

# Change Log for Rate Fitting Code

L. Strow, S. Desouza-Machado

15 August 2011

## 1 Notes

- Src is at `~/Work/Rates/Fit_pkg`

## 2 Changes

### 2.1 *2011-08-15 Mon (LLS)*

- Jacobians now normalized.  $\max(\text{sum}(\text{abs}(\text{temp}))) = \max(\text{sum}(\text{abs}(\text{water}))) = \max(\text{abs}(\text{gasi}))$  Jacobians. See `get_jac.m`
- `get_jac.m` no longer squeezes the jacobians for the parameters being fit. This is left to `retrieval.m`. This way, the jacobians outputted by `get_jacs.m` are “full” and can be used by `adjust_rates.m` to remove the effects of some gas using a known rate.
- Three lambda’s are now used. `gas/water/temp`. The original lambda is now only used if you invoke regularization only on the diagonal.
- Test directory using new covariance matrices (L1 with larger numbers for `i=1:6`) and a true `apriori_zero` = all zeros.

### 2.2 *2011-08-08 Mon*

- Removed “cluster” stuff.
- Many many code changes, mostly to simplify.
  - Large speed-up by doing multiple `pinv`’s only once and saving
  - Assume cov input matrix is the regularization matrix!
  - Used `inv` instead of `pinv` for the gain matrix

- Both the Jacobian file and the covariance file now must have same ordering. [gases columns (1-5)  $T_{\text{surf}}$   $wv(1:97)$ ,  $t(1:97)$ ]
- Removed lambda loop, user creates this loop externally if needed
- For now, put gain and ak matrices in driver output
- Fixed qrenorm to reflect that T-Jacobians were scaled by 0.01, not 0.1
- Changed `find_covderivoperators` to remove qrenorm scaling on L matrices
- lambda input is now not scaled by  $10^{(\text{lambda})}$
- NOTE: testing now done with a L1 matrix (unity lambda2)

### **2.3 2011-08-05 08:00**

- the Cluster directory allows you to put the below job on the cluster, using `run_tara`

### **2.4 2011-08-04 21:00**

- `set_default_struct` is used to set defaults
- these are then overwritten by calling `driver = override_defaults(driver,ix);` where “ix” is the latbin you want to run. So for example a loop to do all latbins would be

```
for ix=1:36
    clear driver;
    run_retrieval;
end
```

### **2.5 2011-08-02 05:05**

- jacobian file now has the qrenormalization information

### **2.6 2011-08-01 21:50**

- `driver.map_jac2cov` allows you to map indices from Jacobian to Covariance

## 2.7 2011-08-01 14:14

- Covariance matrix format changed; no top level structure name. `oem_lls.m` changed to reflect this, see `b0`.
- Apriori format changed; no top level structure name. `oem_lls.m` changed to reflect this, see `zset`, and `zstd`.
- Changed debugging approach, using Matlab code like:

```
if driver.debug
    addpath Debug
    debug_cov
    rmpath Debug
end
```

These scripts are in the `Debug` directory.

- Created directory `Changelog` to hold this file and its various output formats.
- `driver.map_jac2cov` allows you to map indices from Jacobian to Covariance

## 2.8 2012-08-27 to 2012-09-27

- the column `jacs/stemp` are now controlled by `driver.jacobian.qstnames` and `driver.jacobian.qstYesOrNo`; the former gives the names while the latter tells the code whether to turn the `jac` on or off. For example

```
driver.jacobian.qstnames = {'CO2trop' 'CO2strat' 'O3trop' 'O3strat' 'N2O' 'CO' 'CH4'};
driver.jacobian.qstYesOrNo = [1      1      1      1      1      1      1];
```

- the number of layers for (`Q1,Q2 .. QN,T`) must be the same, and specified by `driver.jacobian.numlays` (default 97). In addition, you can have more than one radiating gas whose profile you want to retrieve; this is controlled by `driver.jacobian.numQlays`. So for example if you wish to retrieve 97 layer profiles for `H2O` and `HDO` use

```
driver.jacobian.numlays      = 97;
driver.jacobian.numQlays     = 2;                                     %% in addition to water
driver.jacobian.Q2jacindex   = 1:97;
```

- Changed the  $\lambda$  parameters so you either send in one constant value for all column jacs/stemp, water vapor layers and temperature layers OR can send in individual parameters for each of the jacs that you are sending in

a) driver.oem.lambda_qst can be length(1)	in which case all "gases" get the
driver.oem.lambda_qst can be length(qst)	in which case each "gases" get ind
driver.oem.lambda_qst can be [length(qst) -9999]	in which case each "gases" get ind
b) driver.oem.lambda_Qn can be length(1)	in which case all "Qn" lays get t
driver.oem.lambda_Qn can be length(Qn)	in which case each "Qn" lays get i
driver.oem.lambda_Qn can be [length(Qn) -9999]	in which case each "Qn" lays get i
c) driver.oem.lambda_temp can be length(1)	in which case all "temps" get the
driver.oem.lambda_temp can be length(itype)	in which case each "temps" get ind
driver.oem.lambda_temp can be [length(itype) -9999]	in which case each "temps" get ind