{12}\mathrm{CH_4}R_{std_{13}}\left(\frac{\delta^{13}}{1000}+1\right) + {}^{12}\mathrm{CH_3D} \\
CH_4 &= \frac{{}^{12}CH_3D}{R_{std_2}\left(\frac{\delta^{2}}{1000}+1\right)} + \frac{{}^{12}CH_3D}{R_{std_2}\left(\frac{\delta^{2}}{1000}+1\right)}R_{std_{13}}\left(\frac{\delta^{13}}{1000}+1\right) + {}^{12}\mathrm{CH_3D} \\
CH_4 &= {}^{12}\mathrm{CH_3D}\left[\frac{1}{R_{std_2}\left(\frac{\delta^{2}}{1000}+1\right)} + \frac{R_{std_{13}}\left(\frac{\delta^{13}}{1000}+1\right)}{R_{std_2}\left(\frac{\delta^{2}}{1000}+1\right)} + 1\right]
\end{aligned} \tag{20}$$

$${}^{12}CH_3D = \frac{CH_4}{\left[\frac{1}{R_{std_2}\left(\frac{\delta^{2}}{1000}+1\right)} + \frac{R_{std_{13}}\left(\frac{\delta^{13}}{1000}+1\right)}{R_{std_2}\left(\frac{\delta^{2}}{1000}+1\right)} + 1\right]} \tag{21}$$

B.2 Isotopic concentration - using the method from Griffith, 2018

This method makes no assumption about the isotopologues present in the atmosphere and instead finds the isotopic concentration from the relative abundance of each molecule in each isotopologue.

The isotope ratio defines the proportion of minor to major isotope:

$$R_{13} = \frac{^{13}C}{^{12}C} \tag{22}$$

$$R_2 = \frac{^{2}H}{^{1}H} \tag{23}$$

These isotope ratios can be expressed as delta values, relative to a reference or standard material (R_{std}). To simplify these equations, the factor of 1000 for converting values to ‰is not included here. Any observations will have to be scaled to account for this.

$$\delta^{13} = \left(\frac{R_{13}}{R_{std_{13}}} - 1\right) \tag{24}$$

$$\delta^{2} = \left(\frac{R_{2}}{R_{std_{2}}} - 1\right) \tag{25}$$

and $R_{std_{13}}$ is the Vienna PeeDee Belemnite (VPDB) standard with a value of 0.0112372 and R_{std_2} is the Vienna Standard Mean Ocean Water (VSMOW) with a value of 0.00015575.

Therefore:

$$R_{13} = R_{std_{13}}\left(\delta^{13} + 1\right) \tag{26}$$

$$R_{2} = R_{std_{2}}\left(\delta^{2} + 1\right) \tag{27}$$

Isotopic abundance, the fraction of that isotope relative to all isotopes in a sample, can be expressed as follows:

$$\begin{aligned}
{}^{12}a &= \frac{^{12}\mathrm{C}}{^{12}\mathrm{C} + {}^{13}\mathrm{C}} \\
{}^{13}a &= \frac{^{13}\mathrm{C}}{^{12}\mathrm{C} + {}^{13}\mathrm{C}} \\
{}^{1}a &= \frac{^{1}\mathrm{H}}{^{1}\mathrm{H} + {}^{2}\mathrm{H}} \\
{}^{2}a &= \frac{^{2}\mathrm{H}}{^{1}\mathrm{H} + {}^{2}\mathrm{H}}
\end{aligned} \tag{28}$$

By rearranging Equations 22 and 23 and substituting these into Equations 28, these isotopic abundances can be represented as:

$$
\begin{aligned}
{}^{12}a &= \frac{{}^{12}\mathrm{C}}{{}^{12}\mathrm{C} + {}^{13}\mathrm{C}} = \frac{{}^{12}\mathrm{C}}{{}^{12}\mathrm{C} + {}^{12}\mathrm{C}R_{13}} = \frac{{}^{12}\mathrm{C}}{{}^{12}\mathrm{C}\,(1+R_{13})} = \frac{1}{1+R_{13}} \\
{}^{13}a &= \frac{{}^{13}\mathrm{C}}{{}^{12}\mathrm{C} + {}^{13}\mathrm{C}} = \frac{{}^{12}\mathrm{C}R_{13}}{{}^{12}\mathrm{C} + {}^{12}\mathrm{C}R_{13}} = \frac{{}^{13}\mathrm{C}R_{13}}{{}^{12}\mathrm{C}\,(1+R_{13})} = \frac{R_{13}}{1+R_{13}} \\
{}^{1}a &= \frac{{}^{1}\mathrm{H}}{{}^{1}\mathrm{H} + {}^{2}\mathrm{H}} = \frac{{}^{1}\mathrm{H}}{{}^{1}\mathrm{H} + {}^{1}\mathrm{H}R_{2}} = \frac{{}^{1}\mathrm{H}}{{}^{1}\mathrm{H}\,(1+R_{2})} = \frac{1}{1+R_{2}} \\
{}^{2}a &= \frac{{}^{2}}{{}^{1}\mathrm{H} + {}^{2}\mathrm{H}} = \frac{{}^{1}HR_{2}}{{}^{1}\mathrm{H} + {}^{1}\mathrm{H}R_{2}} = \frac{{}^{1}\mathrm{H}R_{2}}{{}^{1}\mathrm{H}\,(1+R_{2})} = \frac{R_{2}}{1+R_{2}}
\end{aligned}
\tag{29}
$$

Isotopic abundance for each isotopologue can then be found by taking the product of the abundance for each isotope found in the molecule:

$$
\begin{aligned}
a_{^{12}CH_4} &= {}^{12}\mathrm{a} \cdot ({}^{1}\mathrm{a})^{4} = \frac{1}{1+R_{13}} \cdot \left(\frac{1}{1+R_{2}}\right)^{4} = \frac{1}{(1+R_{13})\,(1+R_{2})^{4}} \\
a_{^{13}CH_4} &= {}^{13}\mathrm{a} \cdot ({}^{1}\mathrm{a})^{4} = \frac{R_{13}}{1+R_{13}} \cdot \left(\frac{1}{1+R_{2}}\right)^{4} = \frac{R_{13}}{(1+R_{13})\,(1+R_{2})^{4}} \\
a_{^{12}CH_3D} &= 4 \cdot {}^{12}\mathrm{a} \cdot ({}^{1}\mathrm{a})^{3} \cdot {}^{2}\mathrm{a} = 4 \cdot \frac{1}{1+R_{13}} \cdot \left(\frac{1}{1+R_{2}}\right)^{3} \cdot \frac{R_{2}}{1+R_{2}} = \frac{4R_{2}}{(1+R_{13})\,(1+R_{2})^{4}}
\end{aligned}
\tag{30}
$$

The isotopic abundance for ^{12}CH$_3$D has a factor of 4 because there are 4 (presumably equivalent to the spectrometer?) ways a hydrogen atom (^{1}H) could be substituted for a deuterium atom (^{2}H).

Equations 30 can then be used with Equations 24 and 25 to express these equations in terms of delta value. The mole fraction of each isotopologue can be found by multiplying the total methane mole fraction by each isotopic abundance.

C MCMC process equations

C.1 Likelihood

For the likelihood (the probability of modelled mole fractions or delta values, given the observed data), we assume a multivariate Gaussian distribution:

$$
\rho(\mathbf{y}|\mathbf{x}) = \frac{exp(-\frac{1}{2}(\mathbf{y} - \mathbf{y}_{mod})^{T}\mathbf{Q}^{-1}(\mathbf{y} - \mathbf{y}_{mod}))}{\sqrt{(2\pi)^{k}|\mathbf{Q}|}}
\tag{31}
$$

As the MCMC process involves comparing two sets of these likelihoods (the proposed and current likelihood), scaling terms can be removed to simplify this. Also, the logarithm of the equation is used, to reduce the use of small numbers and simplify computation:

$$
\rho(\mathbf{y}|\mathbf{x}) = \frac{-\frac{1}{2}(\mathbf{y} - \mathbf{y}_{mod})^{T}\mathbf{Q}^{-1}(\mathbf{y} - \mathbf{y}_{mod})}{ln(\sqrt{|\mathbf{Q}|})}
\tag{32}
$$

$$\rho(\mathbf{y}|\mathbf{x}) = \frac{1}{2}ln(|\mathbf{Q}|) - \frac{1}{2}(\mathbf{y} - \mathbf{y}_{mod})^T\mathbf{Q}^{-1}(\mathbf{y} - \mathbf{y}_{mod}) \tag{33}$$

C.2 Prior probabilities

There are three options for the prior PDF of each variable:

C.2.1 Gaussian

This equation has the same form as that used for the log-likelihood (Equation 33):

$$\rho(\mathbf{x}) = \frac{1}{2}ln(|\mathbf{P}|) - \frac{1}{2}(\mathbf{x} - \mathbf{x}_{prior})^T\mathbf{P}^{-1}(\mathbf{x} - \mathbf{x}_{prior}) \tag{34}$$

C.2.2 Truncated Gaussian

This has the same form as the Gaussian probability above (Equation 34), except when the proposed values are beyond the min/max limits:

$$\rho(\mathbf{x}) = \begin{cases} \frac{1}{2}ln(|\mathbf{P}|) - \frac{1}{2}(\mathbf{x} - \mathbf{x}_{prior})^T\mathbf{P}^{-1}(\mathbf{x} - \mathbf{x}_{prior}), & \text{if } x_{min} < \mathbf{x} < x_{max} \\ -\infty, & \text{otherwise} \end{cases} \tag{35}$$

C.2.3 Uniform

Similarly to the truncated Gaussian distribution, if proposed values are beyond the min/max limits, a -∞ probability is assigned, to ensure that this set of values is rejected:

$$\rho(\mathbf{x}) = \begin{cases} \frac{1}{x_{max} - x_{max}}, & \text{if } x_{min} < \mathbf{x} < x_{max} \\ -\infty, & \text{otherwise} \end{cases} \tag{36}$$

C.3 Sampling proposed values

Proposed values of each parameter are updated by taking random samples from a Gaussian distribution with a mean equal to zero and a standard deviation equal to the step size. The step size is hard-coded in for each type of parameter in `data_functions.py`.

$$x_{new} = x + \mathcal{N}(0, ss_x) \tag{37}$$

where ss_x is the step size of variable x.

C.4 Comparing probabilities and acceptance ratio

The process of accepting or rejecting proposed variable values always accepts the proposed values if they are more likely that the previous set of values. Some less likely sets of values are also randomly accepted, to ensure that the MCMC trace does not get stuck in a local minimum or maximum. The following equations are used to determine whether or not to accept the proposed set of values:

$$
\begin{aligned}
d &= \rho_{proposed} - \rho_{current} \\
u &= U(0,1) \\
f(d) &= \begin{cases} \text{accept}, & \text{if } ln(u) < d \\ \text{reject}, & \text{if } ln(u) > d \end{cases}
\end{aligned}
\tag{38}
$$

where ρ is the probability of the proposed or current set of values.

C.5 Matrix inverse and log determinant

Currently, all uncertainty matrices are assumed to be diagonal, so the calculations for the inverse and log determinant of the matrices can be simplified:

The inverse of a diagonal matrix can be found just by taking the inverse of each element of the diagonal.

The log determinant of a diagonal matrix can be found by calculating the sum of the natural logarithm of the diagonal of the matrix:

$$ln(|\mathbf{Q}|) = \sum ln(\mathbf{Q}_{diagonal}) \tag{39}$$

To use uncertainty matrices with off-diagonal terms, the inverse and determinant of the matrix would need to be calculated directly. This can be a slow process for large matrices, so standard (numpy.inv) is likely to be too slow. I've tested out other methods for this (e.g. using a Cholesky decomposition or solving for the matrix rather than inverting it) but I haven't found a viable solution yet.

C.6 Updating step size

Step size is optimised with the aim to produce an acceptance ratio of 0.3. The acceptance ratio is the number of sets of proposed values that are accepted, relative to the number of sets that are rejected. Changing the step size changes how far the proposed set of values deviates from the previous set of values, which could therefore impact the probability of these values being accepted.

Step size optimisation is performed after every MCMC sample, using Algorithm 4 from Andrieu and Thoms, 2008, adapted by Luke Western:

$$
\begin{aligned}
\Delta &= c_1(i - c_2)\left(\frac{a}{i} - r_{target}\right) \\
ss_{new} &= e^{ln(ss)+\Delta}
\end{aligned}
\tag{40}
$$

where ss_{new} is the updated step size, ss is the current step size, c_1 is a constant of value 1.0, c_2 is a constant of value 0.8, a is the total number of accepted set of values so far, r_{target} is the target acceptance ratio and i is the current number of iterations.

C.7 Convergence print out

This process is carried out twice during the whole MCMC trace, to print out information to help with convergence checking. The mean value of each variable in the set is found for the last 40-20% of iterations and for the last 20% of iterations. The absolute and percentage difference between these two sets of means is printed out. The following process is used to produce this output:

$$\begin{aligned} i_{min} &= i - 0.4 i_{total} \\ i_{max} &= i - 0.2 i_{total} \\ \text{trace}_1 &= \text{trace}[i_{min} : i_{max}, :] \\ \text{trace}_2 &= \text{trace}[i_{max} :, :] \end{aligned} \tag{41}$$

where i is the current iteration number, i_{total} is the total number of iterations. The means of $trace_1$ and $trace_2$ are then found, followed by the absolute and percentage differences between the two sets of means.

Bibliography

Andrieu, Christophe and Johannes Thoms (2008). “A tutorial on adaptive MCMC”. In: *Statistics and Computing* 18.4, pp. 343–373. DOI: 10.1007/s11222-008-9110-y.