OEM Retrievals in ARTS

Simon Pfreundschuh
Deparment of Earth, Space and Environment

CHALMERS
UNIVERSITY OF TECHNOLOGY

OEM Retrievals in ARTS

Goals

- Quick and easy setup of retrieval calculations in ARTS
- Good performance

OEM Retrievals in ARTS

Goals

- Quick and easy setup of retrieval calculations in ARTS
- Good performance

Current Status

- OEM implementation based on `invlib` ✓
- Handling covariance matrices ✓
- Computing error statistics ✓
- Setting up retrievals

OEM Retrievals in ARTS

Goals

- Quick and easy setup of retrieval calculations in ARTS
- Good performance

Current Status

- OEM implementation based on `invlib` ✓
- Handling covariance matrices ✓
- Computing error statistics ✓
- Setting up retrievals ⚠

This Presentation

- General retrieval workflow
- 1D ozone retrieval example

CHALMERS
UNIVERSITY OF TECHNOLOGY

The Optimal Estimation Method (OEM)

Formulation

- Gaussian prior: $\mathbf{x}_a \in \mathbb{R}^n, \mathbf{S}_x \in \mathbb{R}^{n \times n}$
- Gaussian measurement errors: $\mathbf{y}_f = \mathbf{F}(\mathbf{x}) \in \mathbb{R}^m, \mathbf{S}_\epsilon \in \mathbb{R}^{m \times m}$

MAP Estimator

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} (\mathbf{F}(\mathbf{x}) - \mathbf{y})^T \mathbf{S}_\epsilon^{-1} (\mathbf{F}(\mathbf{x}) - \mathbf{y}) + (\mathbf{x} - \mathbf{x}_a)^T \mathbf{S}_x^{-1} (\mathbf{x} - \mathbf{x}_a)$$

The Optimal Estimation Method (OEM)

Minimization

- Gauss-Newton:

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \left(\mathbf{K}_{\mathbf{x}_i}^T \mathbf{S}_\epsilon^{-1} \mathbf{K}_{\mathbf{x}_i} + \mathbf{S}_x^{-1}\right)^{-1} \left(\mathbf{K}_{\mathbf{x}_i}^T \mathbf{S}_\epsilon^{-1} (\mathbf{F}\mathbf{x}_i - \mathbf{y}) + \mathbf{S}_x^{-1}(\mathbf{x}_i - \mathbf{x}_a)\right)$$

$$\mathbf{x}_{i+1} = \mathbf{x}_a - \mathbf{S}_x \mathbf{K}_{\mathbf{x}_i}^T \left(\mathbf{K}_{\mathbf{x}_i} \mathbf{S}_x \mathbf{K}_{\mathbf{x}_i}^T + \mathbf{S}_\epsilon\right)^{-1} \mathbf{K}_{\mathbf{x}_i}^T \mathbf{S}_\epsilon^{-1} \left(\mathbf{F}\mathbf{x}_i - \mathbf{y} - \mathbf{K}_{\mathbf{x}_i}(\mathbf{x} - \mathbf{x}_a)\right)$$

CHALMERS
UNIVERSITY OF TECHNOLOGY

The Optimal Estimation Method (OEM)

Minimization

- Gauss-Newton:

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \underbrace{\left(\mathbf{K}_{\mathbf{x}_i}^T \mathbf{S}_\epsilon^{-1} \mathbf{K}_{\mathbf{x}_i} + \mathbf{S}_x^{-1}\right)}_{n \times n}^{-1} \left(\mathbf{K}_{\mathbf{x}_i}^T \mathbf{S}_\epsilon^{-1} (\mathbf{F}\mathbf{x}_i - \mathbf{y}) + \mathbf{S}_x^{-1} (\mathbf{x}_i - \mathbf{x}_a)\right)$$

$$\mathbf{x}_{i+1} = \mathbf{x}_a - \mathbf{S}_x \mathbf{K}_{\mathbf{x}_i}^T \underbrace{\left(\mathbf{K}_{\mathbf{x}_i} \mathbf{S}_x \mathbf{K}_{\mathbf{x}_i}^T + \mathbf{S}_\epsilon\right)}_{m \times m}^{-1} \mathbf{K}_{\mathbf{x}_i}^T \mathbf{S}_\epsilon^{-1} \left(\mathbf{F}\mathbf{x}_i - \mathbf{y} - \mathbf{K}_{\mathbf{x}_i}(\mathbf{x} - \mathbf{x}_a)\right)$$

CHALMERS
UNIVERSITY OF TECHNOLOGY

The Optimal Estimation Method (OEM)

Minimization

- Gauss-Newton:

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \underbrace{\left(\mathbf{K}_{\mathbf{x}_i}^T \mathbf{S}_\epsilon^{-1} \mathbf{K}_{\mathbf{x}_i} + \mathbf{S}_x^{-1}\right)}_{n \times n}^{-1} \left(\mathbf{K}_{\mathbf{x}_i}^T \mathbf{S}_\epsilon^{-1} (\mathbf{F}\mathbf{x}_i - \mathbf{y}) + \mathbf{S}_x^{-1} (\mathbf{x}_i - \mathbf{x}_a)\right)$$

$$\mathbf{x}_{i+1} = \mathbf{x}_a - \mathbf{S}_x \mathbf{K}_{\mathbf{x}_i}^T \underbrace{\left(\mathbf{K}_{\mathbf{x}_i} \mathbf{S}_x \mathbf{K}_{\mathbf{x}_i}^T + \mathbf{S}_\epsilon\right)}_{m \times m}^{-1} \mathbf{K}_{\mathbf{x}_i}^T \mathbf{S}_\epsilon^{-1} \left(\mathbf{F}\mathbf{x}_i - \mathbf{y} - \mathbf{K}_{\mathbf{x}_i} (\mathbf{x} - \mathbf{x}_a)\right)$$

- Levenberg-Marquardt

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \left(\mathbf{K}_{\mathbf{x}_i}^T \mathbf{S}_\epsilon^{-1} \mathbf{K}_{\mathbf{x}_i} + (1+\gamma) \mathbf{S}_x^{-1}\right)^{-1} \left(\mathbf{K}_{\mathbf{x}_i}^T \mathbf{S}_\epsilon^{-1} (\mathbf{F}\mathbf{x}_i - \mathbf{y}) + \mathbf{S}_x^{-1} (\mathbf{x}_i - \mathbf{x}_a)\right)$$

CHALMERS
UNIVERSITY OF TECHNOLOGY

The Optimal Estimation Method (OEM)

Minimization

- Gauss-Newton:

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \underbrace{\left(\mathbf{K}_{\mathbf{x}_i}^T \mathbf{S}_\epsilon^{-1} \mathbf{K}_{\mathbf{x}_i} + \mathbf{S}_x^{-1}\right)}_{n \times n}^{-1} \left(\mathbf{K}_{\mathbf{x}_i}^T \mathbf{S}_\epsilon^{-1} (\mathbf{F}\mathbf{x}_i - \mathbf{y}) + \mathbf{S}_x^{-1} (\mathbf{x}_i - \mathbf{x}_a)\right)$$

$$\mathbf{x}_{i+1} = \mathbf{x}_a - \mathbf{S}_x \mathbf{K}_{\mathbf{x}_i}^T \underbrace{\left(\mathbf{K}_{\mathbf{x}_i} \mathbf{S}_x \mathbf{K}_{\mathbf{x}_i}^T + \mathbf{S}_\epsilon\right)}_{m \times m}^{-1} \mathbf{K}_{\mathbf{x}_i}^T \mathbf{S}_\epsilon^{-1} \left(\mathbf{F}\mathbf{x}_i - \mathbf{y} - \mathbf{K}_{\mathbf{x}_i} (\mathbf{x} - \mathbf{x}_a)\right)$$

- Levenberg-Marquardt

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \underbrace{\left(\mathbf{K}_{\mathbf{x}_i}^T \mathbf{S}_\epsilon^{-1} \mathbf{K}_{\mathbf{x}_i} + (1+\gamma) \mathbf{S}_x^{-1}\right)}_{n \times n}^{-1} \left(\mathbf{K}_{\mathbf{x}_i}^T \mathbf{S}_\epsilon^{-1} (\mathbf{F}\mathbf{x}_i - \mathbf{y}) + \mathbf{S}_x^{-1} (\mathbf{x}_i - \mathbf{x}_a)\right)$$

CHALMERS
UNIVERSITY OF TECHNOLOGY

The Optimal Estimation Method (OEM)

Computation

- Each optimization step requires solution of a linear system:

$$\underbrace{\left(\mathbf{K}_{\mathbf{x}_i}^T \mathbf{S}_\epsilon^{-1} \mathbf{K}_{\mathbf{x}_i} + \mathbf{S}_x^{-1}\right)}_{\mathbf{M}} \mathbf{a} = \mathbf{b}$$

$$\underbrace{\left(\mathbf{K}_{\mathbf{x}_i} \mathbf{S}_x \mathbf{K}_{\mathbf{x}_i}^T + \mathbf{S}_\epsilon\right)}_{\mathbf{M}} \mathbf{a} = \mathbf{b}$$

The Optimal Estimation Method (OEM)

Computation

- Each optimization step requires solution of a linear system:

$$\underbrace{\left(\mathbf{K}_{\mathbf{x}_i}^T \mathbf{S}_\epsilon^{-1} \mathbf{K}_{\mathbf{x}_i} + \mathbf{S}_x^{-1}\right)}_{\mathbf{M}} \mathbf{a} = \mathbf{b}$$

$$\underbrace{\left(\mathbf{K}_{\mathbf{x}_i} \mathbf{S}_x \mathbf{K}_{\mathbf{x}_i}^T + \mathbf{S}_\epsilon\right)}_{\mathbf{M}} \mathbf{a} = \mathbf{b}$$

- **Direct solver**:
  - Decompose **M** and solve the system forward and back substitution
  - Requires explicit computation of **M**

CHALMERS
UNIVERSITY OF TECHNOLOGY

The Optimal Estimation Method (OEM)

Computation

- Each optimization step requires solution of a linear system:

$$\underbrace{\left(\mathbf{K}_{\mathbf{x}_i}^T \mathbf{S}_\epsilon^{-1} \mathbf{K}_{\mathbf{x}_i} + \mathbf{S}_x^{-1}\right)}_{\mathbf{M}} \mathbf{a} = \mathbf{b}$$

$$\underbrace{\left(\mathbf{K}_{\mathbf{x}_i} \mathbf{S}_x \mathbf{K}_{\mathbf{x}_i}^T + \mathbf{S}_\epsilon\right)}_{\mathbf{M}} \mathbf{a} = \mathbf{b}$$

- **Direct solver**:
  - Decompose **M** and solve the system forward and back substitution
  - Requires explicit computation of **M**
- **Conjugate gradient method**:
  - Avoids computation of the linear system
  - Iterative method, performance depends on **M**

Retrieval Workflow

1. Define retrieval quantities and covariance matrices:

```
retrievalDefInit
Covmat1D(...)                # Create covariance matrix block
retrievalAddAbsSpecies(...)  # Add RQ to Jacobian and add block to covmat_sx
...
covmat_blockSetDiagonal(...) # Create diagonal covariance matrix
covmat_seSet(...)            # Set block as covmat_se
retrievalDefClose
```

CHALMERS
UNIVERSITY OF TECHNOLOGY

Retrieval Workflow

1. Define retrieval quantities and covariance matrices:

```
retrievalDefInit
Covmat1D(...)                 # Create covariance matrix block
retrievalAddAbsSpecies(...)   # Add RQ to Jacobian and add block to covmat_sx
...
covmat_blockSetDiagonal(...)  # Create diagonal covariance matrix
covmat_seSet(...)             # Set block as covmat_se
retrievalDefClose
```

2. Define forward model agenda:

```
AgendaSet( inversion_iterate_agenda ){
  ...
}
```

CHALMERS
UNIVERSITY OF TECHNOLOGY

Retrieval Workflow

1. Define retrieval quantities and covariance matrices:

```
retrievalDefInit
Covmat1D(...)                   # Create covariance matrix block
retrievalAddAbsSpecies(...)     # Add RQ to Jacobian and add block to covmat_sx
...
covmat_blockSetDiagonal(...)    # Create diagonal covariance matrix
covmat_seSet(...)               # Set block as covmat_se
retrievalDefClose
```

2. Define forward model agenda:

```
AgendaSet( inversion_iterate_agenda ){
  ...
}
```

3. Run oem calculation:

```
OEM(...)
```

CHALMERS
UNIVERSITY OF TECHNOLOGY

Retrieval Workflow

1. Define retrieval quantities and covariance matrices:

```
retrievalDefInit
Covmat1D(...)                 # Create covariance matrix block
retrievalAddAbsSpecies(...)   # Add RQ to Jacobian and add block to covmat_sx
...
covmat_blockSetDiagonal(...)  # Create diagonal covariance matrix
covmat_seSet(...)             # Set block as covmat_se
retrievalDefClose
```

2. Define forward model agenda:

```
AgendaSet( inversion_iterate_agenda ){
  ...
}
```

3. Run oem calculation:

```
OEM(...)
```

4. Error analysis:

```
avkCalc       # Averaging kernel matrix
covmat_ssCalc # Smoothing error
covmat_soCalc # Observation noise
```

CHALMERS
UNIVERSITY OF TECHNOLOGY

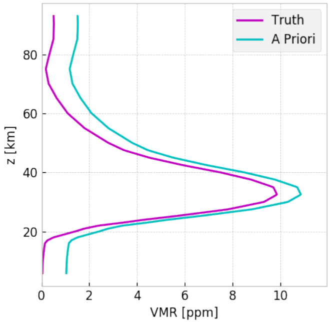
ARTS Nomenclature

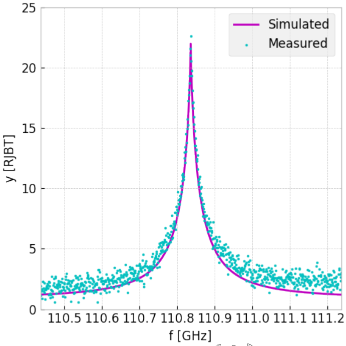
ARTS Name	Mathematical Notation	Description
`x`	$\mathbf{x} \in \mathbb{R}^n$	State vector
`y`	$\mathbf{y} \in \mathbb{R}^m$	Measurement vector
`yf`	$\mathbf{y_f} = \mathbf{F}(\mathbf{x}) \in \mathbb{R}^m$	Simulated measurement vector
`covmat_sx`	$\mathbf{S}_x \in \mathbb{R}^{n \times n}$	A priori covariance matrix
`covmat_se`	$\mathbf{S}_\epsilon \in \mathbb{R}^{m \times m}$	Measurement error covariance matrix
`avk`	$\mathbf{S}_\epsilon \in \mathbb{R}^{n \times n}$	Averaging kernel matrix
`covmat_ss`	$\mathbf{S}_s \in \mathbb{R}^{n \times n}$	Smoothing error
`covmat_so`	$\mathbf{S}_o \in \mathbb{R}^{n \times n}$	Retrieval noise
`covmat_block`		Covariance matrix block
`comvat_inv_block`		Inverse of covariance matrix block

CHALMERS
UNIVERSITY OF TECHNOLOGY

1D Retrieval Example

- Observation of O_3 110.836 GHz
- Ground-based (60° zenith angle)
- A priori off with 1 ppm
- Noisy measurement with polynomial offset





CHALMERS
UNIVERSITY OF TECHNOLOGY

1D Retrieval Example, Retrieval Definition

Retrieval Definition

- `retrievalDefInit` , `retrievalDefClose`
- Similar to definition of Jacobian quantities

Covariance Matrices

- `covmat1D, covmat1DMarkov, ...` create correlation blocks and store them inside `covmat_block (covmat_inv_block)` WSVs
- `retrievalAddAbsSpecies, retrievalAddPolyfit, ...` functions add quantity to Jacobian and correlation block in `covmat_block` to `covmat_sx`
- Use `covmat_seSet` to add `covmat_block` to `covmat_se`

CHALMERS
UNIVERSITY OF TECHNOLOGY

1D Retrieval Example, Retrieval Definition

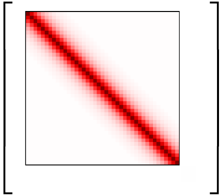
```
retrievalDefInit
```

$$\mathbf{S}_x =$$

1D Retrieval Example, Retrieval Definition

```
retrievalDefInit

# Ozone
covmat1D(g1 = z_ret_grid,
         sigma1 = sigma_x,
         lc1 = lcs,
         fname = "gauss")

retrievalAddAbsSpecies(species = "O3",
                       unit = "logrel",
                       g1 = p_ret_grid,
                       g2 = lat_grid,
                       g3 = lon_grid)
```

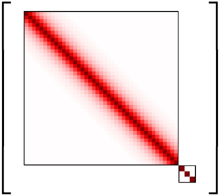
$\mathbf{S}_x =$



CHALMERS
UNIVERSITY OF TECHNOLOGY

1D Retrieval Example, Retrieval Definition

```
retrievalDefInit

# Ozone
covmat1D(g1 = z_ret_grid,
         sigma1 = sigma_x,
         lc1 = lcs,
         fname = "gauss")

retrievalAddAbsSpecies(species = "O3",
                       unit = "logrel",
                       g1 = p_ret_grid,
                       g2 = lat_grid,
                       g3 = lon_grid)

# Polynomial Baseline
covmatSetDiagonal(sigma = sigma_e)
retrievalAddPolyfit(poly_order = 2)
```

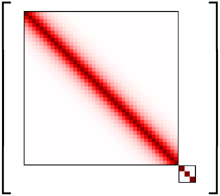
$\mathbf{S}_x =$ 

CHALMERS
UNIVERSITY OF TECHNOLOGY

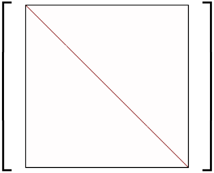
1D Retrieval Example, Retrieval Definition

```
retrievalDefInit

# Ozone
covmat1D(g1 = z_ret_grid,
         sigma1 = sigma_x,
         lc1 = lcs,
         fname = "gauss")

retrievalAddAbsSpecies(species = "O3",
                       unit = "logrel",
                       g1 = p_ret_grid,
                       g2 = lat_grid,
                       g3 = lon_grid)

# Polynomial Baseline
covmatSetDiagonal(sigma = [10.0])
retrievalAddPolyfit(poly_order = 2)

# S_e
covmatSetDiagonal(sigma = sigma_e)
covmat_seSet
```

$\mathbf{S}_x =$



$\mathbf{S}_e =$

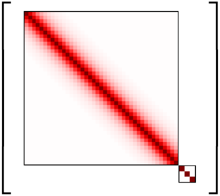


CHALMERS
UNIVERSITY OF TECHNOLOGY

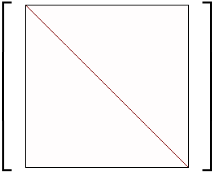
1D Retrieval Example, Retrieval Definition

```
retrievalDefInit

# Ozone
covmat1D(g1 = z_ret_grid,
         sigma1 = sigma_x,
         lc1 = lcs,
         fname = "gauss")

retrievalAddAbsSpecies(species = "O3",
                       unit = "logrel",
                       g1 = p_ret_grid,
                       g2 = lat_grid,
                       g3 = lon_grid)

# Polynomial Baseline
covmatSetDiagonal(sigma = [10.0])
retrievalAddPolyfit(poly_order = 2)

# S_e
covmatSetDiagonal(sigma = sigma_e)
covmat_seSet

retrievalDefClose
```

$\mathbf{S}_x =$



$\mathbf{S}_e =$



CHALMERS
UNIVERSITY OF TECHNOLOGY

1D Retrieval Example, Forwar Model Setup

- Interface between ARTS and OEM implementation
- Should not need to be modified

```
AgendaSet( inversion_iterate_agenda ){

  # Map x to ARTS' variables
  x2artsStandard

  # To be safe, rerun checks dealing with the atmosphere
  atmfields_checkedCalc
  atmgeom_checkedCalc

  # Calculate yf and Jacobian matching x.
  yCalc( y=yf )

  # Add baseline term
  VectorAddVector( yf, yf, y_baseline )

  # This method takes cares of some "fixes" that are needed to get the Jacobian
  # right for iterative solutions. No need to call this WSM for linear inversions.
  jacobianAdjustAfterIteration
}
```

CHALMERS
UNIVERSITY OF TECHNOLOGY

1D Retrieval Example, OEM Calculation

Run OEM Calculation

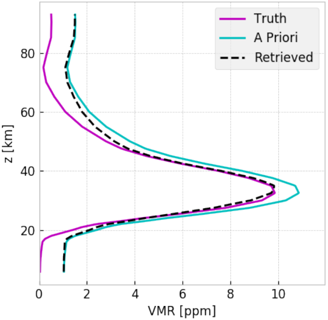
```
OEM(method="gn_cg")
```

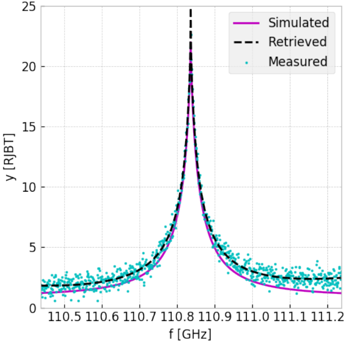
The Method Argument

Argument String	Minimization Method	Solver	Formulation
`"li"`	linear	direct	n-form
`"li_cg"`	linear	CG	n-form
`"li_m"`	linear	direct	m-form
`"li_cg_m"`	linear	CG	m-form
`"gn"`	Gauss-Newton	direct	n-form
`"gn_cg"`	Gauss-Newton	CG	n-form
`"gn_m"`	Gauss-Newton	direct	m-form
`"gn_cg_m"`	Gauss-Newton	CG	m-form
`"lm"`	Levenberg-Marquardt	direct	n-form
`"lm_cg"`	Levenberg-Marquardt	CG	n-form

1D Retrieval Example, Step 2

Results



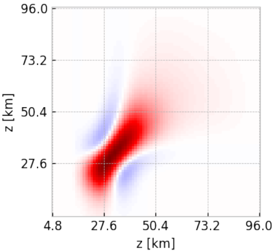


CHALMERS
UNIVERSITY OF TECHNOLOGY

Error Analysis

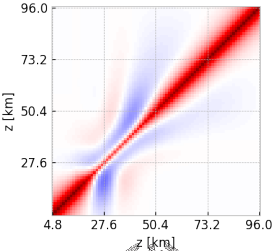
Averaging Kernel Matrix

```
# Averaging Kernel:
avkCalc
```



Smoothing Error

```
# Smoothing Error:
covmat_ssCalc
```

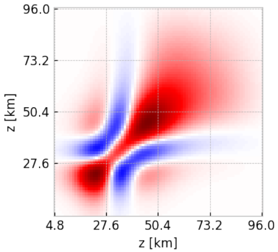


CHALMERS
UNIVERSITY OF TECHNOLOGY

Error Analysis

Observation Noise

```
# Observation Noise
covmat_soCalc
```



Conclusion and Outlook

Conclusions

- Extensive OEM functionality available in ARTS
- Some work on setting up retrievals remains

Future Work

- Perform retrievals
- Extend and improve retrieval functionality in arts
- Examples and documentation