MIPAS Microwindow Processing Scheme (MWPROC)
PO-MA-OXF-GS-0019
CCN5, Task2 Contract 11886/96/NL/GS

Anu Dudhia *

August 30th, 2001

This document describes the installation and use of the MIPAS Microwindow Processing suite MWPROC. This modifies an existing Microwindow Database file and creates compressed Look-Up Tables containing absorber cross-sections, and associated diagnostics.

Processing comprises the following stages

1. **ABS** Update MW absorber list
2. **LBL** Create reference spectra
3. **TAB** Create uncompressed LUTs for each absorber with optimised p, T axes
4. **SVD** Perform SVD to compress LUTs and optimise no. singular values
5. **GRD** Derive MW irregular grid to optimise wavenumber axis
6. **ERR** Calculate database quantifiers for combined LUT+GRD forward model error

The suite consists of a series of FORTRAN programs and shell scripts designed to operate in a Unix environment. In addition, the Reference Forward Model (RFM) and associated HITRAN binary file are required (not supplied with this package).

Document History

30AUG01 Modifications to conform to Aug'01 update of software suite. (NB: the examples given in this document are not reproducible with the new code but serve only to demonstrate the format)

02OCT00 Original version

*Atmospheric, Oceanic and Planetary Physics, Oxford University, UK

Contents

1 Installation

1.1 Obtaining the MWPROC suite

The processing suite can be obtained by anonymous ftp.

```
%ftp florence.atm.ox.ac.uk
anonymous               [ response to ``Name'' ]
[your email address]    [ response to ``Password'' ]
ftp> cd /pub/mipas/mwproc
ftp> bin
ftp> mget mwproc.tar.Z
ftp> bye
```

1.2 Unpacking the tar file

The tar file should be placed in a directory with at least 1Gbyte of free disk space. The file should be uncompressed and then unpacked using

```
%uncompress mwproc.tar.Z
%tar -xvf mwproc.tar
```

This will then generate a directory `/mwproc` containing subdirectories listed in Table 1.

Table 1: List of sub-directories and contents contained with the `mwproc.tar` package.

Directory	Description and initial contents	References
Overall Inputs/Outputs		
`/CS`	New cross-section (compressed LUTs) files. Contains example.	§ 2.4
`/MW`	Microwindow database files. Contains example.	§ B
`/dia`	Diagnostic outputs. Contains examples.	§ 2.1,2.3,2.4,2.5
Control of Processing		
`/exec`	Executable programs. Empty.	§ 1.4
`/rfm`	Directory for RFM executable. Empty.	§ 1.5
`/rfm_files`	Directory containing RFM input files. (excluding binary HITRAN)	§ 1.6
`/shell`	Shell scripts for executing each stage.	§ 2
`/source`	Processing source code and compilation script `compile.sh`.	§ 1.4
`/source/ARPACK`	ARPACK source code required by `svdini.f`.	§ 1.3
Work Space		
`/abs`	Work space for ABS stage. Empty.	§ 2.1
`/err`	Work space for ERR stage. Empty.	§ 2.6
`/grd`	Work space for GRD stage. Empty.	§ 2.5
`/lbl`	Work space for LBL stage. Empty.	§ 2.2
`/svd`	Work space for SVD stage. Empty.	§ 2.4
`/tab`	Work space for TAB stage. Empty.	§ 2.3

1.3 ARPACK modules (/source/ARPACK)

The SVD stage uses a number of modules from the public domain ARPACK library [2]. Rather than install the entire library, only the necessary modules have been extracted and placed in the /source/ARPACK directory. These are compiled into a library /source/libarpack.a by the compilation script compile.sh.

1.4 Compiling the software (/source)

The software suite is compiled from within the /source directory by typing

```
%compile.sh
```

The executables are written into the /exec directory.

1.5 Installing the RFM (/rfm)

The Reference Forward Model (RFM) v3.90 or higher has to be installed directory /rfm.

In the 'standard' distributed version of the RFM, arrays required for use with look-up tables and irregular grids are set to a minimum size to save memory. Attempting to run the RFM within this suite will lead to a number of error messages indicating which array dimensions (contained in include file rfmsiz.inc) have to be increased.

To save repeated compilations, it is recommended that the array dimensions are initially set to the values listed in Table 2 and the RFM recompiled and placed in /rfm.

Table 2: Recommended RFM minimum array dimensions for use with the MWPROC suite. These are all parameters within the RFM include file rfmsiz.inc.

Parameter	Maximum number of ...
MAXCLC=2500	calculated paths
MAXGRD=5000	irregular grid points
MAXLFL=20	LUT files used in single run
*MAXLUG=10000	Spectral grid points in uncompressed LUT
MAXLUL=30	singular values in any compressed LUT
MAXLUT=20	LUTs used in single microwindow
MAXLUV=120000	LUT wavenumber axis points for single microwindow (all LUTs combined)
MAXLUX=30000	LUT (p, T) axis points for single microwindow (all LUTs combined)
MAXXSP=5000000	Tabulation points for cross-section data for one heavy molecule

*Only relevant for RFM v4.10 and higher versions

In addition, the parameter MAXWID may be reduced to 6 to save memory (this is maximum number of widemesh (1 cm^{-1}) intervals required for single spectrum, and 6 is adequate for all MIPAS MWs plus margins for the ILS convolution).

1.6 RFM Input Files (/rfm_files)

The programs *drv.f and tabini.f generate RFM Driver Tables for various processing stages. These refer to input files contained in the directory /rfm_files summarised in Table 3 which are supplied as part of the suite.

In addition, an RFM-compatible binary version of the HITRAN data named hitran_mipas.bin is assumed to exist in the /rfm_files directory. This is not supplied with the suite so must be installed by the

user. Typically this would be the standard 1996 HITRAN dataset plus additional lines specifically added for modelling MIPAS spectra.

The actual names and directories of each RFM input file are all contained in a single include file `rfminp.inc` in the `/source` directory. This can be modified if any changes are required, and the entire suite should then be recompiled.

Table 3: RFM input files supplied with the MWPROC suite (in directory `/rfm_files`).

File	Description
`*.atm`	Atmospheric profiles
`*.xsc`	Heavy molecule cross-section data
`*.ails`	Apodised Instrument Line Shapes
`rfm_1km5.fov`	MIPAS Field of View at 1.5km spacing

2 Running the Processing Suite

This section describes how to run each stage of the processing suite. In each case, the procedure is the same:

```
%cd xxx
%../shell/xxx.sh [mwlabel] [mwfile]
```

where `xxx` = `abs`, `lbl`, `tab`, `svd`, `grd` or `err`; `[mwlabel]` is the microwindow label (e.g. `PT__0001`) and `[mwfile]` is the database file containing the microwindow (e.g. `../MW/MW_PT.DAT`). For convenience, it is recommended that that each stage is run from its appropriate work space directory as above, but any first level subdirectory of `/mwproc` can be used.

Each shell script also has a limited capability to be restarted without having to repeat the entire process. This is achieved by adding a 3rd parameter `RESUME`

```
%../shell/xxx.sh [mwlabel] [mwfile] RESUME
```

Because of the possibility of confusing RFM driver tables, it is not advisable to run more than one processing stage at a time in any directory.

Generally, the processing should complete without any further user intervention. The main exception is the TAB stage, where the initial uncompressed LUTs may contain unacceptably large interpolation errors before any optimisation of the p, T axes can be performed. Appendix A deals with this situation.

2.1 ABS: Update MW absorber list

2.1.1 Overview

All absorbers already listed in the MW database, plus any other absorbers known to be significant in the spectral range are considered. The RFM is used to calculate MW radiances for three atmospheres containing single absorbers in isolation with their minimum, standard and maximum concentrations, all using a maximum temperature profile. The maximum radiance at any altitude from any profile is determined (usually, but not always, this comes from the maximum concentration profile). If this value is larger than NESR/10, the absorber is included, if smaller it is excluded. The included absorbers are ordered in terms of decreasing significance. The modified absorber list is then written to the database file and this list is used subsequent stages. A separate absorber diagnostics file is also created.

2.1.2 Processing

Table 4 summarises the processing within the ABS stage. Processing one microwindow takes around 10 minutes.

Table 4: Summary of the ABS processing stage

program	Description	`input(s)`	`output(s)`
absdrv	Generate RFM driver tables	`[mwfile]`	`rfm_[mwlabel]_[gas]_[atm].drv`
rfm	Generate single-species spectra	`rfm*drv`	`rad_[mwlabel][alt].asc_[gas]_[atm]`
absrad	Analyse spectra	`rad*[atm]`	`[mwlabel].abs`
absimk	Update MW Database absorber list	`[mwlabel].abs`	`[mwfile]`
	Upon successful completion:	`[mwlabel].abs`	*moved to* `../dia/`

`[atm]` is `min`, `max` or `std`; `[alt]` is tangent altitude [m] (e.g., `08000`); `[gas]` is the species (e.g., `clono2`)

2.1.3 Outputs

1. Absorber diagnostics saved in `../dia/[mwlabel].abs`
2. Modified microwindow database `[mwfile]` (see Table 16).

2.1.4 Diagnostics

An example of an absorber diagnostics file `[mwlabel].abs` is given in Table 5. This file summarises the contribution of each absorber considered for the microwindow.

Table 5: Example of an ABS diagnostics file `[mwlabel].abs`

```
PT__0003   697.575   700.575    8   53      59.0    9
co2         2 0.114E+05   38   36   698.450  160.       32   42   698.600
h2o         1 0.650E+04    8   19   698.025 0.427E+04    8    8   697.750
o3          3 0.623E+04   14    5   697.675 0.330E+04   11   63   699.125
hcn        23 0.158E+04    8  105   700.175 0.138E+04    8  105   700.175
c2h2       26 0.110E+04    8   42   698.600 0.101E+04    8   42   698.600
no2        10  150.        11  107   700.225  133.       11  107   700.225
ch3cl   *  24  134.         8   80   699.550 0.000E+00    0    0   697.550  ---
n2o         4  87.7         8  109   700.275  25.7       11  109   700.275
nh3        11  2.55        11   87   699.725  2.53       11   87   699.725  +++
```

Record#1 Contains the microwindow label `PT_0003`, the spectral boundaries `697.575 700.575` [cm^{-1}], the tangent altitude boundaries `8 53` [km], the NESR `59.0` [$\mathrm{nW/(cm^2\ sr\ cm^{-1})}$] and the number of absorbers considered in this analysis `9` (= number of following records).

Record#2 Contains the analysis for the most significant absorber (ordered by maximum radiance contribution, column#3). The columns are the chemical formula `co2`, the HITRAN/RFM index `2`, the maximum radiance contribution `0.114E+05` [$\mathrm{nW/(cm^2\ sr\ cm^{-1})}$], the tangent altitude of the maximum `38` [km], the spectral grid point# `36` and wavenumber `698.450` [cm^{-1}] of the maximum, the maximum *variation* in radiance (between any pair of the min, max and standard VMR profiles) `160.` [$\mathrm{nW/(cm^2\ sr\ cm^{-1}}$], the altitude of the max variation `32` [km] and the spectral grid point# `42` and wavenumber `698.600` [cm^{-1})] of the max variation

Record#$n+1$ Repeat for absorber#n.

Other symbols

`*` (`ch3cl` record) indicates a species not listed for the original microwindow but added at this stage (by program `absdrv`).

`---` (`ch3cl` record) indicates a species where the variability is less than NESR/10 so could be combined with other low-variability species into a single fixed-VMR LUT (this is no longer a processing option).

`+++` (`nh3` record) indicates that the maximum contribution of the absorber is less than NESR/10 so the absorber can be excluded from further consideration.

2.2 LBL: Create reference spectra

2.2.1 Overview

Performs line-by-line calculations for the 'exact' spectra for all altitudes and atmospheres. These are used as reference spectra for optimisations within the TAB and SVD stages. After the SVD stage is successfully completed, these spectra are deleted. The GRD stage also uses reference spectra, but these are required on the fine grid so are generated only during that stage.

2.2.2 Processing

Table 6 summarises the processing within the LBL stage. Processing one microwindow takes around 10 minutes.

Table 6: Summary of the LBL processing stage

program	Description	`input(s)`	`output(s)`
lbldrv	Generate RFM driver tables	`[mwfile]`	`rfm_[mwlabel]_lbl_[atm].drv`
rfm	Generate reference spectra	`rfm*drv`	`rad_[mwlabel][alt].asc_lbl_[atm]`
	Upon successful completion:	`rad*[atm]`	*saved for TAB & SVD stages*

`[atm]` is `min`, `max` or `std`; `[alt]` is tangent altitude [m] (e.g., `08000`); `[gas]` is the species (e.g., `clono2`)

2.2.3 Outputs

1. Reference spectra `rad_[mwlabel][alt].asc_lbl_[atm]` (deleted at the end of the SVD stage)

2.3 TAB: Create Uncompressed LUTs

2.3.1 Overview

Uncompressed absorption coefficient tables $k(\nu, p, T)$ are generated for each absorber, and stored as binary files to save space and access time. The RFM is then used to simulate MW spectra using these tables and the results compared with the line-by-line reference spectra.

If the initial calculation already produces differences greater than NESR/30, the processing will cease and user intervention is required. See Appendix A.

If the initial maximum difference is less than NESR/30, the (p, T) axes for each table are progressively undersampled to establish the minimum number of points required along each axis. A total of 3 iterations are performed. Starting with the least significant absorber, the T then p axes are undersampled by an integer factor (performed within the RFM, using the original full binary file each time). If the calculated spectra differ from the reference LBL spectra by less than the designated maximum (which increases with each iteration) the reduced spacing is kept. Upon successful completion, the original binary LUTs are rewritten as ASCII files with reduced p, T axes.

A control file records the results of each iteration, and this also serves as a diagnostic file.

2.3.2 Processing

Table 7 summarises the processing within the TAB stage. Processing one microwindow takes several hours and is the most time-consuming processing stage. This is also the stage that drives the disk storage requirements.

Table 7: Summary of the TAB processing stage

program	Description	`input(s)`	`output(s)`
tabini	Generate control file and	`[mwfile]`	`[mwlabel].tab`
	RFM driver files for LUTs		`rfm_[mwlabel]_mak[gas].drv`
rfm	Generate uncompressed binary LUTs	`rfm*drv`	`tab_[mwlabel][gas].bin_big`
Iterate from here until p, T axes fully optimised			
tabdrv	Generate RFM driver files for spectra	`[mwlabel].tab`	`rfm_[mwlabel]_tab_[atm].drv`
rfm	Generate spectra using subsampled tables	`rfm*drv`	`rad_[mwlabel][alt].asc_tab_[atm]`
tabrad	Analyse spectra and modify control file	`rad*[atm]`	`[mwlabel].tab`
End iteration loop			
tabfin	Resample LUTs with reduced p, T axes	`tab*big`	`tab_[mwlabel][gas].asc`
	Upon successful completion:	`[mwlabel].tab`	*moved to* `../dia/`
		`tab*asc`	*saved for SVD stage*

`[atm]` is `min`, `max` or `std`; `[alt]` is tangent altitude [m] (e.g., `08000`); `[gas]` is the species (e.g., `clono2`)

2.3.3 Outputs

1. Control/diagnostics file saved in `../dia/[mwlabel].tab`
2. Uncompressed LUTs with optimised (p, T) axes, saved as `tab_[mwlabel][gas].asc`

2.3.4 Diagnostics

An example of a tabulation diagnostics file `[mwlabel].tab` is given in Table 8. This file summarises the optimisation of the (p,T) axes for the uncompressed LUTs for each microwindow absorber, one record corresponding to each attempt to reduce either the p or T axis of one absorber by an integer factor.

Table 8: Example of a TAB diagnostics file `[mwlabel].tab`

```
! Diagnostics for TAB optimisation. MODE=000
! co2    o3     h2o    c2h2   hcn    ch3cl  n2o    no2
PT__0001  703.300  706.300    8   53 53.39  8
 DiflimDifmax Ht Wn At It Pa  2   3   1  26  23  24   4  10
* 1.005 0.618 32  7  2  1 16 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
* 1.005 0.618 32  7  2  1 16 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
* 1.005 0.618 32  7  2  1 15 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0
[30 records omitted]
  1.780 1.785 26 25  3  3  4 1 1 1 4 5 3 0 0 0 0 0 0 0 0 0 0
  1.780 1.924 26 83  1  3  3 1 1 2 3 5 3 0 0 0 0 0 0 0 0 0 0
  1.780 2.678 26 84  1  3  2 1 2 1 3 5 3 0 0 0 0 0 0 0 0 0 0
* 1.780 1.701 32  7  2  3  1 2 1 1 3 5 3 0 0 0 0 0 0 0 0 0 0
  1.780 2.892 32  7  2  3  1 3 1 1 3 5 3 0 0 0 0 0 0 0 0 0 0
  1.780 0.000  0  0  0  4 15 2 1 1 3 5 3 0 0 0 0 0 0 0 0 0 0
I-TABFIN: Final max.difference = NESR *  0.032
```

Record#1 Header. Mode 000 is the default

Record#2 List of absorbers in decreasing order of significance

Record#3 Microwindow label `PT__0001`, the spectral boundaries `703.300 706.300` [cm^{-1}], the tangent altitude boundaries `8 53` [km], the NESR `53.39` [$\mathrm{nW/(cm^2\ sr\ cm^{-1})}$] and the number of absorbers `8`.

Record#4 Headers for remaining records. Numbers are the HITRAN/RFM indices of the absorbers listed in Record#2.

Record#5 etc Results of tests with reduced p,T axes. Asterisk `*` indicates an acceptable result, `Diflim` and `Difmax` are the maximum acceptable and observed differences; the following 3 columns locate the maximum radiance difference: `Ht` is the altitude [km], `Wn` the spectral grid point# within the MW, and `At` the atmosphere (1=min, 2=std, 3=max); `It` is the iteration number, `Pa` is the parameter being tested (i.e., which of following columns is modified); the following pairs of numbers are the current p and T axis increments (species identified by the HITRAN/RFM index at the top of the columns). Note that `0` corresponds to removing the entire axis and using a single, mid-point value. This is often acceptable for the minor absorbers, so is tested first.

Last Record Indicates the final radiance error reached, as a factor of the NESR

2.4 SVD: Singular Value Decomposition

2.4.1 Overview

Performs SVD on look-up tables (with previously optimised (p, T) axes) to create compressed LUTs with optimised number of singular values.

This starts by creating compressed LUTs containing 30 singular values for each absorber (or fewer if the number of (p, T) or wavenumber axis points is smaller than 30). It is assumed that 30 S.V.'s is more than required for any situation. These are used by the RFM to calculate spectra for all altitudes/atmospheres and compared with the reference line-by-line spectra. The number of S.V.'s is progressively reduced until the maximum radiance difference reaches the allocated limit. During this process the compressed LUTs containing 30 S.V.'s remain unchanged while the RFM driver tables are used to determine the reduced number of singular values that are actually used.

A total of 3 iterations are performed, each starting with the weakest absorber and progressing through to the strongest. Each iteration the limit on the maximum radiance difference is increased, reaching NESR/20 on the final iteration. Also, to save time, the number of S.V.'s is reduced by factors of 2 on the first iteration, factors of 1.5 on the second iteration, and by integer steps on the last iteration.

A control file records the results of each iteration, and this also serves as a diagnostic file.

Upon successful completion, the compressed LUTs are rewritten with the reduced number of singular values, renamed to conform to the MIPAS convention[1], and transferred to the `/CS` directory.

2.4.2 Processing

Table 9 summarises the processing within the SVD stage. Processing one microwindow takes several hours, although generally only half the time of the TAB stage.

Table 9: Summary of the SVD processing stage

program	Description	`input(s)`	`output(s)`
svdini	Create control file and	`[mwfile]`	`[mwlabel].svd`
	create compressed LUTs (30SV)	`../tab/tab*asc`	`lut_[mwlabel][gas].asc_big`
Iterate from here until No.S.V.s fully optimised			
svddrv	Generate RFM driver files	`[mwlabel].svd`	`rfm_[mwlabel]_lut_[atm].drv`
rfm	Generate spectra using tables	`rad*[atm]`	*svdrad*
svdrad	Analyse spectra and modify control file	`rad*[atm]`	`[mwlabel].svd`
End iteration loop			
svdfin	Resample tables with reduced No.S.V.s	`lut*asc`	`CS_[mwlabel]_[id].DAT`
svdimk	Write No.S.V.s to MW database	`CS*DAT`	`[mwfile]`
	Upon successful completion:	`../lbl/rad*[atm]`	*deleted*
		`../tab/tab*.asc`	*deleted*
		`[mwlabel].svd`	*moved to* `../dia/`
		`CS*DAT`	*moved to* `../CS/`

`[atm]` is `min`, `max` or `std`; `[alt]` is tangent altitude [m] (e.g., `08000`); `[gas]` is the species (e.g., `clono2`)
`[id]` is the HITRAN/RFM index for the molecule (e.g., `01` for `[gas]=h2o`)

2.4.3 Outputs

1. Control/diagnostics file saved in `../dia/[mwlabel].svd`

2. SVD-Compressed LUTs saved as `../CS/CS_[mwlabel]_[id].DAT`

3. Modified microwindow database (see Table 16).

2.4.4 Diagnostics

An example of an SVD compression diagnostics file `[mwlabel].svd` is given in Table 10. This file summarises the optimisation of the number of singular values for the compressed LUTs for each microwindow absorber, one record corresponding to each attempt to reduce the number of S.V.'s for one absorber LUT.

Table 10: Example of a SVD diagnostics file `[mwlabel].svd`

```
! Diagnostics for SVD optimisation
! co2    o3
PT__0002  741.550  741.775   53   71 30.54  2
 DiflimDifmax Ht Wn At It Pa  2   3
* 1.235 0.871 53  3  1  1  2 30  30
* 1.235 0.853 53  3  1  1  2 30  15
* 1.235 0.878 53  3  1  1  2 30   7
  1.235 2.572 53  2  3  1  2 30   3
* 1.235 0.856 53  3  1  1  1 15   7
  1.235 3.989 56  7  3  1  1  7   7
* 1.599 0.825 53  3  1  2  2 15   4
  1.59928.490 53 10  3  2  2 15   2
  1.599 1.693 53  3  1  2  1 10   4
  1.963 2.621 53  2  3  3  2 15   3
* 1.963 0.896 53  3  1  3  1 14   4
* 1.963 0.989 53  3  1  3  1 13   4
* 1.963 1.045 62  7  2  3  1 12   4
* 1.963 1.021 62  7  2  3  1 11   4
* 1.963 1.693 53  3  1  3  1 10   4
* 1.963 1.267 53  3  1  3  1  9   4
* 1.963 1.708 53  3  1  3  1  8   4
  1.963 0.000  0  0  0  4  2  8   3
I-SVDFIN: Final max.difference = NESR *  0.056
```

Format is similar to the TAB diagnostics (Table 8) except that there is only a single parameter for each absorber, the number of singular values.

2.5 GRD: Derive Irregular Grid

2.5.1 Overview

This starts by using the RFM to calculate spectra on fine grid (0.0005 cm^{-1}) for all MW tangent altitudes and min/std/max atmospheres, using both line-by-line calculations and the compressed LUTs derived in the SVD stage.

The AILS-convolutions of these spectra are compared on the 0.025 cm^{-1} spectral grid, and fine grid points are progressively replaced by interpolated values from adjacent points so as to keep the maximum difference as small as possible.

Because there is some initial error from the use of compressed LUTs compared to line-by-line calculations, the maximum radiance difference is reduced to start with as the algorithm finds compensating spectral interpolation errors of opposite sign, but eventually the maximum difference starts to increase again. This continues until replacing any point will lead to a maximum difference greater than NESR/10.

Upon successful completion the irregular grid is written into the microwindow database, and also produced as a separate file.

2.5.2 Processing

Table 11 summarises the processing within the GRD stage. Processing one microwindow typically takes 2 hours.

Table 11: Summary of the GRD processing stage

program	Description	`input(s)`	`output(s)`

grddrv	Generate IGTOOL driver file and RFM driver files	`[mwfile]`	`[mwlabel]_igtool.igdrv`
			`rfm_[mwlabel]_lut_[atm].drv`
			`rfm_[mwlabel]_ref_[atm].drv`

rfm	Generate spectra on fine grid	rfm*drv	`rad_[mwlabel][alt].asc_lut_[atm]` `rad_[mwlabel][alt].asc_ref_[atm]`
igtool	Generate Irregular Grid	`rad*[atm]`	`[mwlabel].grd`
grdimk	Write irreg.grid to MW database	`[mwlabel].grd`	`[mwfile]`
	Upon successful completion:	`[mwlabel].grd`	*saved for ERR stage & copied to* `../dia/`

`[atm]` is `min`, `max` or `std`; `[alt]` is tangent altitude [m] (e.g., `08000`)

2.5.3 Outputs

1. Irregular grid as separate file saved in `[mwlabel].grd` with copy in `../dia/[mwlabel].grd`
2. Modified microwindow database (see Table 16).

2.6 ERR: Update Database Quantifiers

2.6.1 Overview

The use of LUTs and Irregular Grids leads to errors in the forward model. The processing limits these to a maximum of NESR/10 at any tangent altitude for any of the three atmospheres, but the microwindow database contains 'quantifiers' which represent various errors transformed into an equivalent retrieval error. This processing stage calculates the 'model error' associated with the LUT+Irreg.Grid spectral errors and adds these to the database.

For this, the difference spectra (line-by-line) - (LUT+irreg.grid) and Jacobian spectra for the various retrieved parameters are calculated for each tangent altitude and multiplied as a dot product.

2.6.2 Processing

Table 12 summarises the processing within the ERR stage. Processing one microwindow takes around 15 minutes.

Table 12: Summary of the ERR processing stage

program	Description	`input(s)`	`output(s)`
errdrv	Generate RFM driver files	`[mwfile]`	`rfm_[mwlabel][alt]_lbl.drv` `rfm_[mwlabel]_lutgrd.drv`
rfm	Generate LBL spectra,	`rfm*drv`	`rad_[mwlabel][alt].asc_lbl`
	Jacobian spectra and	`../CS/CS*DAT`	`rad_[mwlabel][alt]_[gas][alt].asc_lbl`
	spectra using LUT+Grd	`../grd/[mwlabel].grd`	`rad_[mwlabel][alt].asc_lutgrd`
errimk	Error Analysis and	`rad*lbl`	
	update MW Database	`rad*lutgrd`	`[mwfile]`
	Upon successful completion:	`../grd/[mwlabel].grd`	*deleted*

`[alt]` is tangent altitude [m] (e.g., `08000`); `[gas]` is the species (e.g., `clono2`)

2.6.3 Outputs

1. Modified microwindow database (see Table 16).

A User-controlled TAB Processing

The program `tabini` attempts to generate reasonable values for the range and increments of the (p, T) axes for the initial uncompressed Look-Up Tables. However, even without any axis reduction, it is possible that the interpolation errors for these full-size tables already exceeds the NESR/30 limits, in which case it is necessary for the user to determine different initial axes. This can all be controlled through the same shell script `tab.sh` but providing an additional (3rd) parameter. A typical procedure would be as follows.

A.1 Identifying Failure

The following symptoms identify the situation where the initial interpolation error is too large:

- `tab.sh` Processing stops with the following messages printed to the terminal

  ```
  I-TABFIN: No tabulation possible
  W-tab.sh: Incomplete
  ```

- The original binary LUTs `tab*bin` will remain (and will be required for further use).
- No ASCII files `tab*.asc` are produced.
- The diagnostics file remains in the `/tab` directory and contains a single iteration record, as illustrated in Table 13.

Table 13: Example of a TAB diagnostics file `[mwlabel].tab` when the initial interpolation error exceeds NESR/30

```
! Diagnostics for TAB optimisation. MODE=000
! co2    o3     h2o    c2h2   hcn    ch3cl  n2o    no2
PT__0001  703.300  706.300    8   53 53.39  8
 DiflimDifmax Ht Wn At It Pa  2   3   1  26  23  24   4  10
  1.78010.470 41  1  3  4 16 1 1 1 1 1 1 1 1 1 1 1 1 1 1
  1.780 0.000  0  0  0  4 16 1 1 1 1 1 1 1 1 1 1 1 1 1 2
F-TABFIN: Final max.difference = NESR *  0.196
```

The first iteration row shows an initial interpolation error `10.470` which is much larger than NESR/30 `1.780`, therefore it is necessary to adjust the p, T dimensions of one or more of the component absorption tables.

A.2 'Force' Option

If the NESR/30 criterion is only marginally exceeded (say, up to 0.4 NESR), the TAB stage can be forced to accept a larger tolerance and proceed as normal:

```
%../shell/tab.sh [mwlabel] [mwfile] FORCE
```

Processing under such conditions is identified with `MODE=001` in the diagnostics header.

A.3 Identifying Problem Table(s)

If the initial error is too large to use the 'Force' option, it is then necessary to determine which absorber table (usually only one) is causing the interpolation error.

First, rerun the comparison using just the most significant absorber. This is achieved by adding a third parameter '100'.

```
%../shell/tab.sh [mwlabel] [mwfile] 100
```

This will generate new reference spectra using line-by-line calculations for only the most significant absorber, which are then compared to spectra calculated using the look-up table only for the same absorber. Whatever the resulting difference, only a single run through is performed (no iteration of table dimensions) and a diagnostics file `[mwlabel].tab_M100` is generated, an example is shown in Table 14.

Table 14: Example of a TAB diagnostics file `[mwlabel].tab_M100` for single absorber analysis.

```
! Diagnostics for TAB optimisation. MODE=100
! co2
PT__0001  703.300  706.300    8   53 53.39  1
 DiflimDifmax Ht Wn At It Pa  2
*11.17210.460 41  1  3  4  2 1 1
 11.172 0.000  0  0  0  4  2 1 2
```

The similarity of the `Difmax` value `10.460` to the corresponding value in Table 13 (`10.470`) suggests that this (CO_2) is the absorber table creating the large interpolation error, so manual setting of p, T dimensions is required for this table.

If this shows that the interpolation error is large (i.e., comparable with the initial difference originally contained in `[mwlabel].tab`, as is the case in Tables 13 and 14), this is the problematic table so proceed to the next stage. If the difference is small, it is necessary to add the next most significant absorber:

```
%../shell/tab.sh [mwlabel] [mwfile] 200
```

If the resulting file `[mwlabel].tab_M200` shows a large difference, then the problem is with the second absorber table, if not, repeat using 3 absorbers (`300`), and so on until the problematic table is determined.

A.4 Manually setting table dimensions

Having established that absorber#`n` (numbered in order of decreasing significance) is causing the large initial interpolation error, the table must be regenerated with different (p, T) axes, overwriting the existing `tab*bin` file for this absorber. This is performed by

```
%../shell/tab.sh [mwlabel] [mwfile] [n][i][j]
```

where `n` is the absorber number, `i` is the low-pressure limit (values i=(1,2,3) corresponding to (1.0,0.1,0.01) mb), `j` is the temperature increment in integer degrees Kelvin. The pressure axis spacing is automatically adjusted to be as large as possible within a certain maximum array size, so if the temperature interval is

increased, and/or the low-pressure limit is increased (i.e., decreased in altitude), then the $\ln p$ axis increments become smaller.

Again, only a single run is performed with the output difference written to file `[mwlabel].tab_M[n][i][j]`. This may have to be repeated for different values `i,j` until the interpolation error for `n` absorbers is satisfactory, at which point proceed to the next stage.

The diagnostics file provides information which may be useful in establishing which parameter should be adjusted. Some hints are:

- If the maximum error occurs at medium or high altitude, the problem may be that the low pressure limit is too large, in which case try `nij=n34` (particularly if the absorber is CO_2 or other species with strong features at high altitudes).
- If the maximum error occurs at low altitude, the problem is likely to be that the pressure axis resolution is too coarse, in which case try increasing the low pressure limit and/or increasing the temperature spacing `nij=n15`.
- If all else fails, the `ij` parameter space around `ij=24` has to be explored until the values that give a minimum difference are found.

In the example shown (Table 13) the maximum error occurs at 41 km and the gas is CO_2, so rerunning with Mode 134 gives the diagnostics file shown in Table 15.

Table 15: Example of a TAB diagnostics file `[mwlabel].tab_M134` for single absorber with manually adjusted dimensions.

```
! Diagnostics for TAB optimisation. MODE=134
! co2
PT__0001  703.300  706.300    8   53 53.39  1
 DiflimDifmax Ht Wn At It Pa  2
* 2.153 0.659 32  7  2  4  2 1 1
  2.153 0.000  0  0  0  4  2 1 2
```

Compared to the single absorber radiance using the initial, automatically generated table dimensions (Table 14), the maximum radiance difference `0.659` is now within NESR/30 so TAB processing for all absorbers can probably be resumed using this new table.

A.5 Rerunning with Manually-set Table Dimensions

Having created a new absorber table with manually set-axes that gives a suitably reduced interpolation error, the iterative TAB procedure can now be restarted for all absorbers by using the `FORCE` option

```
%../shell/tab.sh [mwlabel] [mwfile] FORCE
```

Note that the `FORCE` argument is required to avoid overwriting the existing binary tables, including that with the manually set dimensions, but, unlike the `RESUME` argument, also ensures a new `[mwlabel].tab` file is generated.

Upon completion, the diagnostic file `[mwlabel].tab` (transferred to `../dia`) should be examined to ensure that the maximum difference is satisfactory — there is a small probability that a subsequent absorber table may also have an unacceptably large interpolation error, in which case the procedure must be repeated optimising that particular table as well.

B Microwindow Database Modifications

The Microwindow Database file[1] contains all the information required to describe individual microwindows. This file serves both as input to the processing suite and is also modified by the processing (Table 16) to contain an updated absorber list, the irregular grid and additional database quantifiers.

These modifications are repeatable, i.e., it is possible to rerun the processing on a modified database file and reproduce the same results.

The microwindow file is always supplied by the user as an argument to each processing stage, but it is suggested that the files to be modified are placed in the directory `/MW`.

Table 16: Microwindow Database Modifications

Stage	Parameter	Record	Description
ABS	NGAS	m.3	Number of gases to be included in forward model
	GAS	m.5.n	Gases to be considered
SVD	NSV	m.3	Number of singular values
GRD	NREC_GRD	m.3.1.0	Number of grid records following
	NTOT_GRD	m.3.1.0	Total number of fine grid points
	NUSE_GRD	m.3.1.0	Total number of selected fine grid points
	WNLO_GRD	m.3.1.0	Lower wavenumber of grid
	WNHI_GRD	m.3.1.0	Upper wavenumber of grid
	INT_GRD	m.3.1.0	Interpolation Scheme (1=linear)
	VERS_GRD	m.3.1.0	Version of grid generation algorithm
	VLUT_GRD	m.3.1.0	Version of LUTs for which grid is optimised
	FINE_GRD	m.3.1.n	Fine grid of microwindow in ASCII coding
ERR	NMOD	m.6.J.2	Number of model errors
	SOURCE_MOD	m.6.J.6.n	Identifier of model error
	ERROR_MOD	m.6.J.6.n	Model error of retrieval quantity I
	ERROR_MOD_TOT	m.6.J.6.k	Total model error of retrieval quantity I
	COV_TOT	m.7.J.7.n	Total retrieval error covariance matrix

References

[1] ASCII Input Data Interface Control Document (DD 53), PO-IF-DOG-GS-0002

[2] `http://www.caam.rice.edu/software/ARPACK/` (1996 version)

[3] Reference Forward Model, `http://www.atm.ox.ac.uk/RFM/`