

Exercise No. 8 – Inversion theory: Optimal Estimation Method (OEM)

In this exercise you will work with “realistic” data measured by a water vapor radiometer. The data is not real but has been simulated for a well-known atmospheric state using ARTS. Simulated measurements allow to compare retrieval results to the true atmospheric state. The radiometer (Fig. 1) measures thermal radiation in a frequency range around the 22 GHz water vapor absorption line. As the pressure broadening of absorption lines varies with height the measurement contains information about the vertical water vapor profile. This information can be retrieved using the “Optimal Estimation Method” (OEM). The radiometer is placed in 10 km height, which resembles an upward looking airborne measurement. The scarce concentration of water vapor in the stratosphere allows to perform a linear retrieval approach. Retrievals that cover the whole atmosphere, including the highly absorbent lower troposphere, need more advanced retrieval approaches like an iterative OEM.



Figure 1: Onsala water vapor radiometer (photo: Peter Forkman)

1. Run the ARTS control file `forward_model.arts` to perform a forward simulation for a given a priori state (stored in `input/x_apriori.xml`).

Modify the plotting script `oem.py` to work on the following tasks.

2. Plot both the measured spectrum `y_measure` and the simulated *a priori* spectrum `y_apriori` as a function of the frequency grid. Add reasonable labels to your plots.
3. Plot the Jacobians `K` in a suitable way.

4. Implement the function `retrieve()` according to the OEM solution:

$$\hat{x} = x_0 + (K^T S_y^{-1} K + S_0^{-1})^{-1} K^T S_y^{-1} (y - y_0)$$

In Python, a matrix M can be transposed using `M.T` and inversed using `inv(M)`¹. Two matrices $M1$ and $M2$ can be multiplied using `M1 @ M2`.

5. Plot the retrieved water vapor profile `x_oem` alongside the *a priori* state `x_apriori` and the true atmospheric state `x_true` as a function of height z . Discuss the results.
6. Implement the function `averaging_kernel_matrix()` to calculate the same-named matrix:

$$A = (K^T S_y^{-1} K + S_0^{-1})^{-1} K^T S_y^{-1} K$$

Plot the columns (kernels) as a function of height and interpret the result.

7. The measurement response is defined as the sum over all averaging kernels in a given height (row). The measurement response indicates in which heights the measurement actually adds information to the retrieval result. Calculate the measurement response and plot it together with the averaging kernels. In which heights does the measurement provide useful information? Is it possible to estimate the vertical resolution?

¹We are using the inverse function `scipy.linalg.inv()` provided by the SciPy package.