

Supplementary Materials: An Algorithm for In-Flight Spectral Calibration of Imaging Spectrometers. Remote Sensing 2016, 8, 1017

Gerrit Kuhlmann ^{1,*}, Andreas Hueni ², Alexander Damm ² and Dominik Brunner ^{1,*}

The spectral calibration algorithm described in the paper was built on top of the flexDOAS library which is a Python library that we are currently developing for the flexible implementation of DOAS-like retrieval algorithms. The development version of the flexDOAS library is available at [1].

The calibration algorithm is currently beta tested and will be added as an example script to the flexDOAS library in near future. The algorithm consists of the forward model and code for handling the APEX dataset and visualizing the results of the retrieval. A simplified version of the forward model is already available at [2].

In order to adapt the code to other hyperspectral sensors, users need to write their own Python script to write a sensor specific dataset class that can be used with flexDOAS. Interested users or testers are encouraged to contact us for more details. In the following we give a broad outline for such a script:

```
import doas
```

```
class MyDataset(doas.Dataset):
```

```
    @classmethod
```

```
    def from_datafile(cls, filename):
```

```
        """\
```

```
        Create a sensor-specific dataset containing radiance spectra ,
        initial calibration , other parameters and a priori information .
        """
```

```
        return cls (...)
```

```
    def get_observation(self, p):
```

```
        """\
```

```
        Return observation vector (y) and its error covariance matrix (Se).
        """
```

```
        return self.get("y", p), self.get("Se", p)
```

```
    def get_prior(self, p):
```

```
        """\
```

```
        Return a priori state vector (xa) and error covariance matrix (Sa)
        """
```

```
        return self.get("xa", p), self.get("Sa", p)
```

```
    def get_starting_vector(self, p):
```

```
        """\
```

```
        Return starting vector (x0).
        """
```

```
        return self.get("x0", p)
```

```

def get_parameter_vector(self, p):
    """\
    Return parameter vector (b).
    """
    return self.get("b", p)

def main(data_filename, output_filename):
    """\
    Main function for spectral calibration: It loads the dataset,
    initializes the forward model, creates the retrieval class,
    does the in-flight spectral calibration and saves results to
    "output_filename".
    """
    # create own dataset and forward model
    data = MyDataset.from_datafile(data_filename)

    # see GitHub repository for an example
    model = APEXCalibModel(...)

    # initialize retrieval class and do spectral calibration
    retrieval = doas.Retrieval(model, data)
    retrieval.solve_all()

    # save result to netCDF file
    retrieval.results.to_netcdf(output_filename)

if __name__ == "__main__":
    main(data_filename, output_filename)

```

References

1. Kuhlmann, G. flexDOAS Library, 2016. Available online: <https://github.com/gkuhl/flexDOAS> (accessed on 30 November 2016).
2. Kuhlmann, G. flexDOAS Library: Examples, 2016. Available online: <https://github.com/gkuhl/flexDOAS/blob/master/examples/> (accessed on 30 November 2016).