MultilayerPy: a python package for creating and optimising multi-layer models of aerosol and film processes

Adam Milsom,[a] Amy Lees,[a] Adam M. Squires[b] and Christian Pfrang[a,c]

[a]Geography, Earth and Environmental Sciences, University of Birmingham, Birmingham, B15 2TT, UK
[b]Department of Chemistry, University of Bath, Bath, BA2 7AX, UK
[c]Department of Meteorology, University of Reading, Reading, RG6 6BB, UK

E-mail: a.milsom.2@bham.ac.uk | Twitter: @adam_milsom94

Abstract

Kinetic multi-layer models of aerosols and films have become the state-of-the-art method of describing complex aerosol processes at the particle and film level. We present MultilayerPy:[1] an open-source framework for building, running and optimising kinetic multi-layer models – namely the kinetic multi-layer model of aerosol surface and bulk chemistry (KM-SUB) and the kinetic multi-layer model of gas–particle interactions in aerosols and clouds (KM-GAP). MultilayerPy abstracts the model building process into separate building blocks, increasing the reproducibility of results and minimising human error.

(1) Introduction to kinetic multi-layer models

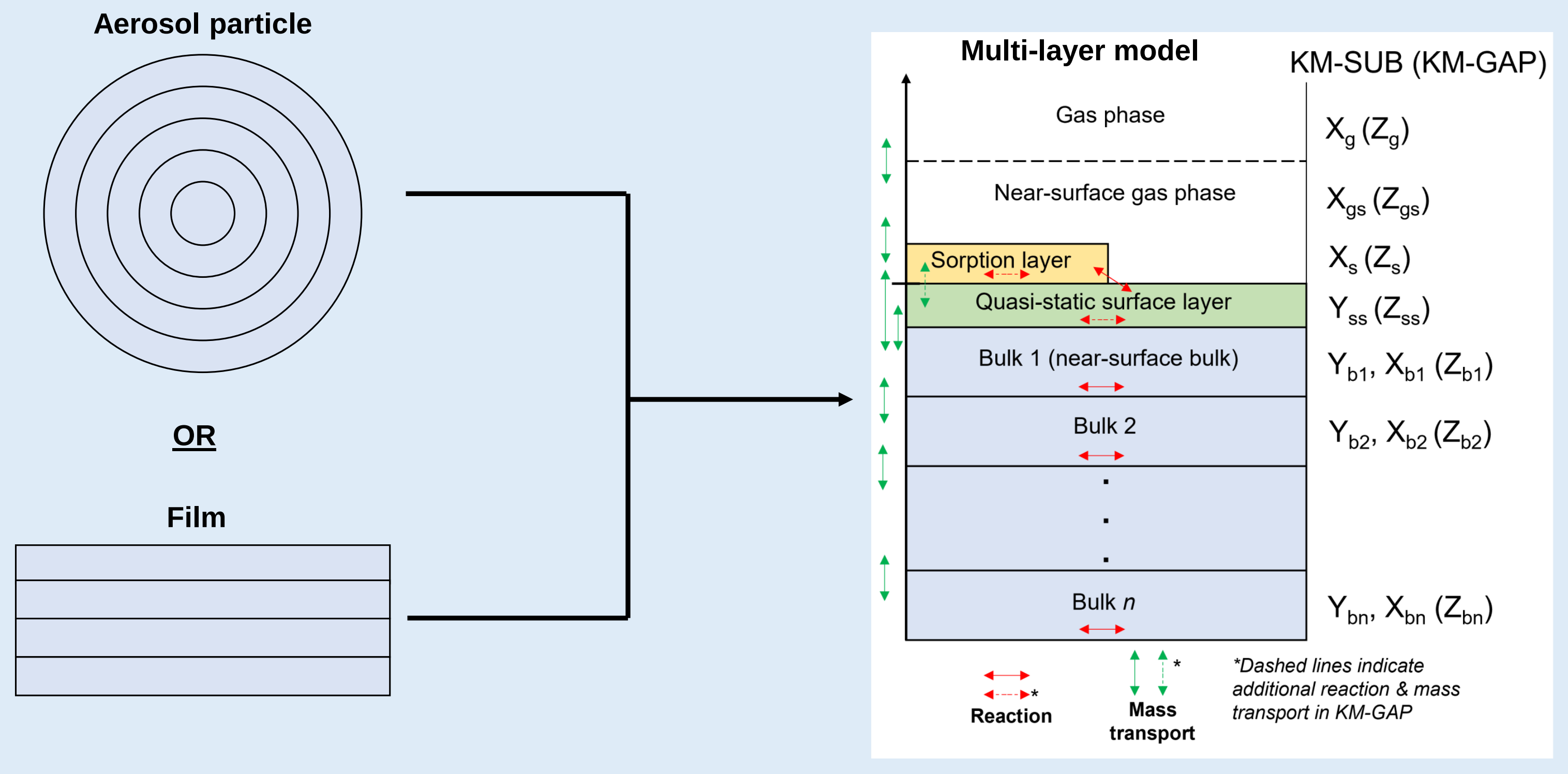
Aerosol particle
OR
Film
Multi-layer model
KM-SUB (KM-GAP)
Gas phase
Near-surface gas phase
Sorption layer
Quasi-static surface layer
Bulk 1 (near-surface bulk)
Bulk 2
Bulk n
X_g (Z_g)
X_{gs} (Z_{gs})
X_s (Z_s)
Y_{ss} (Z_{ss})
Y_{b1}, X_{b1} (Z_{b1})
Y_{b2}, X_{b2} (Z_{b2})
Y_{bn}, X_{bn} (Z_{bn})
Reaction
Mass transport
*Dashed lines indicate additional reaction & mass transport in KM-GAP

Figure 1. A schematic of kinetic multi-layer models of aerosol and film processes.

Kinetic multi-layer models split an aerosol particle or film into a number of bulk layers and resolve chemical reactions within and mass transport between each layer.

Key features include:

- Resolution of surface and bulk phase processes.
- Consideration of bulk phase viscosity.
- Particle size changes.
- Calculation of uptake coefficients.
- Resolution of concentration gradients within the bulk phase.

Two of the main kinetic multi-layer models are the kinetic multi-layer model of aerosol surface and bulk chemistry (**KM-SUB**)[2] and gas-particle interactions (**KM-GAP**).[3] Writing these models manually is time-consuming and error prone. **There is a need for a tool to facilitate the creation and optimisation of these models.**

(2) MultilayerPy

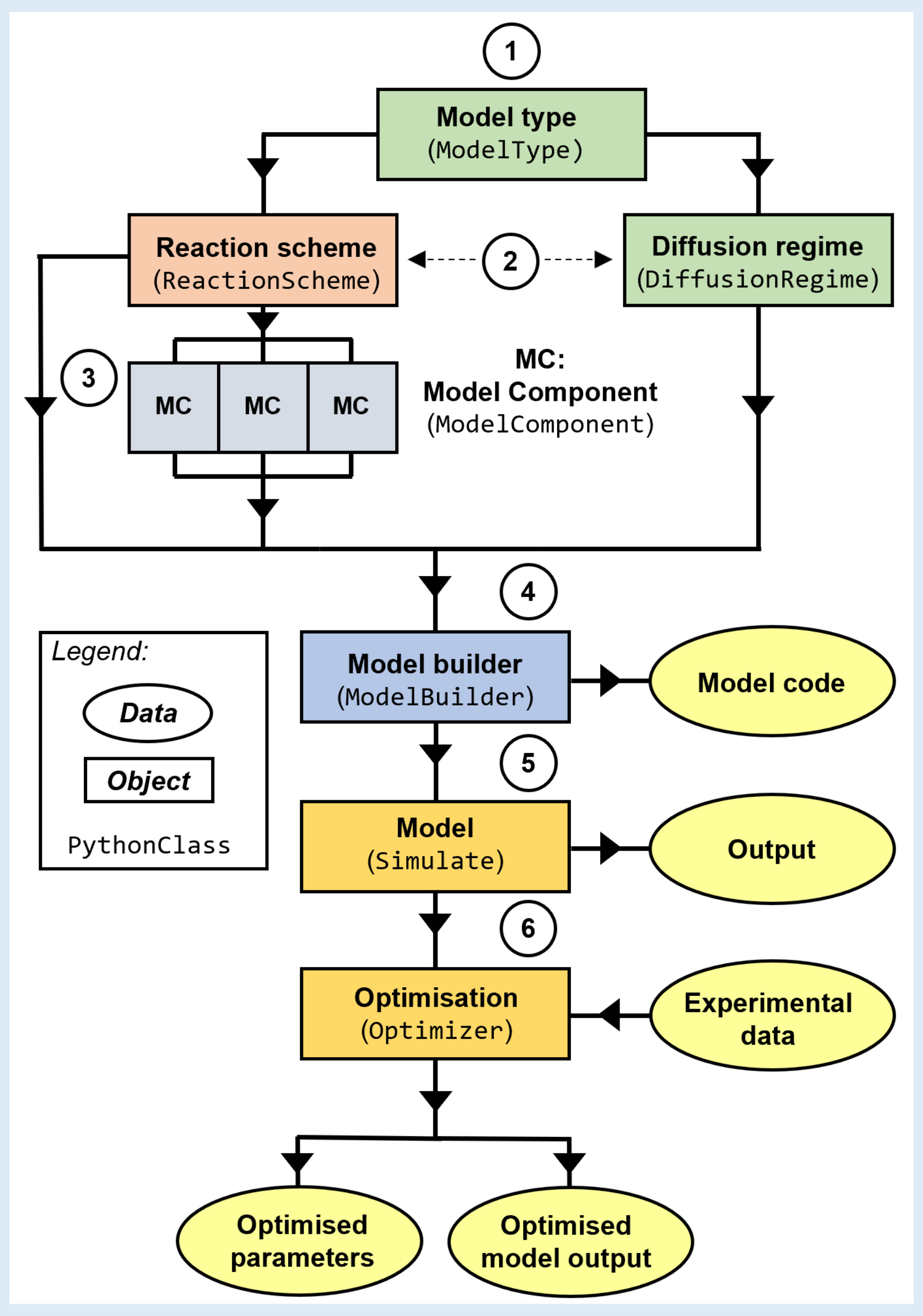

Figure 2.[1] A schematic of MultilayerPy showing its modularity and the key steps in the model building process.

MultilayerPy[1] is a Python package made to facilitate the creation, running and optimisation of kinetic multi-layer models.
The key features include:

- **Modularity** – the model building process is split into chunks so that the user can iterate through different models with ease.
- **Reproducibility** – the model output and code are generated in a readable manner, encouraging the user to share their code with e.g. a publication.
- **Open-source** – the package is released under an open-source license and collaboration on the project is encouraged.
- **Scalability** – it is possible to parallelise MultulayerPy model optimisation algorithms over many computer cores (e.g. on a supercomputer).

(3) Estimating parameter uncertainty

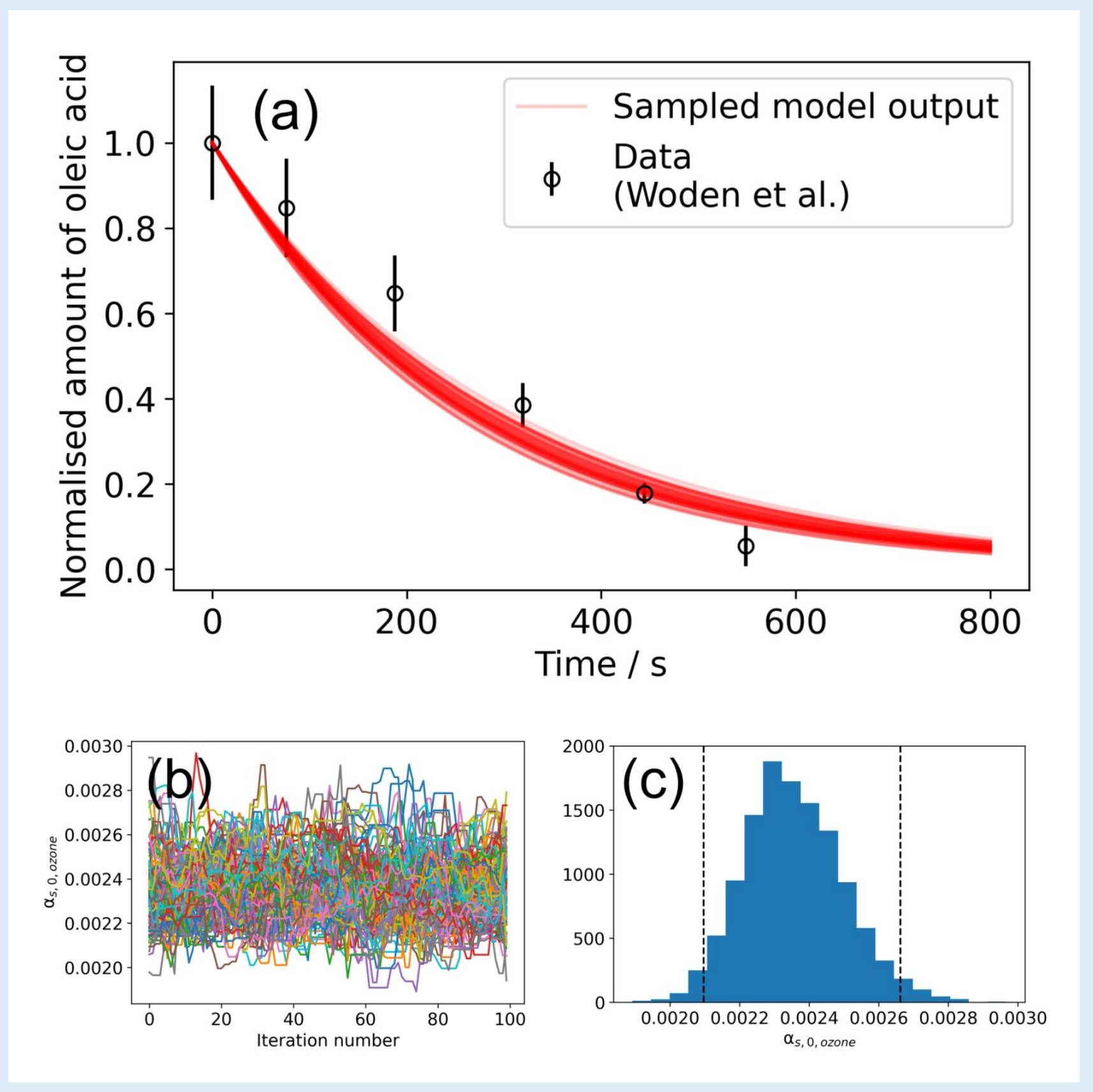

Figure 3.[1] The MCMC sampling process for one varying parameter. (a) a set of MCMC samples consistent with the data. (b) The chains of values for each walker at each MCMC algorithm iteration. (c) The distribution of parameter values derived from the converged walkers.

Fitting parameter uncertainty is determined using Markov Chain Monte Carlo (MCMC) sampling.

This involves setting a number of “walkers” on a random walk in the parameter space.

The likelihood of the next step in the walk is determined by how well the model fits to the data at the current position.

The “chain” of positions converges around the region of best fit and a probability distribution is obtained for the varying parameters (Fig. 3(c)).[1]

(4) Concentration gradients

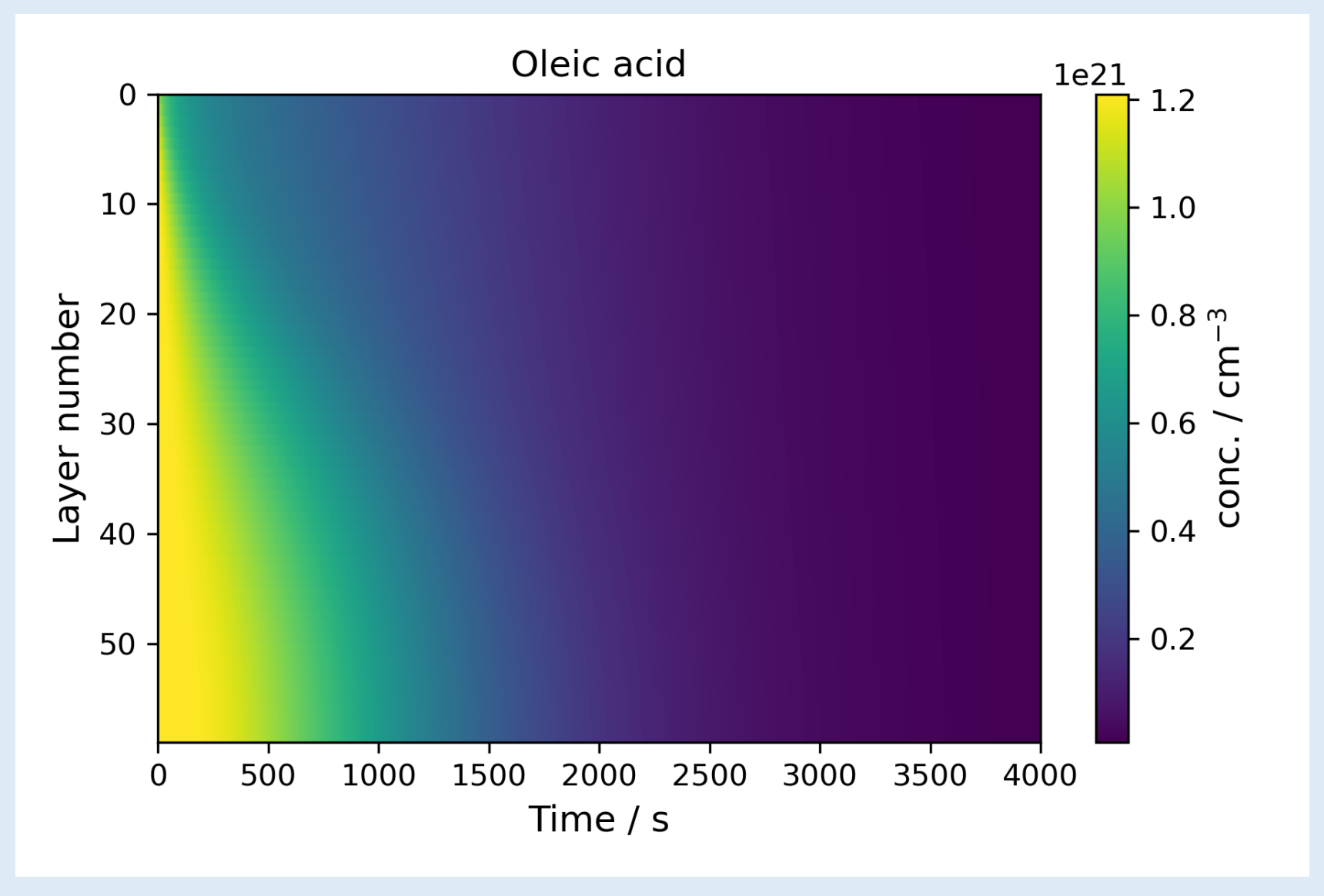

Figure 4. A heatmap plot showing the depth- and time-resolved oleic acid concentration in a film exposed to ozone.

Bulk concentration gradients can be resolved by the models constructed by MultilayerPy.

(5) Use case: films of real atmospheric material

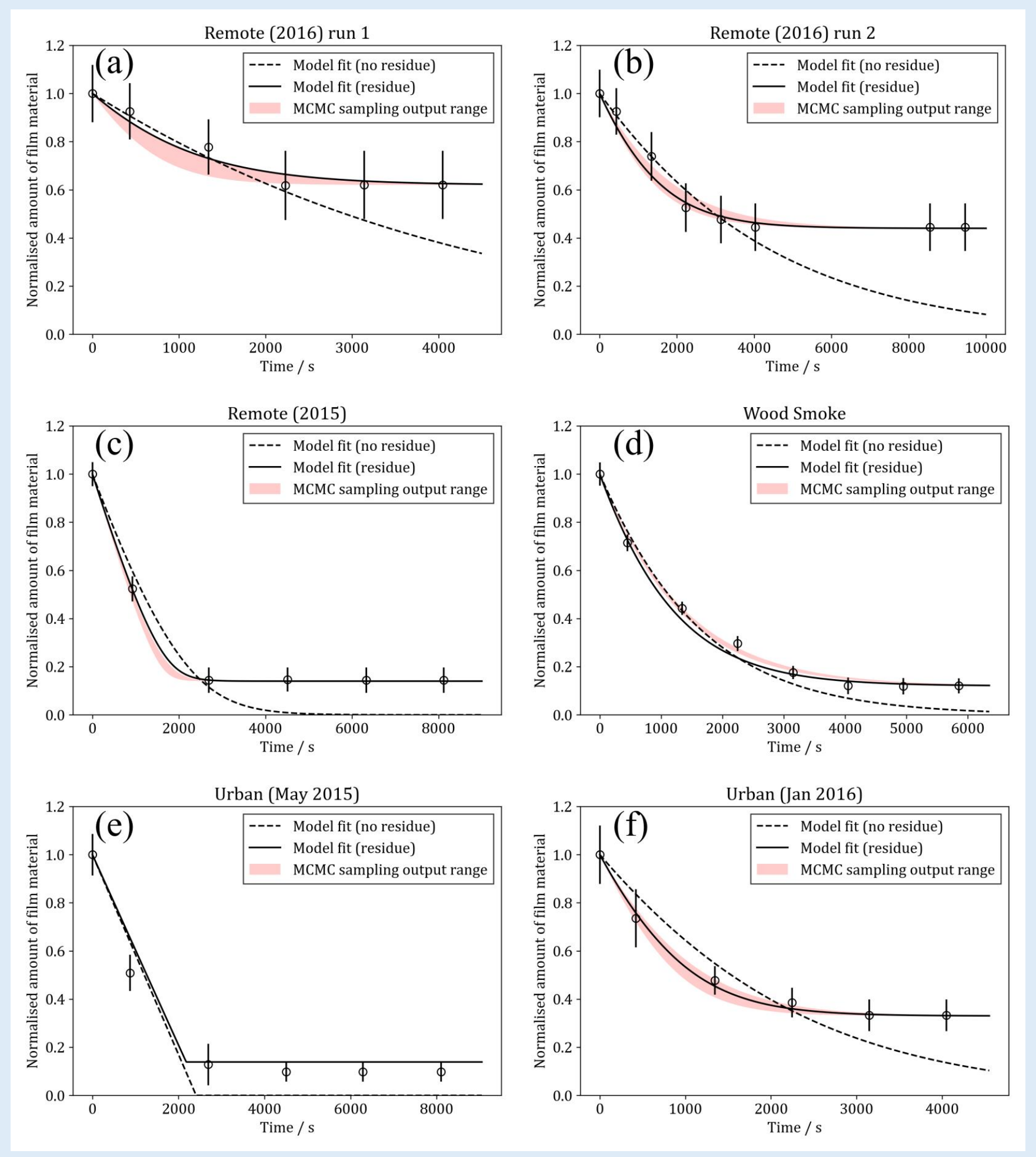

Figure 5. Model-data fits from a kinetic model created in MultilayerPy. Each plot contains kinetic data from real films deposited on water reacting with OH radicals.[4]

MultilayerPy was used to create a simple kinetic model for films of real atmospheric material reacting with OH radicals (Fig. 5).[4]

Two models were tested:
(i) Without an unreacted residue.
(ii) With an unreacted residue.

We found that the model with a residue fit better to the data.

These optimised models were used to calculate the chemical half-life over a range of [OH] concentrations.

The lifetime of these organic films can range from minutes to days. The reactivity varies between samples from different environments

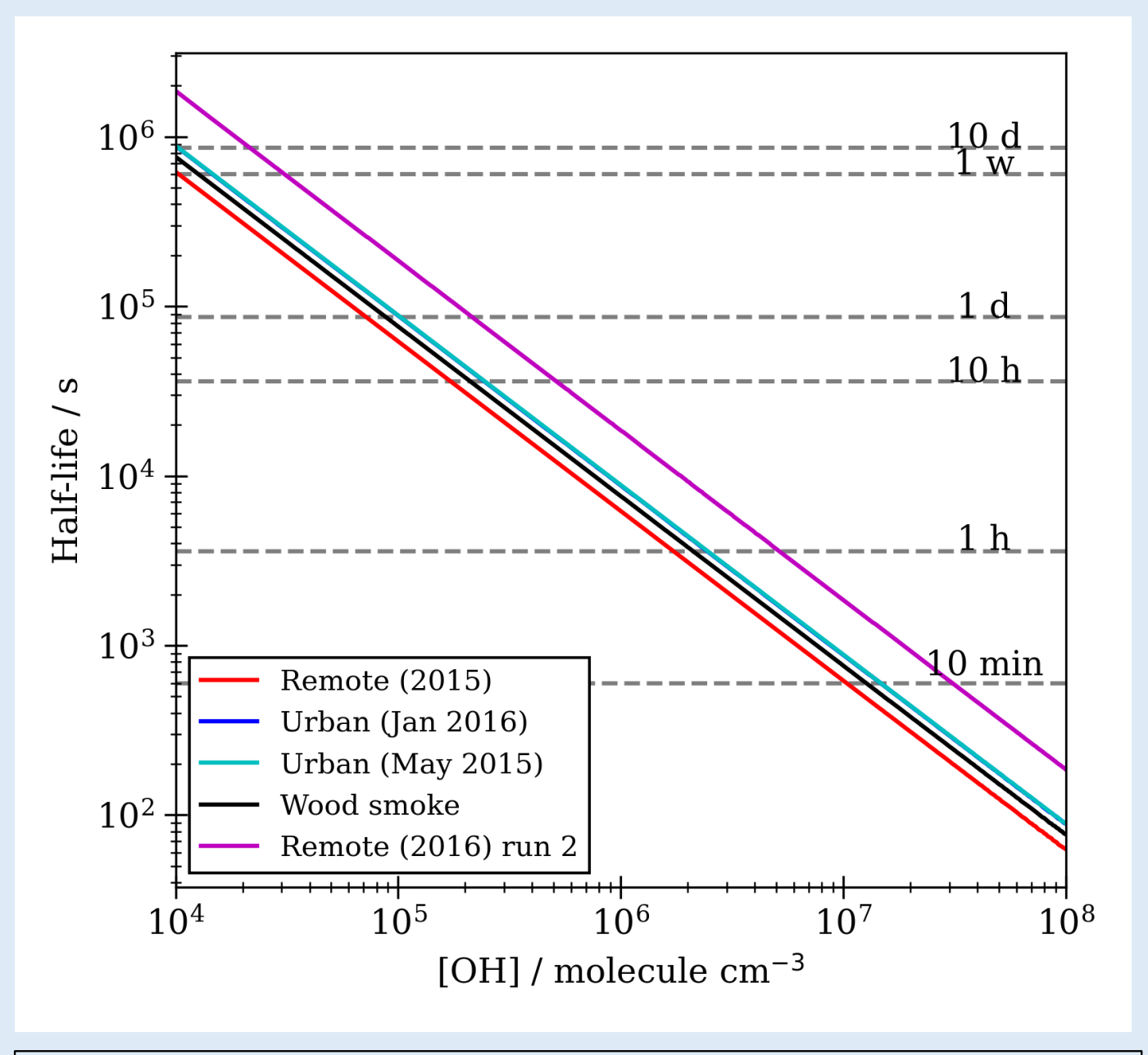

Figure 6.[4] The chemical half-life vs [OH] for models optimised to the kinetic data in Fig. 5.[4]

References:
1. Milsom et al., *Geosci. Model Dev.*, 2022, doi: 10.5194/gmd-15-7139-2022.
2. Shiraiwa et al., *Atmos. Chem. Phys.*, 2010, doi: 10.5194/acp-10-3673-2010.
3. Shiraiwa et al., *Atmos. Chem. Phys.*, 2012, doi: 10.5194/acp-12-2777-2012.
4. Shepherd et al., *Environ. Sci.: Atmos.*, 2022, doi: 10.1039/D2EA00013J.

GitHub repository (includes tutorials): https://github.com/tintin554/multilayerpy

Search “MultilayerPy” on YouTube for tutorial videos…