Technical Note on:

High Level Software Architecture and Retrieval Modules Interfaces

3 March, 1998

Issue: 4
Revision A
(Compliant with ORM_ABC Version 1.1)

Delivery of the study:
"Development of an Optimised Algorithm for Routine P, T and VMR Retrieval from MIPAS Limb Emission Spectra"

Prepared by:

Name	Institute
B.Carli	IROE-CNR
A.Gignoli	IROE-CNR
P.Raspollini	IROE-CNR
M.Ridolfi	IROE-CNR

Approved by:

Name	Institute
M.Carlotti	University of Bologna

TABLE OF CONTENTS

1. Introduction

MIPAS (Michelson Interferometer for Passive Atmospheric Sounding) is an ESA developed instrument to be operated on Board ENVISAT-1 as part of the first Polar Orbit Earth Observation Mission program (POEM-1). MIPAS will perform limb sounding observations of the atmospheric emission spectrum in the middle infrared region. Concentration profiles of numerous trace gases can be derived from MIPAS observed spectra.

According to the current baseline ESA data processing will routinely retrieve from MIPAS measurements altitude profiles of atmospheric pressure and temperature (p,T), and of volume mixing ratio (VMR) of five high priority species (O_3, H_2O, HNO_3, CH_4 and N_2O). The retrieval of these parameters from calibrated spectra (Level 1b data) is performed by the Level 2 processor.

Level 2 processing is expected to be a critical part of the Payload Data Segment (PDS) because of both the long computing time that may be required and the need for a validated algorithm capable of producing accurate and reliable results.

The study for the "Development of an Optimised Algorithm for Routine P, T and VMR Retrieval from MIPAS Limb Emission Spectra" is meant to provide a scheme for Level 2 analysis, suitable for implementation in ENVISAT PDS and optimised for the requirements of speed and accuracy. The result of the study will be used by industry as an input for the development of the industrial prototype of Level 2 code.

In this document the high level architecture and interfaces of the scientific software are described following the guidelines provided by ESA.

1.1 Purpose of document

The purpose of this document is to define the high level architecture of Level 2 scientific processor and to describe the format of its I/O files.

1.2 Applicable and reference documents

The applicable and reference documents of the present technical note are listed below.

No	Document	Issue	Title
Applicable documents:			
AD 1	PO-TN-BOM-GS-0010	1	MIPAS Input/Output Data Definition
AD 2	PO-RS-ESA-GS-0177	2	MIPAS Level 2 Processing Input/Output Data Definition
AD3	PF-TN-ESA-GS-0009	3.1	ENVISAT Payload to Target Parameters Calculation Software Interface and Installation Guide
AD4	PPF-TN-ESA-GS-0006	3.1	ENVISAT Orbit Propagator
AD5	PO-TN-ESA-GS-0242	5.0	ENVISAT-1 Product Format Guidelines
AD6	TN-IROE-RSA9601	1	High level algorithm definition on physical and mathematical optimisations
AD7	TN-IROE-RSA9602	3	Software Architecture and Algorithms Definitions
AD8	TN-IROE-RSA9603	1	ORM Test Data and Procedure Definition
AD9	TN-IROE-RSA9604	1	OFM Test Data and Procedure Definition
Reference documents:			
RD1	PO-RS-DOG-GS-0001	1	MIPAS Level 2 Processor Prototyping, Software Requirements Document
RD2	PO-TN-OXF-GS-0010	-	Generation of Optimized Spectral Grids
RD3	PO-TN-OXF-GS-0011	-	Generation of compressed cross-section look-up tables for NRT MIPAS Retrievals

1.3 Acronyms

The acronyms which will be used in the present technical note are listed below:

AILS	Apodized Instrument Line Shape
FOV	Field Of View
HW	Half Width
ILS	Instrument Line Shape
MW	Microwindow
PDS	Payload Data Segment
UTC	Universal Time Coordinated
VCM	Variance Covariance Matrix
VMR	Volume Mixing Ratio
ZPD	Zero Path Difference
r.u.	Radiance Units: $nW / (cm^2 * sr * cm^{-1})$
LUT	Look-Up Table
ORM	Optimized Retrieval Model
OFM	Optimised Forward Model

1.4 Definitions and general information

The data sets contained in the files constituting the external interfaces of the scientific code are described using the same approach as in AD1.

The following definitions apply to this document:

Ancillary data
This term will be used to identify all data used at the input of level 2 processor which do not originate from the Level 1b input data and which will be updated only very unfrequently.
Example: Microwindows spectral intervals definition, look-up tables with various spectroscopic and geometric parameters, initial guess profile data, ….
Update rate: < once per several month.

Annotation data (Level 2 products only)
This term denotes data required for an exhaustive interpretation of a Level 2 data product. These data cover, in particular, output data of the Level 1b processor which are included in the input data from the Level 1b product but which are not processed in the Level 2 chain. Level 2 will also contribute with other annotation data.
Example: Instrument & processing parameters / settings, output parameters of spectral calibration, ILS retrieval parameters, ….
Update rate: > once per orbit/Level 2 product.

Auxiliary data
Auxiliary data are all data except the calibrated scene spectra, contained in Level 1b data, which are used by Level 2 processor.
Example: Tangent point geolocation data, referenced NESR assessment data, …
once per elevation scan sequence > Update rate > once per month.

Main Level 1b product
Calibrated spectra.
once per elevation scan sequence > Update rate > once per month.

External data
External data are all Level 2 processor input data not generated within the PDS and updated at medium rates (update rate between once per orbit and once per several months).
Example: meteorological forecast data,…
once per orbit > Update rate > once per several month.

Internal data
This term denotes all ground processor input and output data that will be generated and overwritten during the processing of a single Level 2 data product. These data will not be archived or disseminated routinely.
Example: Level 2 processor output data (profile data, synthetic spectra) of individual iteration cycles, residual spectra, …
Update rate: > once per orbit.

Scan/Sweep
A scan is a limb scanning sequence that includes a number of sweeps. A sweep is a spectral measurement of the MIPAS Fourier Transform Spectrometer.

Level 2 products
Data retrieved in level 2 processor with their errors.
Update rate: > each limb scanning sequence.

This general information apply to the present document:

Data / file size
The data (file) size will be expressed in bytes if not otherwise specified. Data are described according to the product convention defined in [AD5]
The sizes of the VCM's are computed without taking into account the symmetry of these matrices as indicated in AD2.

Programming language
The scientific code is written in FORTRAN and can be compiled on a SUN SPARCstation under Solaris 2.5 using the Fortran SPARCompiler 3.0.1 for Solaris.

1.5 Main changes relative to previous issue of the document

Since the document on 'Software Architecture and Algorthms Definitions' (AD7) has evolved and it is now exhaustive regarding the architecture of the ORM code, the part of this document refering to the description of the ORM modules architecture has been stopped at high level. The reader should therefore refer to AD7 for the detailed description of the algorithms and the architecture of the ORM code.

Issue 4 of the present document is compliant with Version 2.3 of the ORM code. Furthermore, compared with Issue 3a, in the present issue, three new entries have been added while describing the I/O data sets:

- File name: is the name of the file(s) in which the current data set is contained. However, as in the previous version of the document, the correspondence between data sets and file names is also summarised in the cross correlation matrix reported in Sect. 2.1.7.
- Typical file size: is the typical size of the files(s) containing the current data set. The size is expressed in bytes if not otherwise specified.
- Fortran code used to read / write the file: we have added at the end of the description of each data set, the source FORTRAN code that is used for reading / writing the corresponding file(s).

Revision A of Issue 4 of the present document is compliant with ORM_ABC Version 1.1. The main output files of both p,T and VMR retrievals have been modified in order to include qualifiers characterising continuum retrieved parameters (see description of files pt_out.dat and VMRxx_out.dat).

2. Level 2 scientific code

Level 2 scientific code consists of a self standing forward model named 'Optimised Forward Model (OFM)' and of an Optimized Retrieval Model (ORM).
The OFM is only used to simulate MIPAS observations to be provided as input to the ORM for testing the ORM code itself. The OFM will not be integrated in the Level 2 processing chain and therefore, a detailed description of its I/O files is not striclty required.
The task of the ORM is to retrieve altitude profiles of atmospheric pressure and temperature (p,T), and of volume mixing ratio (VMR) of the five high priority constituents (O_3, H_2O, HNO_3, CH_4 and N_2O) starting from a selection of Level 1b data products and some others ancillary and external data. Since the ORM code does not directly interface with Level 1b outputs, a detailed description of the I/O data files of this code is required in order to make possible the installation of the code itself in MIPAS Level 2 processing chain.

2.1 ORM Input/Output data sets

The external interfaces of the ORM code consist of several I/O data files. In this section all the data sets contained in the files are described using the same approach as in [AD1]. It is assumed that each data set is contained into a single file; the names of the files accessed by the scientific code are connected with the data set names as shown in the field named 'File name' and in the cross correlation table reported in Sect. 2.1.7. The complete Fortran source routines which allow to read / write the I/O files are reported after each data set description as well as in the appendices of the present document.
Furthermore, whenever the scientific code requires data that are not explicitly recorded in Level 1b products, the procedure to extract the related information from Level 1b data files is analysed.
The data sets having 'xx' as a suffix correspond to 5 files: one file for each VMR retrieval.
The data sets having 'PT(VMRxx)' as a suffix correspond to 6 files: one file for each retrieval.

2.1.1 Input data sets containing Level 1b data

Identifier: MP2-L1B-PT(VMRxx)

Name: Data set of the apodised microwindows for p,T and VMR retrievals.

File names: observ_pt(VMRxx).dat

Type: Input file ASCII

Description: in this input data set are present the calibrated spectra in correspondence of the microwindows which will be processed by p,T (VMRxx) retrieval module. The microwindows, contained in the spectra of Level 1b main-products and related to the single scan which is going to be processed, are selected, apodised and provided to the scientific code by an external software module. This data set contains also auxiliary data used by p,T (VMRxx) retrieval.

Data size: $16+((29+4*N_{ailsdp})*N_{mwPT(VMRxx)})+(77+((4*N_{spdp})+400))*N_{acq}$

Typical file size: ≈ 150 Kb

Throughput: size / each limb scanning sequence

Remarks:
$N_{mwPT(VMRxx)}$ is the number of microwindows needed for p,T (VMRxx) retrieval. The microwindows are selected using the logical matrix and the definition of the microwindows contained in the data set MP2-ANC-03.
In Level 2 scientific code it is assumed that the spectral data points are sampled in a standard frequency grid matching integer multiples of 0.025 cm^{-1}. For this reason the frequency step is not required from Level 1b data, but is a setting parameter (see MP2-ANC-01).

Data set: MP2-L1B-PT(VMRxx)				
Field name	**Format**	**Units**	**Size**	**Comments**
N. of sweeps into the scan which is going to be processed (N_{acq})	long (ul)	-	4	N_{acq} = total numb. of sweeps in the present scan
Total number of microwindows selected for this retrieval ($N_{mwPT(VMRxx)}$)	long (ul)	-	4	
Zero filling expressed as the ratio between measured and transformed interferogram	float	-	4	
Number of AILS data points (N_{ailsdp})	long (ul)	-	4	
Microwindow #1				
Microwindow label	character string	-	25	see note 1
Number of spectral data points ($N_{spdp}(1)$)	long (ul)	-	4	
AILS for microwindow #1	float vector	r.u.	4* N_{ailsdp}	see note 2
….	….	….	….	….
….	….	….	….	….
Microwindow # $N_{mwPT(VMRxx)}$				
Microwindow label	character string		25	see note 1
Number of spectral data points ($N_{spdp}(N_{mwPT(VMRxx)})$)	long (ul)	-	4	
AILS for microwindow #$N_{mwPT(VMRxx)}$	float vector	r.u.	4* N_{ailsdp}	see note 2
Sweep #1				
1) Auxiliary data				
Quality indicator (PCD)	character string	-	1	
N. of detected spikes in the present sweep	long (ul)	-	4	
Latitude, longitude and altitude defining geolocation of platform	three doubles	deg, deg, km	24	

(… Continued)				
Azimuth and elevation angles for the present sweep	two doubles	deg, deg.	16	
Latitude, longitude and altitude defining geolocation of tangent point	three doubles	deg, deg, km	24	
Local earth's radius curvature at the nadir of the tangent point of the present sweep	double	km	8	see note 3
Microwindow #1				
Spectral noise for this microwindow	float	r.u.	4	see note 4
Observed spectral data points (apodised microwindow)	float vector	r.u.	≈ 400, see AD1	see note 5
….	….	….	….	….
….	….	….	….	….
Microwindow # $N_{mwPT(VMRxx)}$				
Spectral noise for this microwindow	float	r.u.	4	see note 4
Observed spectral data points (apodised microwindow)	float vector	r.u.	≈ 400, see AD1	see note 5
The same structure of sweep #1 is repeated for sweeps #2 ... #N_{acq}				

Notes:

1. The same microwindow label as in MP2-ANC-03 shall be used. The microwindows are always supposed as sorted in the wavenumbers domain, starting from low wavenumbers.
2. AILS is the Apodised Instrument Line Shape equal to the ILS at the central frequency of the microwindow (derived from Level 1b file MP-C05 sect. 4.3.5 of AD1) convoluted with the apodisation vector given by the FFT of the vector contained in MP2-ANC-05. The frequency spacing of AILS is equal to that of the spectral data (interpolation in the fine grid is performed by the scientific code). The size of AILS is constant, since the length of the apodising vector is fixed. We foresee a size of 1024 (power of 2) +1 data points for each microwindow. It is still under investigation whether the calculation of AILS can be performed only at the centre of each MIPAS spectral band or a personalised AILS is needed for each microwindow (as assumed in the current approach).
3. This quantity is available as an output of TARGET (see AD3) but is not yet included in Level 1b products.
4. The noise in each microwindow is the standard deviation of the signal at the central frequency of the microwindow. The standard deviation at the required frequency is computed by linear

interpolation in the frequency domain, of the standard deviations contained in the Level 1b file MP-V01 (see sect. 4.4.1 of [AD1]).

5. The following comments apply to the apodised microwindows:
   - The spectral interval of each microwindow is defined in the ancillary data set: MP2-ANC-03 in such way that the start and end frequencies coincide with sampling points. Start and end frequencies are included in the selected spectral intervals.
   - The operational microwindows (i.e. the microwindows which have been extracted from Level 1b data and are going to be processed) are selected using the microwindow definition and the occupation matrix contained in MP2-ANC-03.
   - The apodisation process consists in the convolution of the measured spectra with the apodisation function defined in the data set: MP2-ANC-05. At the end of the process the width of the obtained microwindows shall match the width defined in MP2-ANC-03.

FORTRAN code used to read the file

```
            nf=11                   ! number of the file
            open(nf,file='./INP_FILES/observ_pt.dat',status='old')
            call skip_pt(nf)    ! skipping comment lines in the file
            read(nf,100) ilimb
* ilimb = number of sweeps belonging to the scan which is going to be processed (i5)
            call skip_pt(nf)
            read(nf,100) nselmw
* nselmw = total number of MW's selected for this retrieval
            call skip_pt(nf)
            read(nf,110) rzerof
* rzerof = zero filling expressed as the ratio between measured and
* transformed interferogram (f10.5)
            call skip_pt(nf)
            read(nf,100) nailsdp
* nailsdp = number of AILS data points (i5)
            do 10 k=1,nselmw   ! loop on microwindows
                  call skip_pt(nf)
                  read(nf,'(a25)') slab(k)
* slab(k) = label for MW k (a25)
                  call skip_pt(nf)
                  read(nf,100) nsam(k)
* nsam(k) = number of spectral data points of the k-th MW (i5)
                  call skip_pt(nf)
                  read(nf,120)(rails(l,k),l=1,nailsdp)
* rails = apodized instrument line shape for all selected MW's (8g10.4)
            10 continue                         ! end loop on microwindows
            do 20 j=1,ilimb             ! start loop on sweeps
                  call skip_pt(nf)
                  read(nf,'(a1)') squal(j)
* squal(j) = quality indicator of sweep j (a1)
                  call skip_pt(nf)
                  read(nf,100) nspikes(j)
* nspikes(j) = number of detected spikes in sweep j (i5)
                  call skip_pt(nf)
                  read(nf,110) dlatp(j),dlongp(j),daltp(j)
* dlatp(j), dlongp(j), daltp(j) = latitude, longitude, altitude defining the
*                 geolocation of platform at sweep j (3f10.5)
                  call skip_pt(nf)
                  read(nf,110) dazima(j),deleva(j)
* dazima(j), deleva(j) = azimuth and elevation angles of sweep j (2f10.5)
                  call skip_pt(nf)
                  read(nf,110) dlatt(j),dlongt(j),rztang(j)
* dlatt(j), dlongt(j), rztang(j) = latitude, longitude, altitude defining
*                 geolocation of tangent point of sweep j (3f10.5)
                  call skip_pt(nf)
                  read(nf,110) dlocer(j)
* dlocer(j) = local earth's radius curvature for sweep j (f10.5)
                  do 30 k=1,nselmw      ! start loop on microwindows
```

```
                call skip_pt(nf)
                read(nf,120) rnoise(k,j)
* rnoise(k,j) = noise for MW k in sweep j (g10.4)
                call skip_pt(nf)
                read(nf,120) (robs(l,j,k),l=1,nsam(k))
* robs = observed spectral data points for MW k of sweep j (8g10.4)
          30  continue                    ! end loop on microwindows
          20 continue          ! end loop on sweeps
      close (nf)
      100 format (i5)
      110 format (3F10.5)
      120 format (8g10.4)
      end
```

2.1.2 Ancillary data sets (input data sets)

Here we describe the ancillary data sets used as input of Level 2 scientific code.

Identifier: MP2-ANC-01

Name: Settings and switches ancillary data set for p,T retrieval module

File name: settings_pt.dat

Type: Input ASCII

Description: in this data set are listed settings and switches for p,T retrieval run.

Data size: 483

Typical file size: 3050

Throughput: Size / Each limb scanning sequence

Data set: MP2-ANC-01				
Field name	**Format**	**Units**	**Size**	**Comments**
Frequency step that simulates infinitesimal spectral resolution in forward simulations	double	cm^{-1}	8	this is the step of the so called 'fine-grid'
Frequency spacing between observed spectral data points	double	cm^{-1-}	8	standard Lev. 2 frequency grid = 0.025 cm^{-1}
Definition of the output level	character string	-	25	see note 1 (not used yet)
Vector of convergence criteria to stop iterations	float vector	-	12	see note 2
maximum number of allowed Gauss-Newton macro-iterations	ul	-	4	
maximum number of allowed Marquardt micro-iterations	ul	-	4	
Switch for the use of cross sections look up tables	logical	-	4	
Switch for the combination of results with external information	logical vector	-	20	
Maximum allowed T variation between levels (low altitudes)	float	K	4	see note 3
Maximum variation allowed of T variation between levels (high altitudes)	float	K	4	see note 3
Altitude where the temperature thresholds are exchanged	float	km	4	
Maximum allowed HW variation of a reference line between the levels used for modelling the atmosphere	float	-	4	see note 4
Max. thickness of the layers used for modelling the atmosphere	float	km	4	
Reduction factor to the max. thickness of the layers to produce thinner layers stepwise	float	-	4	
Central frequency of the reference line chosen for testing the pressure levels used for modelling the	double	cm^{-1}	8	

atmosphere				
Half-width of the reference line chosen for testing the pressure levels used for modelling the atmosphere	float	cm^{-1}	4	
Exponent for temp. dependence of half width in the reference line chosen for testing the pressure levels used for modelling the atmosphere	float	-	4	
Molecular weight of the reference gas chosen for testing the pressure levels used for modelling the atmosphere	float	g/mol	4	
Boundary of atmosphere	float	km	4	
Convergence criterion for Curtis-Godson integrals	double	-	4	
Actual number of extra paths	long (ul)	-	4	see note 8
Switch for the calculation of CO_2 χ factor	long (sl)	-	4	see note 5
Switch for the interpolation on the cross sections	long (sl)	-	4	see note 6
Greater base of the trapezium representing the FOV function	float	km	4	
Half difference between the two bases of the trapezium representing the FOV function	float	km	4	
HITRAN code of the main molecule of the retrieval	long (ul)	-	4	
Logical vector that identifies the tangent altitudes which correspond to fitted parameters	logical vector	-	80	see note 7
Switch for fitting atmospheric continuum and offset (ifco)	ul	-	4	
Altitude above which continuum is not fitted	float	km	4	
Altitude above which continuum is forced to zero	float	km	4	
Relative distance between two MWs (with respect to the umbrella radius)	float	-	4	

below which the continuum is considered the same in the two MWs				
Initial value of Marquardt damping factor (λ)	float	-	4	
drop factor of λ	float	-	4	
boost factor of λ	float	-	4	
Switch to enable the use of irregular grids (if available)	logical	-	4	
Spare field for retrieval configuration	character string	-	150	

Notes:

1. This string should define the information which is asked to appear in the annotation output data set (MP2-ANN-PT), however this parameter is not used in the ORM_V2.3 (or earlier versions).
2. Convergence criteria for Gauss-Newton-Marquardt algorithm (see Sect. 4.2.3 of [AD6]).
3. Maximum allowed temperature variation between two neighbouring pressure levels used for modelling the atmosphere.
4. Maximum allowed line half-width variation between two neighbouring pressure levels used for modelling the atmosphere.
5. 0= No factor
   1=N_2/O_2 broadening
   2=N_2 broadening only
6. -1 = No interpolation, all cross sections
   0 = All those above the lowest geometry are interpolated
   1 = New calculation only of tangent layer
   2 = New calculation for tangent layer and one layer above
7. Logical vector produced as part of the procedure which determines the occupation matrix given in MP2-ANC-03-PT (this vector is not operational in the ORM vers. 2).
8. This is a technical variable used for the optimisation of the radiative transfer computation.

FORTRAN code used to read the file

```
      open(12,file='./INP_FILES/settings_pt.dat',status='old')
      nf=12
     call skip_pt(nf)
     read(nf,100) delta
* delta = frequency step that simulates infinitesimal spectral resolution
*        in forward simulations (f10.5)
     call skip_pt(nf)
     read(nf,100) dstep
* dstep = frequency spacing between observed spectral data points (f10.5)
     call skip_pt(nf)
     read(nf,'(a25)') sdol
* sdol = definition of the output level (a25)
     call skip_pt(nf)
     read(nf,100)(rconvc(j),j=1,3)
* rconvc = vector of convergence criteria to stop iterations (3f10.5)
* rconvc(1) = variation of chi**2 between two consecutive iterations
* rconvc(2) = variation between calculated chi**2 and linear chi**2
* rconvc(3) = largest variation of parameters between consecutive iterations
     call skip_pt(nf)
     read(nf,120) imxiterg
* imxiterg = maximum number of Gauss-Newton macro-iterations
```

```
      call skip_pt(nf)
      read(nf,120) imxiterm
* imxiterm = maximum number of Marquard micro-iterations
      call skip_pt(nf)
      read(nf,110) lookupc
*  lextinf1 = switch for using a-priori info on LOS
*  lifenf   = switch for using a-priori info on LOS after the retrieval
*  lextinf2 = switch for using a-priori info on atm. continuum
*  lextinf3 = switch for using a-priori info on instr. offset
*  lextinf4 = switch for using a-priori info on temperature
      call skip_pt(nf)
      read(nf,'(5l2)') lextinf1,lifend,lextinf2,lextinf3,lextinf4
      call skip_pt(nf)
      read(nf,100) rmaxtv1,rmaxtv2,rzt12
* parameters_pt.incor the layering of the atmosphere (3f10.5)
* rmaxtv1 = max. allowed T variation (K) between levels: 0 < altitude < rzt12
* rmaxtv2 = max. allowed T variation (K) between levels: rzt12<altitude<rulatm
* rzt12 = altitude (km) where the thresholds rmaxtv1 and rmaxtv2 are exchanged
      call skip_pt(nf)
      read(nf,100) rhwvar
* rhwvar = max half-width change allowed between levels for ref. transiton (f10.5)
      call skip_pt(nf)
      read(nf,100) rincz
* rincz = max thickness of the layers used for modelling the atmosphere (f10.5)
      call skip_pt(nf)
      read(nf,100) redfact
* redfact = reduction factor for rincz to produce thinner layers
      call skip_pt(nf)
      read(nf,100) dsigm0,rhw0ref,rexphref,rwmolref
* data relative to the transition that is used for layering criterion (4f10.5)
* dsigm0 = frequency position
* rhw0ref = half-width at the reference T and P
* rexphref = T dependence of half-width
* rwmolref = molecular weight
      call skip_pt(nf)
      read(nf,100) rulatm
* rulatm = boundary of the atmosphere (km) (f10.5)
      call skip_pt(nf)
      read(nf,100) deps
* deps = convergence criterion for the C.G. integrals (f10.5)
      call skip_pt(nf)
      read(nf,120) iept
* iept = actual number of extra paths (i5)
      call skip_pt(nf)
      read(nf,120) nswco2
* nswco2 = switch for the calculation of the CO2 chi factor (i5)
*       =0 no factor, =1 n2/o2 broadening, =2 n2 broadening only
* NOTE this switch is no longer active but is maintained in the case of future
*      applications
      call skip_pt(nf)
      read(nf,120) ninterpol
*   ninterpol = switch for the interpolation on the cross-sections (i5)
*             = -1: no interpolation, all cross-sections recalculated
*             =  0: all those above the lowest geometry are interpolated
*             =  1: new calculation only of tangent-layer
*             =  2: new calculation for tangent-layer and the one above
      call skip_pt(nf)
      read(nf,100) rbase,rsl
* parameters_pt.incor the FOV convolution (2f10.5)
* rbase = greater base of the trapetium FOV-function (km)
* rsl   = half difference between bases of the trapetium FOV-function (km)
      call skip_pt(nf)
      read(nf,120)imaingas
* imaingas = HITRAN code of the main molecule of the retrieval (i5)
      call skip_pt(nf)
      read(nf,110)(lfit(j),j=1,ilimb)
* lfit(j) = logical vector that identifies the tangent altitudes which
* correspond to fitted parameters (40l2)

* Switch for fitting atmospheric continuum and offset (ifco):
```

```
*      ifco = 2 -->    p,T, continuum and offset are fitted
*      ifco = 1 -->    p,T and continuum are fitted
*      ifco = 0 -->    only p,T are fitted
       call skip_pt(nf)
       read(nf,*) ifco
* Upper continuum limit (rucl): atmospheric continuum is not fitted at the
* sweeps having tangent altitude > rucl. Units: km
       call skip_pt(nf)
       read(nf,*) rucl
* Altitude above which the continuum is set = 0
       call skip_pt(nf)
       read(nf,*) rzc0
* Max. relative distance between MWs to be considered
* as having the same continuum:
       call skip_pt(nf)
       read(nf,*) rperc
* Parameters controlling the evolution of Marquardt damping factor:
*      rlambdain = initial value of rlambda
*      rlambdadiv = factor used to decrease rlambda at each Gauss iteration
*      rlambdamul = factor useed to increase rlambda at each Marquardt iteration
       call skip_pt(nf)
       read(nf,*) rlambdain, rlambdadiv, rlambdamul
       call skip_vmr(nf)
       read(nf,*)lirrgrid
       call skip_pt(nf)
       read(nf,'(a150)') spare
* spare = spare field for retrieval configuration
       drd=dstep/delta
       nrd=nint(drd)
       if(abs(drd-float(nrd)).gt.0.00001) then
        write(*,'(f12.6)')'ratio dstep/deltas not integer program stopped
      & nrd=',dstep/delta
        stop
       end if
  100 format(4f10.5)
  110 format(40l2)
  120 format(i5)
130   format(40l2)
       return
       end
```

Identifier: MP2-ANC-02xx

Name: Settings and switches ancillary data set for VMR retrieval module

File name: settings_(VMRxx).dat

Type: Input ASCII

Description: in this data set are listed settings and switches for VMRxx retrieval.

Data Size: 483

Typical file size: 3250

Throughput: Size / Each limb scanning sequence

Remarks:

Data set: MP2-ANC-02xx				
Field name	**Format**	**Units**	**Size**	**Comments**
Switch for enabling the use of previously retrieved p,T profiles	logical	-	4	
Switch for enabling the use of previously retrieved VMR profiles	logical	-	4	
Frequency step that simulates infinitesimal spectral resolution in forward simulations	double	cm^{-1}	8	
Frequency spacing between observed spectral data points	double	cm^{-1-}	8	standard Lev. 2 frequency grid = 0.025 cm^{-1}
Definition of the output level	character string	-	25	see note 1
Vector of convergence criteria to stop iterations	float vector	-	20	see note 2
maximum number of allowed Gauss-Newton macro-iterations	ul	-	4	
maximum number of allowed Marquardt micro-iterations	ul	-	4	
Switch for the use of cross sections look up tables	logical	-	4	
Switch for the combination of results with external information	logical vector	-	12	
Maximum allowed T variation between levels (low altitudes)	float	K	4	
Maximum variation allowed of T variation between levels (high altitudes)	float	K	4	
Altitude where the temperature thresholds are exchanged	float	km	4	
Maximum allowed HW variation of a reference line between the levels used for modelling the atmosphere	float	-	4	
Max. thickness of the layers used for modelling the atmosphere	float	km	4	
Reduction factor to the max. thickness of the layers to produce thinner layers stepwise	float	-	4	
Central frequency of the reference line chosen for testing the pressure levels used for modelling the atmosphere	double	cm^{-1}	8	
Half-width of the reference line chosen for testing the pressure levels used for modelling the atmosphere	float	cm^{-1}	4	
Exponent for temp. dependence of half width in the reference line chosen for testing the pressure levels used for modelling the atmosphere	float	-	4	

Molecular weight of the reference gas chosen for testing the pressure levels used for modelling the atmosphere	float	g/mol	4	
Boundary of atmosphere	float	km	4	
Convergence criterion for Curtis-Godson integrals	double	-	8	
Actual number of extra paths	long (ul)	-	4	see note 5
Switch for the calculation of CO_2 χ factor	long (sl)	-	4	
Switch for the interpolation on the cross sections	long (sl)	-	4	see note 3
Greater base of the trapezium representing the FOV function	float	km	4	
Half difference between the two bases of the trapezium representing the FOV function	float	km	4	
Hitran code of the gas whose VMR is going to be retrieved	long (ul)	-	4	
Logical vector that identifies the tangent altitudes which correspond to fitted parameters	logical vector	-	80	see note 4
Switch for fitting atmospheric continuum and offset (ifco)	ul	-	4	
Altitude above which continuum is not fitted	float	km	4	
Altitude above which continuum is forced to zero	float	km	4	
Relative distance between two MWs (with respect to the umbrella radius) below which the continuum is considered the same in the two MWs	float	-	4	
Initial value of Marquardt damping factor (λ)	float	-	4	
drop factor of λ	float	-	4	
boost factor of λ	float	-	4	
Switch for enabling the use of irregular grids (if available)	logical	-	4	
Spare field for retrieval configuration	character string	-	150	

Notes:

1. This string defines the information which is asked to appear in the output files.
2. Convergence criteria for Gauss-Newton-Marquardt routine.
3. -1 = No interpolation, all cross sections
   0 = All those above the lowest geometry are interpolated
   1 = New calculation only of tangent layer
   2 = New calculation for tangent layer and one layer above
4. Logical vector produced as part of the procedure which determines the occupation matrix given in MP2-ANC-03-VMRxx.

5. This is a technical variable used for the optimisation of the radiative transfer.

FORTRAN code used to read the file

```
      open(12,file='INP_FILES/settings_'//sgas(im)(1:j1)//'.dat', status='old')
      nf=12
      call skip_vmr(nf)
* lifptret = switch for the use of p,T retrieved data
      read(nf,*)lifptret
      call skip_vmr(nf)
* lifvmret = switch for the use of last VMR retrieved data
      read(nf,*)lifvmret
      call skip_vmr(nf)
      read(nf,100) delta
* delta = frequency step that simulates infinitesimal spectral resolution
*         in forward simulations (f10.5)
      call skip_vmr(nf)
      read(nf,100) dstep
* dstep = frequency spacing between observed spectral data points (f10.5)
      call skip_vmr(nf)
      read(nf,'(a25)') sdol
* sdol = definition of the output level (a25)
      call skip_vmr(nf)
      read(nf,100)(rconvc(j),j=1,3)
* rconvc = vector of convergence criteria to stop iterations (3f10.5)
* rconvc(1) = variation of chi**2 between two consecutive iterations
* rconvc(2) = variation between calculated chi**2 and linear chi**2
* rconvc(3) = largest variation of parameters between consecutive iterations
      call skip_vmr(nf)
      read(nf,120) imxiterg
* imxiterg = maximum number of Gauss-Newton macro-iterations
      call skip_vmr(nf)
      read(nf,120) imxiterm
* imxiterm = maximum number of Marquard micro-iterations
      call skip_vmr(nf)
      read(nf,110) lookupc
* lookupc = switch for the use of cross sections look-up tables (l1)
      call skip_vmr(nf)
      read(nf,'(3l2)') lextinf1,lextinf2,lextinf3
* lextinf = switches for the combination of results with external information (l1)
      call skip_vmr(nf)
      read(nf,100) rmaxtv1,rmaxtv2,rzt12
* parameters_vmr.incor the layering of the atmosphere (3f10.5)
* rmaxtv1 = max. allowed T variation (K) between levels: 0 < altitude < rzt12
* rmaxtv2 = max. allowed T variation (K) between levels: rzt12<altitude<rulatm
* rzt12 = altitude (km) where the thresholds rmaxtv1 and rmaxtv2 are exchanged
      call skip_vmr(nf)
      read(nf,100) rhwvar
* rhwvar = max half-width change allowed between levels for ref. transiton (f10.5)
      call skip_vmr(nf)
      read(nf,100) rincz
* rincz = max thickness of the layers used for modelling the atmosphere (f10.5)
      call skip_vmr(nf)
      read(nf,100) redfact
* redfact = reduction factor for rincz to produce thinner layers
      call skip_vmr(nf)
      read(nf,100) dsigm0,rhw0ref,rexphref,rwmolref
* data relative to the transition that is used for layering criterion (4f10.5)
* dsigm0 = frequency position
* rhw0ref = half-width at the reference T and P
* rexphref = T dependence of half-width
* rwmolref = molecular weight
      call skip_vmr(nf)
      read(nf,100) rulatm
* rulatm = boundary of the atmosphere (km) (f10.5)
      call skip_vmr(nf)
      read(nf,100) deps
* deps = convergence criterion for the C.G. integrals (f10.5)
      call skip_vmr(nf)
```

```
      read(nf,120) iept
* iept = actual number of extra paths (i5)
      call skip_vmr(nf)
      read(nf,120) nswco2
* nswco2 = switch for the calculation of the CO2 chi factor (i5)
*       =0 no factor, =1 n2/o2 broadening, =2 n2 broadening only
* NOTE this switch is no longer active but is maintained in the case of future
*      applications
      call skip_vmr(nf)
      read(nf,120) ninterpol
*   ninterpol = switch for the interpolation on the cross-sections (i5)
*            = -1: no interpolation, all cross-sections recalculated
*            =  0: all those above the lowest geometry are interpolated
*            =  1: new calculation only of tangent-layer
*            =  2: new calculation for tangent-layer and the one above
      call skip_vmr(nf)
      read(nf,100) rbase,rsl
* parameters_vmr.incor the FOV convolution (2f10.5)
* rbase = greater base of the trapetium FOV-function (km)
* rsl   = half difference between bases of the trapetium FOV-function (km)
      call skip_vmr(nf)
      read(nf,120)imaingas
* imaingas = HITRAN code of the main molecule of the retrieval (i5)
      call skip_vmr(nf)
      read(nf,110)(lfit(j),j=1,ilimb)
* lfit(j) = logical vector that identifies the tangent altitudes which
* correspond to fitted parameters (40l2)

* Switch for fitting atmospheric continuum and offset (ifco):
*      ifco = 2 -->    p,T, continuum and offset are fitted
*      ifco = 1 -->    p,T and continuum are fitted
*      ifco = 0 -->    only p,T are fitted
      call skip_pt(nf)
      read(nf,*) ifco
* Upper continuum limit (rucl): atmospheric continuum is not fitted at the
* sweeps having tangent altitude > rucl. Units: km
      call skip_pt(nf)
      read(nf,*) rucl
* Altitude above which the continuum is set = 0
      call skip_pt(nf)
      read(nf,*) rzc0
* Max.relative distance between MWs (with respect to the umbrella radius)
* so that the continuum is considered equal in the two MWs.
      call skip_pt(nf)
      read(nf,*) rperc
* Parameters controlling the evolution of Marquardt damping factor:
*     rlambdain = initial value of rlambda
*     rlambdadiv = factor used to decrease rlambda at each Gauss iteration
*     rlambdamul = factor useed to increase rlambda at each Marquardt iteration
      call skip_pt(nf)
      read(nf,*) rlambdain, rlambdadiv, rlambdamul
      call skip_vmr(nf)
      read(nf,*)lirrgrid
      call skip_vmr(nf)
      read(nf,'(a150)') spare
* spare = spare field for retrieval configuration
      drd=dstep/delta
      nrd=nint(drd)
      if(abs(drd-float(nrd)).gt.0.00001) then
       write(*,'(f12.6)')'ratio dstep/deltas not integer program stopped
     & nrd=',dstep/delta
       stop
      end if
  100 format(4f10.5)
  110 format(40l2)
  120 format(i5)
  130   format(40l2)
      return
      end
```

Identifier: MP2-ANC-03-PT(VMRxx)

Name: Definition of operational microwindows set ancillary data

File name: mwoccmat_pt(VMRxx).dat

Type: Input ASCII

Description: This data set defines a logical occupation matrix used for the microwindow selection of p,T (VMRxx) retrieval and a qualifier matrix used for applying constraints to the fitted atmospheric continuum (ubrella radii as a function of MW and altitude). The definition of the selected MWs and some related information are contained in this data set as well.
In Level 2 processing the occupation matrix will be selected from a standard set of occupation matrices or derived on-line, from the microwindow database, the available sweeps and the useful bands within each sweep according to a recipe which is still being developed.
The qualifier matrix is derived from the occupation matrix and the microwindow database by using a recipe TBD.

Size: $((14*N_{acq})+34)*N_{mwPT(VMRxx)}$

Typical file size: 5000

Throughput: Read after each scan processing, unlikely changed.

Remarks: $N_{mwPT(VMRxx)}$ is the number of microwindows used in the actual retrieval (~ 20) , N_{acq} is the number of sweeps of the scan that is going to be processed (~ 16). One file for each retrieval (PT, VMRxx) is needed.

Data set: MP2-ANC-03-PT(VMRxx)				
Field name	**Format**	**Units**	**Size**	**Comments**
Microwindow #1				
Progressive number	long (ul)	-	4	
Label of the microwindow	character string	-	10	
Index of the first sampling point	long (ul)	-	4	
Number of sampling points for the microwindow	long (ul)	-	4	
Minimum and maximum altitude coverage of the microwindow	two floats	km	8	
Number of transitions lines in the spectroscopic database	long (ul)	-	4	
Field indicating whether LUT's exist for this MW	long (ul)	-	4	
Number of gases for which LUT's exist	long (ul)	-	4	ngas(1)
Field indicating whether the irregular grid exists for this MW	long (ul)	-	4	
Hitran codes of the gases for which LUT's exist	long (ul) vector	-	4*ngas(1)	
Sweep #1				
Spectral interval around the actual microwindow in which the atmospheric continuum can be assumed as varying linearly	double	cm^{-1}	8	
Occupation matrix element for sweep #1 and microwindow #1	logical	-	2	see note 1
Sweep #N_{acq}				
Spectral interval around the actual microwindow in which the atmospheric continuum can be assumed as varying linearly	double	cm^{-1}	8	
Occupation matrix element for sweep #N_{acq} and microwindow #1	logical	-	2	see note 1
The same structure of microwindow #1 is repeated for microwindows #2...$N_{mwPT(VMRxx)}$				

Notes:

1. The row index of the logical matrix refers to the sweep number, while the column index refers to the microwindow.

Example: let's call **L** the logical occupation matrix contained in this data set and **v** the vector of the labels. If **L(*i*,*j*)** = 'TRUE' the microwindow whose label is equal to **v**(*j*) is considered at the sweep *i*. The same microwindow will not be considered at the sweep *i* if **L(*i*,*j*)** = 'FALSE'.

FORTRAN code used to read the file

```
      subroutine r_mwoccmat_vmr(nselmw,ilimb)
* subroutine to read file of microwindows and occupation matrix
      implicit none
      include 'parameters_vmr.inc'
      integer*4 nf,j,k,nselmw,iordmw,ilimb,ifspmw(imxmw),nsam,iline(imxmw)
      integer*4 ngas(imxmw),ih(imxgmw,imxmw),ilookupmw(imxmw),i
      real*8 rmin,rmax,rconint(imxlmb,imxmw)
      character slab*10
      character smw(imxmw)*6,cc*2
      logical lokku(imxgeo,imxmw),lmgas(imxgmw,imxmw),lirrgridmw(imxmw),lirrgrid
      common/mwoccmat_v/ ifspmw,iline,rconint
      common/mwlookup_v/ ilookupmw,smw,ngas,ih,lmgas
      common/lmwoccmat_v/ lokku
      common/irrgrid_v/ lirrgridmw,lirrgrid
      nf=13
      call skip_vmr(nf)
      do 10 j=1,nselmw
        read(nf,130) iordmw,slab,ifspmw(j),nsam,rmin,rmax,iline(j),ilookupmw(j),ngas(j),lirrgridmw(j)
        if(ngas(j).ne.0) then
          write(cc,'(i2)')ngas(j)
          read(nf,'('//cc//'i3)') (ih(i,j),i=1,ngas(j))
        end if
        smw(j)=slab(5:10)
        call blind_vmr(iline(j),'imxlin','transitions in r_mwoccmat')
10    continue
```

- ifspmw(j) = index of the first sampling point of MW j. NOTE: the sampling point at frequency=0 has index=1
- iline(j) = number of transitions in the spectroscopic database for MW j
- iordmw,slab,nsam,rmin,rmax = are scratch fields reporting:
- iordmw = progressive number
- slab = label of the MW as read in the observ_h2o.dat file
- nsam = number of sampling points for the MW
- rmin,rmax = minimum and maximum altitude coverage of the MW
- ilookupmw = 1 at least a cross-section look-up table (LUT) is available for this MW, ilookupmw = 0 no LUT's available for this MW
- ngas(j) = number of gases for which the LUTs exist for MW j
- lirrgridmw(j) = logical variable which tells whether a file of the irregular grid exists for this MW
- ih(i,j) = Hitran codes of the gases for which LUT's exist for this MW (these parameters are read only if ngas(j).ne.0

```
      do 20 j=1,nselmw
        call skip_vmr(nf)
        read(nf,110)(rconint(k,j),k=1,ilimb)
20    continue
* rconint(k,j) = spectral interval, around MW j, in which atmospheric continuum can be assumed to vary linearly at geometry k (8f10.5)
      call skip_vmr(nf)
      do 30 j=1,ilimb
        read(nf,120)(lokku(j,k),k=1,nselmw)
30    continue
* lokku(j,k) = ocupation matrix element for geometry j and MW k (40l2)
100   format(2i5)
110   format(8f10.5)
120   format(40l2)
130   format(i5,a10,2i10,2f10.5,i5,2i2,l2)
      return
      end
```

Identifier: MP2-ANC-04-PT(VMRxx)

Name: VCM of the forward model

File name: vcm_fm_pt(VMRxx).dat

Type: Spare input file, not used in ORM_ABC Vers. 1.1 code, probably this has to be a binary file.

Description: In this data set the Variance Covariance Matrix (VCM) of the FM should be recorded. This matrix takes into account the uncertainties on the simulated spectra due to the errors on the parameters in input to the forward model.

Data size: 200 Mb (TBC)

Typical file size: TBD

Throughput: read each LS sequence, modified unfrequently.

Remarks: This is a spare file which is not be used in the version 2.3 of the scientific code and which can be cancelled if it causes any difficulty.

Data set: MP2-ANC-04-PT(VMRxx)				
Field name	**Format**	**Units**	**Size**	**Comments**
VCM data of the FM	float matrix	-	200 Mb	

FORTRAN code used to read the file
Does not exist

Identifier: MP2-ANC-05

Name: Apodising vector ancillary data set

File name: apod.dat

Type: Input ASCII

Description: This data set contains the definition of the apodising vector which has been used for the apodisation of the spectra and for the calculation of AILS, and which will be used by the ORM for the calculation of the VCM of the observations.

Data size: 2136

Typical file size: 5885

Throughput: Read each scan processing, modified only unfrequently

Remarks: The apodising vector is provided in the interferogram domain. Only 513 data points are needed: the first point corresponds to the zero path difference position, the last point corresponds to a path difference equal to 20 cm.

Data set: MP2-ANC-05				
Field name	**Format**	**Units**	**Size**	**Comments**
Description of the apodisation function	character string	-	80	
Number of points of the apodising vector	ul	-	4	
Apodising vector	float vector	-	2052	see notes 1,2

Notes:
1. This vector is externally pre-computed.
2. The baseline is to use the same apodising vector (Norton-Beer strong) for all the MIPAS spectral bands (all the microwindows and all the retrievals).

FORTRAN code used to read the file

```
*  Apodisation function in interferogram domain
          open(31,file='INP_FILES/apod.dat',status='old')
          nf=31
          call skip_pt(nf)          ! skipping comments in the file
          read(nf,*) napod
* napod = number of data points constituting the apodisation function
*  Control on the value of napod: IT HAS TO BE (2^n+1), with n=integer
          rx=(log(dble(napod-1)))/log(2.)
          write(*,*)'rx=',rx
           n=nint(rx)
          write(*,*)'n=',n
           if(abs(rx-dble(n)).gt.1.e-6) then
                    write(*,*)'program stopped in finput_pt.f'
                    write(*,*)'(napod-1) has to be 2^n, with n integer'
```

```
            write(*,*)'(napod-1) =',(napod-1)
            stop
        end if
        read(nf,*) (rapod(j),j=1,napod)
        close(nf)
        end
```

Identifier: MP2-ANC-06

Name: VCM of the pointings ancillary data set

File name: vcmlos.dat

Type: Input ASCII

Description: In this data set are recorded:

- the vector of the engineering estimate of the differences between the tangent altitudes of the analysed sweeps.
- the inverse of the Variance Covariance Matrix (VCM) related to the vector of the differences between the tangent altitudes of the analysed sweeps. This matrix is obtained from satellite characterisation (either ground simulations or in flight performance tests).

Data size: $4*(N^{SW}-1)* N^{SW}$

Typical file size: 3270

Throughput: Changed only rarely (several months). The scientific code reads this data set for the processing of each limb scanning sequence, operational code may adopt a different strategy.

Remarks: Used by ORM since Vers. 2.2 (only p,T retrieval uses this file)

Data set: MP2-ANC-06				
Field name	**Format**	**Units**	**Size**	**Comments**
Vector of the differences between tantent altitudes	float vector	km	$4*(N^{SW}-1)$	
VCM data of the pointings	float matrix	km^2	$4*(N^{SW}-1)^2$	

FORTRAN code used to read the file

```
      subroutine r_infolos(ilimb,imxlmb,rdzeng,rinvclos, lextinf1,lifend)
      implicit none
      integer*4 imxlmb, ilimb, nf, i, j
      real*8 rdzeng(imxlmb),rinvclos(imxlmb,imxlmb)
      logical lextinf1,lifend
* If LOS info ahs to be used the file is read:
      if (lextinf1.or.lifend) then
        nf=14
        call skip_pt(nf)
        read(nf,*)(rdzeng(j),j=1,ilimb-1)
        call skip_pt(nf)
        do i=1,ilimb-1
          read(nf,'(8e12.4)')(rinvclos(i,j),j=1,ilimb-1)
        end do
```

```
    else
• If LOS info is NOT to be used the variables are set to zero:
    do i=1,ilimb
       rdzeng(i) = 0.0d0
      do j=1,ilimb
        rinvclos(i,j) = 0.0d0
      end do
    end do
  end if
  end
```

Identifier: MP2-ANC-07-PT(VMRxx)

Name: Cross section look-up tables for p, T and VMR retrievals.

Field name: lookup_[*mwname*][*gasname*]

Type: Input binary

Description: In this ancillary data set are contained the values of absorption cross sections of the gas = *gasname* required for the calculation of the spectrum related to the MW = *mwname.* The cross-sections are given for a discrete set of temperatures, pressures and frequencies; one file per microwindow per gas will be used. The files containing lookup tables are generated by Oxford University and their format is described in RD3.

Identifier: MP2-ANC-08

Name: Definition of initial guess pressure profile.

File name: in_pres.dat

Type: Input ASCII

Description: This ancillary data set contains an atmospheric pressure profile referred to the altitude grid reported in data set MP2-ANC-12. These data are used to build a first guess for the unknowns P_{tang} and T(P) in P, T retrieval process. When the switch for the use of p,T retrieved data has not been enabled in VMR retrieval, the data recorded in this file are used by VMR retrieval as well.

Data size: 225

Typical file size: 633

Throughput: size / each limb - scanning sequence

Remarks: The following remarks apply to this data set as well as to data sets MP2-ANC-09, MP2-ANC-10 and MP2-ANC-11 containing initial guess profiles:

1. Since the fitting procedure is more efficient if the starting profiles are similar to the real ones, in all the data sets containing initial guess profiles should be present different profiles depending on the measurement conditions (season and latitude). As a suggestion, either profiles arising from external information or from a previous retrieval could be copied in this data set. A procedure for preparing the initial guess profiles will be identified on the basis of test retrievals.
2. All the initial guess profiles are referred to 50 steps altitude grid defined in the data set MP2-ANC-12.

Data set: MP2-ANC-08				
Field name	**Format**	**Units**	**Size**	**Comments**
Identifier	character string	-	25	
Starting profile of pressure	float vector	hPa	200	

FORTRAN code used to read the file

```
* initial P profile for p,T retrieval
            open(16,file='./INP_FILES/in_pres.dat',status='old')
            nf=16
            call skip_pt(nf)          ! skipping comments on the file
            read(nf,'(a25)') sident
* sident = alphanumeric identifier of the file (a25)
            call skip_pt(nf)
            read(nf,*)(rpprof(j),j=1,ipro)
* rpprof(j) = value of P at altitude j of the initial profile (8f10.5)
100         format(8f10.5)
            close(nf)
      end
```

Identifier: MP2-ANC-09

Name: Definition of temperature profile initial guess

File name: in_temp.dat

Type: Input ASCII

Description: This ancillary data set contains a temperature profile referred to the altitude grid defined in the ancillary data set MP2-ANC-12. These data are used to build a first guess for the unknowns T(P) in P, T retrieval process. When the switch for the use of p,T retrieved data has not been enabled in VMR retrieval, the data recorded in this file are used by VMR retrieval as well.

Data size: 225

Typical file size: 626

Throughput: size / each limb - scanning sequence

Remarks: See remark of MP2-ANC-08. Data in MP2-ANC-08 and MP2-ANC-09 obey to hydrostatic equilibrium law.

Data set: MP2-ANC-09				
Field name	**Format**	**Units**	**Size**	**Comments**
Identifier	character string	-	25	
Starting profile of temperature	float vector	K	200	

FORTRAN code used to read the file

```
* initial T profile
      open(17,file='./INP_FILES/in_temp.dat',status='old')
      nf=17
      call skip_pt(nf)
      read(nf,'(a25)') sident
* sident = alphanumeric identifier of the file (a25)
      call skip_pt(nf)
      read(nf,100)(rtprof(j),j=1,ipro)
* rpprof(j) = value of T at altitude j of the initial profile (8f10.5)
100   format(8f10.5)
      end
```

Identifier: MP2-ANC-10-PT(VMRxx)

Name: Definition of continuum profiles initial guess.

File name: in_cont_pt(VMRxx).dat

Type: Input ASCII

Description: This ancillary data set contains continuum profiles referred to the altitude grid defined in the data set MP2-ANC-12. The profiles are referred to the frequency grid identified by the central frequencies of the microwindows used in p, T (VMRxx) retrieval. These data are used as a first guess in order to fit the atmospheric continuum emission in p,T(VMRxx) retrieval.

Data size: 225 *$N_{mwPT(VMRxx)}$

Typical file size: 8000

Throughput: size / each limb - scanning sequence

Remarks: See remark of MP2-ANC-08. A different file is accessed for different microwindow selections.

Data set: MP2-ANC-10				
Field name	**Format**	**Units**	**Size**	**Comments**
Microwindow #1:				
Microwindow label	character string	-	25	
Continuum profile	float vector	cm^2	200	
...	...	...	...	
Microwindow #$N_{mwPT(VMRxx)}$ ***:***				
Microwindow label	character string	-	25	
Continuum profile	float vector	cm^2	200	

FORTRAN code used to read the file

```
* initial continuum profiles
        open(18,file='./INP_FILES/in_cont_pt.dat',status='old')
        nf=18
        call skip_pt(nf)          ! skipping comment lines
        do 10 j=1,nselmw
                read (nf,'(a25)') smwlabel
* smwlabel = scratch label of the microwindow (a25)
                read(nf,110)(rcprof(k,j),k=1,ipro)
* rcprof(k,j) = continuum cross section for MW j at altitude k (8e10.3)
  10    continue
 110    format (8E10.3)
        end
```

Identifier: MP2-ANC-11

Name: Definition of initial guess VMR profiles

File name: in_vmr_prof.dat

Type: Input

Description: This ancillary data set contains VMR atmospheric profiles referred to the altitude. The overall set of gases contained in HITRAN96 catalogue is included.
Retrieved VMR profiles of the MIPAS target species are possibly used as initial guess profiles in VMR retrievals.
VMR profile of CO_2 is considered as known and is used in p,T retrieval.
VMR profiles of interfering species are contained in this data set as well and are considered as known quantities in all the retrievals.

Data size: 25 + 229*36 (36 is the total number of gases present in HITRAN 96 spectroscopic database)

Typical file size: 18923

Throughput: Size/each limb scanning sequence.

Remarks: See remark of MP2-ANC-08. The profiles recorded in this data set are referred to the altitude grid defined in MP2-ANC-12

Data set: MP2-ANC-11				
Field name	**Format**	**Units**	**Size**	**Comments**
Identifier	character string	-	25	
GAS # 1				
Progressive number	long (ul)	-	4	
Name of the gas	character string	-	25	
VMR profile of gas	float vector	ppm	200	
…	…	…	…	…
GAS # 36				
Progressive number	long (ul)	-	4	
Name of the gas	character string	-	25	
VMR profile of gas	float vector	ppm	200	

FORTRAN code used to read the file

```
* initial VMR profiles
      open(19,file='./INP_FILES/in_vmr_prof.dat',status='old')
      nf=19
      do 10 j=1,igas
```

```
                        rewind nf
                        call skip_pt(nf)
                        read(nf,'(a25)') sident
                        call skip_pt(nf)
                        do 10 k=1,igashi(j)
                                read(nf,100) igc,sname(j)
* igc   = progressive number denoting hitran gas code
* sname = name of the gas
                        read(nf,110)(rvmrprof(l,j),l=1,ipro)
  10                    continue
100         format(i5,1x,a25)
110         format(8e10.3)
            end
```

Identifier: MP2-ANC-12

Name: Definition of the altitude grid to which the initial guess profiles are referred

File name: in_alt.dat

Type: Input ASCII

Description: This ancillary data set contains the altitude grid to which all the initial guess profiles are referred.

Data size: 229

Typical file size: 737

Throughput: Read each limb scanning sequence, modified only very unfrequently

Remarks: 50 altitude steps are assumed for the calculation of the data size.

Data set: MP2-ANC-12				
Field name	**Format**	**Units**	**Size**	**Comments**
Identifier	character string	-	25	
Number of points of the altitude grid	long (ul)	-	4	
Altitude grid	float vector	km	200	

FORTRAN code used to read the file

```
* altitudes of initial profiles
            open(20,file='./INP_FILES/in_alt.dat',status='old')
            nf=20
            call skip_pt(nf)
            read(nf,'(a25)') sident
* sident = alphanumeric identifier of the file (a25)
            call skip_pt(nf)
```

```
      read(nf,100) ipro
* ipro = number of elements of the vector of altitudes (i5)
      call skip_pt(nf)
      read(nf,110)(rzprof(j),j=1,ipro)
* rzprof(j) = altitude to which element j of T, P and VMR profiles are referred to (8f10.5)
100   format(i5)
110   format(8f10.5)
      close (nf)
      end
```

Identifier: MP2-ANC-13-PT(VMRxx)

Name: Spectroscopic database ancillary data set.

File name: spect_db_pt(VMRxx).dat

Type: Input ASCII

Description: this ancillary data set contains spectroscopic data which will be used for the simulation of atmospheric spectra (forward calculations). The database is organised by microwindows; all (and only) the microwindows used in the actual retrieval are included.

Data size: ≈ 8,000

Typical file size: < 1 Mb

Throughput: size / each scan processing

Remarks:

- We assume that natural abundance of isotopic species is already taken into account in this data set and we do not need a separate data set for this quantity.
- $N_{mwPT(VMRxx)}$ is the number of microwindows used processed in the current retrieval.

Data set: MP2-ANC-13				
Field name	**Format**	**Units**	**Size**	**Comments**
Microwindow #1				
Microwindow label	character string	-	25	
Parameters line 1	character string	-	80	
….	….	….	….	….
Parameters line NL(1)	character string	-	80	
...	...	...	...	...
...	...	...	...	...
Microwindow #$N_{mwPT(VMRxx)}$				
Microwindow label	character string	-	25	
Parameters line 1	character string	-	80	
….	….	….	….	….
Parameters line NL($N_{mwPT(VMRxx)}$)	character string	-	80	

FORTRAN code used to read the file

```
* spectroscopic data
            open(21,file='./INP_FILES/spect_db_pt.dat',status='old')
            nf=21
            do 10 j=1,nselmw
                        call skip_pt(nf)
                        read(nf,'(a25)') smwlabel
* smwlabel = scratch label of the microwindow (a25)
                        do 20 k=1,iline(j)
                                    read(nf,100) dsilin(k,j),rint0(k,j),rhw0(k,j),relow(k,j),
     &                  rexph(k,j),ioutin(k,j),iiso(k,j),icode(k,j),
     &                  ruplin(k,j),rlolin(k,j)
                                    if (rlolin(k,j).le.8.)then
                                                rlolin(k,j)=0.
                                    else
                                                rlolin(k,j)=rlolin(k,j)-.5
                                    end if
                                    if(abs(ruplin(k,j)-70.) .le. 1.e-4)then
                                                ruplin(k,j)=rulatm
                                    else
                                                ruplin(k,j)=ruplin(k,j)+10.
                                    end if
* dsilin = line position
* rint0  = line strength
* rhw0   = pressure broadening
* relow  = energy of starting level
* rexph  = exponent for T dependence of rhw0
* ioutin = control on continuum treatment
* iiso   = isotope code
* icode  = hitran code of the molecule
* ruplin = highest altitude where the line has to be used
* rlolin = lowest altitude where the line has to be used
*
20                  continue
10          continue
            close (nf)
100         format(f12.6,(1pe11.3),0pe11.3,2f11.4,2x,i1,2x,i2,1x,i2,2f7.2)
            end
```

Identifier: MP2-ANC-14-PT(VMRxx)

Name: Non local thermal equilibrium (NLTE) data for p, T (VMRxx)retrieval

File name: vib_temp_pt(VMRxx).dat

Type: Input (TBD if ASCII or binary)

Description: This ancillary data set contains tables of the vibrational temperatures related to the spectral features involved in the forward simulations of the actual retrieval. Needed for taking into account Non-LTE effects. This is a spare file not used by the ORM.

Data size: < 20,400 (TBC)

Typical file size: TBD

Throughput: Read each limb scanning sequence. An update may be required at each limb scanning sequence (TBC).

Remarks: This is for the moment a spare data set because a final strategy for Non-LTE retrievals has not been identified.

Data set: MP2-ANC-14-PT(VMRxx)				
Field name	**Format**	**Units**	**Size**	**Comments**
Vibrational temperatures	float matrix	K	< 20,000	
Vector of corresponding altitude	float vector	km (TBC)	400	see note 1

Notes:
1. This vector contains altitudes to which temperature profiles are referred.

FORTRAN code used to read the file
Does not exist

Identifier: MP2-ANC-15

Name: Files of irregular spectral grids

File name: irrgrid_[*mwname*].dat

File type : ASCII

Throughput: Read each limb scanning sequence. Unlikely modified

Description: These ancillary data set contain the definition of optimized spectral grids used by the ORM code for the calculation of the radiative transfer for the MW = *mwname*. The optimized grids are generated by University of Oxford and the file formats are described in RD2. One file per microwindow will be used.

Identifier: MP2-ANC-16

Name: User-defined pressure grid

File name: fixed_p_grid.dat

Type: Input ASCII

Description: This ancillary data set contains the pressure grid which could be used for the retrieval at fixed pressure levels if this option is required

Data size: 280

Typical file size: 475

Throughput: Read each limb scanning sequence, modified only very infrequently

Remarks: For the moment this is only a spare data set which may be required for the representation of output profiles in a standard pressure grid.
.

Data set: MP2-ANC-16				
Field name	**Format**	**Units**	**Size**	**Comments**
Description of pressure grid	character string	-	80	
Fixed pressure grid	float vector	-	200	

FORTRAN code used to read the file
does not exist.

2.1.3 External data sets (input data)

Herewith are described the external data sets that could be used as input of Level 2 scientific code. All the external data sets are for the moment spares, not used.

Identifier: MP2-EXT-01

Name: External p,T profiles (spare file)

File name: ext_pt.dat

Type: Input ASCII

Description: In this data set are present external p,T profiles that could be optionally used by p,T retrieval module (only if the switch for the use of external information is enabled).

Data size: 30,700

Typical file size: TBD

Throughput: size / each limb scanning sequence

Remarks:
- In this data set are contained the profiles that we optionally want to combine, by means of weighted average, with the retrieved profiles; for this reason, different profiles depending on measurement conditions (season, latitude, local time) are expected in this data set. A final strategy for the choice of the profiles which should be present in this data set has not been identified yet.
- The profiles contained in this data set are supposed as referred to the altitude grid defined in MP2-ANC-12.

Data set: MP2-EXT-01				
Field name	**Format**	**Units**	**Size**	**Comments**
Spare field	character string	-	100	
Temperature profile	float vector	K	200	
Pressure profile	float vector	hPa	200	
VCM data of T profile	float vector	K^2	10,000	
VCM data of p profile	float vector	hPa^2	10,000	
p,T correlation matrix	float matrix	-	10,000	

FORTRAN code used to read the file
not available

Identifier: MP2-EXT-02xx

Name: External VMRxx profiles (spare file)

File name: ext_vmr_xx.dat (with xx= h2o, o3, hno3, ch4, n2o)

Type: Input ASCII

Description: In this data set are present external data that could be optionally used in VMRxx retrievals. For the moment only external VMR profiles of the five MIPAS high priority species have been foreseen. These profiles would be used during VMR retrievals only if the switch for the use of external information is enabled.

Data size: 10,300

Typical file size: TBD

Throughput: Size / each limb - scanning sequence

Remarks:

- In this data set are contained the profiles we optionally want to combine, by means of weighted average, with the retrieved profiles; for this reason, different profiles depending on measurement conditions (season, latitude, local time) are expected in this data set. A final strategy for the choice of the profiles which should be present in this data set has not been identified yet.
- The profiles contained in this data set are supposed referred to the altitude grid defined in MP2-ANC-12.

Data set: MP2-EXT-02xx				
Field name	**Format**	**Units**	**Size**	**Comments**
Spare field	character string	TBD	100	
VMR profile of gas xx	float vector	ppm	200	
VCM of gas xx profile	float matrix	-	10,000	

FORTRAN code used to read the file
not available

Identifier: MP2-EXT-03-PT(VMRxx)

Name: Cross sections of continuum: vertical profiles externally provided and related to the microwindows used in p, T (VMRxx) retrieval. (spare file).

File name: ext_cont_pt(VMRxx).dat

Type: Input ASCII

Description: This external data set contains cross sections profiles of atmospheric continuum for different values of frequency. The frequency grid in which the profiles are given is identified by the central frequencies of the microwindows used in the actual retrieval (see MP2-ANC-03).

Data size: ≈ 4 Mb (if $N_{mwPT(VMRxx)}$ = 20 and altitude grid of 50 height steps)

Typical file size: TBD

Throughput: size / each limb - scanning sequence

Remarks:
- The cross section profiles of atmospheric continuum are supposed referred to the altitude grid specified in MP2-ANC-12.
- $N_{mwPT(VMRxx)}$ is the number of different microwindows used in p, T (VMRxx) retrieval.

Data set: MP2-EXT-03-PT(VMRxx)				
Field name	**Format**	**Units**	**Size**	**Comments**
Description	character string	-	80	
Frequency #1				
Microwindow label	character string	-	25	see note 1
Continuum profile	float vector	cm^2	200	
...	...	...	...	
Frequency # $N_{mwPT(VMRxx)}$				
Microwindow label	character string	-	25	
Continuum profile	float vector	cm^2	200	
VCM of continuum profiles	float matrix	cm^4	$4*(50*N_{mwPT(VMRxx)})^2$	see note 2

Notes:
1. The same microwindow label of MP2-ANC-03 shall be used.
2. Possible correlations between profiles at different frequencies are also accounted in this VCM.

FORTRAN code used to read the file
does not exist

Identifier: MP2-EXT-04-PT(VMRxx)

Name: A-priori knowledge on instrument radiometric offset (spare file)

File name: instr_off_pt(VMRxx).dat

Type: Input

Description: This external data set contains an a-priori estimate of the instrument radiometric offset together with its variance-covariance matrix.

Data size: $4 * N_{mwPT(VMRxx)} * (N_{mwPT(VMRxx)} + 1)$

Typical file size: TBD

Throughput: Read each limb scanning sequence, modified only very unfrequently

Remarks:
- MW_{xx} is the total number of microwindows used in the actual retrieval (see MP2-ANC-04-PT(VMRxx)).

Data set: MP2-EXT-04-PT(VMRxx)				
Field name	**Format**	**Units**	**Size**	**Comments**
Vector of instrument offset	float vector	r.u.	$4 * N_{mwPT(VMRxx)}$	see note 1
VCM of the vector of instrument offset	float matrix	$r.u.^2$	$4 * (N_{mwPT(VMRxx)})^2$	

Notes:
1. This vector contains an a-priori estimate of the instrument radiometric offset at the central frequencies of the microwindows used in the actual retrieval. These microwindows are defined in MP2-ANC-03-PT(VMRxx).

FORTRAN code used to read the file
does not exist

2.1.4 Internal data sets

The internal data sets contain data that are exchanged between p,T and VMR modules or data echanged in subsequent VMR retrievals.

Identifier: MP2-INT-01-PT

Name: Temperature and pressure profiles computed in p, T retrieval.

File name: pt_dump.dat

Type: Output of P, T retrieval / Input of VMR retrievals, binary file

Description: This data set contains p,T, profiles computed by p, T retrieval, as well as the retrieved values of the tangent altitudes. These data are used by the subsequent VMR retrievals when the switch for the use of p,T retrieved data in VMR retrievals is enabled.

Data size: 464

Typical file size: 2640

Throughput: Size / Each limb scanning sequence

Data set: MP2-INT-01-PT				
Field name	**Format**	**Units**	**Size**	**Comments**
N. of points in the p,T,z profiles	ul	-	4	
parameters used for variables dimensioning in p,T retrieval	two ul	-	8	
n. of sweeps used in p,T retrieval	ul	-	4	
Pressure provile	float vector	hPa	128	
Temperature profile	float vector	K	128	
Altitude grid to which the profiles are referred	float vector	km	128	
Retrieved tangent altitudes	float vector	km	64	

FORTRAN code used to write / read the file

```
* Writing p,T retrieved values into a dump file:
open(34,file='./OUT_FILES/pt_dump.dat',  form = 'unformatted', status = 'unknown')
    write(34) ibase
    write(34) imxpro
    write(34) imxgeo
    write(34) ilimb
    write(34) rpbase
    write(34) rtbase
    write(34) rzbase
    write(34) rztang
    close (34)
```

Identifier: MP2-INT-02-VMRxx

Name: VMRxx profile from VMRxx retrieval

File name: xx_dump.dat (with xx = h2o, o3, hno3, ch4, n2o)

Type: Output of VMRxx retrieval

Description: This data set contains the VMRxx retrieved profile. This file is written at the end of each VMR retrieval and, in turn, each VMR retrieval can read the files written by the VMR retrievals (related to the same scan) that have already been completed. The flow of the data is also explained in the diagram of fig.1.

Data size: 136

Typical file size: 520

Throughput: Size / Each VMR retrieval

Data set: MP2-INT-04-VMRxx				
Field name	**Format**	**Units**	**Size**	**Comments**
n. of points in the VMR profile	ul	-	4	
HITRAN code of the retrieved gas	ul	-	4	
VMR profile of the retrieved gas	float vector	ppm	128	

FORTRAN code used to write / read the file

```
* Writing of vmr retrieved profile into a dump file:
      open(34,file='./OUT_FILES/'//sr1//'_dump.dat',
     &          form='unformatted',status='unknown')
      write(34) ibase
      write(34) igashi(1)
      do j=1,ibase
         write(34) rvmrbase(j,1)
      end do
      close (34)
```

2.1.5 ORM data products

Identifier: MP2-PRD-PT

Name: p,T retrieval main products

File name: pt_out.dat

Type: Output ASCII

Description: This data set contains the main products of p,T retrieval

Data size: ≈ 210 Kb if $N_p = N_{acq} = 16$ and NWC = 100

Typical file size: 270 Kb

Throughput: Size / Each limb scanning sequence

Remarks: N_p is the total number of steps of the grid in which the profiles are retrieved; this number is equal to the number of true elements of the logical vector of MP2-ANC-01(≈16). NWC is equal to the number of retrieved continuum parameters.
N_{mwPT} is the number of microwindows used in p,T retrieval at the *i-th* sweep. See also notes of the data set MP2-L1B-PT.

Data set: MP2-PRD-PT				
Field name	**Format**	**Units**	**Size**	**Comments**
Retrieved values of tangent pressure	float vector	hPa	$4 * N_p$	
Retrieved values of temperature at tangent points	float vector	K	$4 * N_p$	
Retrieved values of continuum cross sections	float vector	cm^2	4 * NWC	see note 1
Instrumental offset	float vector	r.u.	$4* N_{mwPT}$	
VCM of the retrieved parameters	float matrix	variable	$4*(NWC+2*N_p)^2$	
Occupation matrix of the continuum retrieved parameters	logical matrix	-	$4* N_{mwPT}*N_p$	
Logical matrix which identifies tightly grouped MWs/altitudes for continuum retrieval	logical matrix	-	$4* N_{mwPT}*N_p$	
Microwindow 1				
Sweep 1				
Qualifier related to the continuum parameter of this MW & sweep	integer	-	4	group_type
continuum cross-section at the current MW, sweep	float	cm^2	4	xsect
variance of the continuum parameter at the current MW, sweep	float	cm^4	4	var
covarance between current continuum parameter and pressure at the same sweep	float	cm^2*hPa	4	covp
covarance between current continuum parameter and temperature at the same sweep	float	cm^2*K	4	covt
The same structure is repeated for sweeps 1, …, N_{acq} and microwindows 1, ...N_{mwPT}				

Continued ...				
Height corrections for tangent altitudes	float vector	km	$4 * (N_p-1)$	
VCM of the height corrections for tangent altitudes	float matrix	km^2	$4 * (N_p-1)^2$	
Sweep #1:				
Microwindow #1				
Microwindow label	character string	-	25	
N. of spectral data points in the actual microwindow	long (ul)	-	4	
Residual spectrum	float vector	r.u.	< 400	
Value of partial c^2 function	float	-	4	see note 2
…	…	…	…	
Microwindow #N_{mwPT}				
Microwindow label	character string	-	25	
N. of spectral data points in the actual microwindow	long (ul)	-	4	
Residual spectrum	float vector	r.u.	< 400	
Value of partial c^2 function	float	-	4	see note 2
Same structure repeated for sweeps 2, …, N_{acq}				
Spare field for confidence data of the current retrieval	float	TBD	400	

Notes:

1. NWC is equal to the number of retrieved continuum parameters according to the qualifier matrix of MP2-ANC-03-PT.
2. This is the value of the partial c^2 function related to the actual microwindow and sweep.

FORTRAN code used to write the file

```
      write(29,*) 'FILE PT_OUT.DAT'
* Retrieved values of pressure at tangent points
      write(29,*) 'Retrieved values of pressure at tangent points'
      write(29,*) (rxpar(j),j=1,ipar)
* Retrieved values of temperature at tangent points
      write(29,*) 'Retrieved values of temperature at tangent points'
      write(29,*) (rxpar(j),j=ipar+1,2*ipar)
* Retrieved values of continuum cross sections at tangent points
      write(29,*) 'Retrieved values of continuum cross sections '
      write(29,*) 'at tangent points'
      write(29,*) (rxpar(j),j=2*ipar+1,2*ipar+icontpar)
* Instrumental offset
      write(29,*) 'Instrumental offset '
```

```
            write(29,*) (rxpar(j),j=2*ipar+1+icontpar,itop)
* VCM of the retrieved parameters
            write(29,*) 'VCM of the retrieved parameters'
            do j=1,itop
                  write(29,*) (rainv(j,k),k=1,itop)
            end do
* Occupation matrix of continuum retrieved parameters:
            write(29,*) 'Occup. Matrix of continuum retrieved parameters:'
            write(29,*) '(continuum parametres are row-wise numbered '//
     &                        'starting from top)'
            do j=1,ilimb
                  write(29,'(40l2)')(lcfit(j,k),k=1,nselmw)
            end do
* Matrix identifying tightly grouped MWs for continuum retrieval:
      write(29,*) 'lccmat = '
      do j=1,ilimb
        write(29,'(40l2)')(lccmat(j,k),k=1,nselmw)
      end do
*
* Writing parameters characterising retrieved continuum (see also source code of cont_char_pt.f)
      call cont_char_pt(rxpar,rainv,ilimb,ipar,nselmw,nucl,
     &           lokku,lcfit,lccmat)
* Height corrections for tangent altitudes
            write(29,*) 'Height corrections for tangent altitudes'
            write(29,*) (rztanginit(j)-rztang(j),j=1,ilimb-1)
* VCM of the height corrections for tangent altitudes
            write(29,*) 'VCM of the height corrections for tangent altitudes'
             do i=1,ilimb-1s
               write (29,*) (rvchcorr(i,j),j=1,ilimb-1)
             end do
            write(29,*) 'sweep cycle'
            do k=1,nselmw
               m(k)=0
            end do
            do j=1,ilimb                ! sweep cycle
                  write(29,*) 'Sweep',j
                  do k=1,nselmw         ! Microwindows CYCLE
                    if (lokku(j,k)) then
                      m(k)=m(k)+1
                      write (29,*) ' Microwindow label'
                      write (29,*)  slab(k)
                      write(29,*) 'number of spectral data points in the actual mw'
                      write(29,*) nsam(k)
                      write(29,*) 'residual spectrum'
                      write (29,*) (robs(l,j,k)-rspfov(l,m(k),k),l=1,nsam(k))
                      write(29,*) 'Value of partial chisquare function'
                      write(29,*) rchisqp(m(k),k)
                    end if
                  end do                ! end  Microwindows CYCLE
            enddo                       ! END SWEEP CYCLE
            write(29,*) 'end sweep cycle'
```

Identifier: MP2-PRD-VMRxx

Name: VMR retrieval main products

File name: xx_out.dat (with xx = h2o, o3, hno3, ch4, n2o)

Type: Output ASCII

Description: This data set contains the main products of VMRxx retrieval

Data size: ≈ 200 Kb if $N_p = N_{acq} = 16$ and NWCxx = 100

Typical file size: < 300 Kb

Throughput: Size / Each limb scanning sequence

Remarks: N_p is the total number of steps of the grid at which the profiles are retrieved; this number is equal to the number of true elements of the logical vector of MP2-ANC-02xx (≈16). NWC is equal to the number of retrieved continuum parameters.
$N_{mwVMRxx}$ is the number of microwindows used in VMRxx retrieval. See also notes of the data set MP2-L1B-VMRxx.

Data set: MP2-PRD-VMRxx				
Field name	**Format**	**Units**	**Size**	**Comments**
Retrieved values of VMRxx at tangent points	float vector	ppm	$4 * N_p$	see note 1
Retrieved values of continuum cross sections	float vector	cm^2	4 * NWCxx	see note 2
Instrumental offset	float vector	r.u.	$4* NW_{VMRxx}$	
VCM of retrieved parameters	float matrix	variable	$4*(N_p+NWCxx))^2$	
Occupation matrix of the continuum retrieved parameters	logical matrix	-	$4* NW_{VMRxx}*N_p$	
Logical matrix which identifies tightly grouped MWs/altitudes for continuum retrieval	logical matrix	-	$4* N_{mwPT}*N_p$	

(… Continued)				
Microwindow 1				
Sweep 1				
Qualifier related to the continuum parameter of this MW & sweep	integer	-	4	group_type
continuum cross-section at the current MW, sweep	float	cm^2	4	xsect
variance of the continuum parameter at the current MW, sweep	float	cm^4	4	var
covarance between current continuum parameter and VMR at the same sweep	float	cm^2*ppmv	4	covp
The same structure is repeated for sweeps 1, …, N_{acq} and microwindows 1, ...$N_{mwVMRxx}$				
Column for gas xx (at tangent altitude)	float vector	cm^{-2}	4 * N_p	
VCM of column for gas xx	float matrix	cm^{-4}	4*N_p^2	
Concentration for gas xx (at tangent altitude)	float vector	cm^{-3}	4 * N_p	
VCM of the concentration for gas xx	float matrix	cm^{-6}	4*N_p^2	
Sweep #1:				
Microwindow #1:				
Microwindow label	character string	-	25	
N. of spectral data points in the actual microwindow	long (ul)	-	4	
Residual spectrum	float vector	r.u.	< 400	
Value of partial c^2 function	float	-	4	see note 3
...	...	...	...	
Microwindow #NW_{VMRxx}(1):				
Microwindow label	character string	-	25	
N. of spectral data points in the actual microwindow	long (ul)	-	4	
Residual spectrum	float vector	r.u.	< 400	

(… Continued)				
Value of partial c^2 function	float	-	4	see note 3
Same structure repeated for sweeps 2, ..., N_{acq}				
Spare field for confidence data of the current retrieval	float	TBD	400	

Notes:

1. Tangent altitudes have been retrieved by P,T retrieval module
2. NWCxx is equal to the number of retrieved continuum parameters according to the qualifier matrix of MP2-ANC-03-VMRxx.
3. This is the value of the partial c^2 function related to the actual microwindow and sweep.

FORTRAN code used to write the file

```
      write(29,*) 'FILE VMR_OUT.DAT'
      write(29,*) 'Retrieved values of VMR at tangent points'
      write(29,*) (rxpar(j),j=1,ipar)
      write(29,*) 'Retrieved values of continuum cross sections '
      write(29,*) 'at tangent points'
      write(29,*) (rxpar(j),j=ipar+1,ipar+icontpar)
      write(29,*) 'Instrumental offset '
      write(29,*) (rxpar(j),j=ipar+1+icontpar,itop)
      write(29,*) 'VCM of the retrieved parameters'
      write(29,*) '(Sorting = VMR, continuum, offset)'
      do j=1,itop
        write(29,*) (rainv(j,k),k=1,itop)
      end do
     write(29,*) 'Occup. Matrix of continuum retrieved parameters:'
     write(29,*) '(continuum parametres are row-wise numbered starting from top)'
     do j=1,ilimb
       write(29,'(40l2)')(lcfit(j,k),k=1,nselmw)
     end do
* Matrix identifying tightly grouped MWs for continuum retrieval:
     write(29,*) 'lccmat = '
     do j=1,ilimb
       write(29,'(40l2)')(lccmat(j,k),k=1,nselmw)
     end do
*
* Writing parameters characterising retrieved continuum: (see also the source code of cont_char_vmr.f)
      call cont_char_vmr(rxpar,rainv,ilimb,ipar,nselmw,nucl,
     &           lokku,lcfit,lccmat)
     write(29,*)'Vertical columns of the gas of the current retrieval:'
     write(29,*)(rvcol(j),j=1,ipar)
     write(29,*) 'VCM of columns:'
     do j=1, ipar
       write(29,*) (rvcmcol(j,k),k=1,ipar)
     end do
     write(29,*) 'Concentrations:'
     write(29,*)(rconc(j),j=1,ipar)
     write(29,*) 'VCM of the concentrations: '
     do j=1, ipar
       write(29,*) (rvcmconc(j,k),k=1,ipar)
     end do
       write(29,*) 'sweep cycle'
       do k=1,nselmw
          m(k)=0
       end do
       do j=1,ilimb         !  SWEEP CYCLE
```

```
  write(29,*) 'Sweep',j
  do k=1,nselmw  ! Microwindows CYCLE
    if (lokku(j,k)) then
      m(k)=m(k)+1
      write (29,*) ' Microwindow label'
      write (29,*)  slab(k)
      write(29,*) 'number of spectral data points in the actual mw'
      write(29,*) nsam(k)
      write(29,*) 'residual spectrum'
      write (29,*) (robs(l,j,k)-rspfov(l,m(k),k),l=1,nsam(k))
      write(29,*) 'Value of partial chisquare function'
      write(29,*) rchisqp(m(k),k)
    endif
   enddo            !  end  Microwindows CYCLE
 enddo              !  END SWEEP CYCLE
 write(29,*) 'end sweep cycle'
```

2.1.6 Level 2 annotation data sets

Identifier: MP2-ANN-PT

Name: Annotation data of P, T retrieval

File name: pt_annot.dat

Type: Output ASCII

Description: this data set contains test data and intermediate results which are useful to validate and inspect the evolution of P, T retrieval process.

Data size: ≈ 80,000

Typical file size: 156 Kb

Throughput: Size / Each limb scanning sequence

Data set: MP2-ANN-PT				
Field name	**Format**	**Units**	**Size**	**Comments**
Number of degrees of freedom	long (ul)	-	4	see note 1
Number of iterations, N_{it} (macro+micro)	long (ul)	-	4	
Iteration #1				
Iteration number (for macro iterations)	long(ul)	-	4	
Values of λ damping factor at each iteration cycle	float vector	-	30	
Curreent values of retrieved parameters	float vector	-	480	
$\chi 2$ test linear appoximation	float	-	4	see note 2
$\chi 2$ test	float	-	4	see note 2
Sweep #1				
Synthetic spectrum for microwindow #1		r.u,	≈ 400, see AD1	
…	…	…	…	…
Synthetic spectrum for microwindow #$N_{mwPT(VMRxx)}$			≈ 400, see AD1	
….	….	….	….	…
Sweep # N_{acq}				
Synthetic spectrum for microwindow #1			≈ 400, see AD1	
…	…	…	…	…
Synthetic spectrum for microwindow #$NW_{pmwPT(VMRxx)t}$			≈ 400, see AD1	
Same structure repeated for iterations 2, ..., N_{it}				

Notes:

1. Number of degrees of freedom is equal to number of measured spectral points minus number of retrieved parameters.
2. $\chi 2$ values normalised to number of degrees of freedom

FORTRAN code used to write the file

```
if (linloop) then      ! inside the iterations cycle, do that:
              write(30,*) 'Iteration number'
              write (30,*) iterg
              write(30,*) 'lambda damping factor'
              write(30,*) rlambda
              write(30,*) 'Actual values of the unknown parameters'
              write(30,*) (rxpar(j),j=1,itop)
              write(30,*) 'chisquare linear approx'
              write(30,*) rlinchisq
              write(30,*) 'chisquare '
              write(30,*) rchisq(iterg)
              write(30,*) 'sweep cycle'
              do k=1,nselmw
                        m(k)=0
              end do
              do j=1,ilimb                          !          SWEEPS CYCLE (1..ilimb)
                do k=1,nselmw          !          Microwindows CYCLE
                        if (lokku(j,k)) then
                          m(k)=m(k)+1
                          write(30,*) 'Sweep',j,' Microwindow',k
                          write(30,*) (rspfov(l,m(k),k),l=1,nsam(k))
                        endif
               enddo                                !          end   Microwindows CYCLE
              enddo        !     end SWEEPS CYCLE
end if
if (.not. linloop) then     ! outside the iterations loop:
                        write(30,*) 'Number of degrees of freedom'
                        write(30,*) iobs-itop
                        write(30,*) 'number of iterations, macro+micro'
                        write(30,*) iterg+iterm
endif
```

Identifier: MP2-ANN-VMRxx

Name: Annotation data of VMRxx retrieval

File name: xx_annot.dat (with xx = h2o, o3, hno3, ch4, n2o)

Type: Output ASCII

Description: this data set contains test data and intermediate results which are useful to validate and inspect the evolution of VMR retrieval process.

Data size: 164,420

Typical file size: 300 Kb

Throughput: Size / Each limb scanning sequence

Data set: MP2-ANN-VMRxx				
Field name	**Format**	**Units**	**Size**	**Comments**
Number of degrees of freedom	long (ul)	-	4	see note 1
Number of iterations, N_{it} (macro+micro)	long (ul)	-	4	
Iteration #1				
Iteration number (for macro iterations)	long (ul)	-	4	
Values of λ damping factor at each iteration cycle	float vector	-	30	
Values of retrieved parameters	float vector	-		
$\chi 2$ test linear appoximation	float	-	4	see note 2
$\chi 2$ test	float	-	4	see note 2
Sweep #1				
Synthetic spectrum for microwindow #1	float vector	r.u.	≈ 400, see AD1	
…	…	…	…	…
Synthetic spectrum for microwindow #$N_{mwPT(VMRxx)}$	float vector	r.u.	≈ 400, see AD1	
….	….	….	….	…
Sweep # N_{acq}				
Synthetic spectra for microwindow #1	float vector	r.u.	≈ 400, see AD1	
…	…	…	…	…
Synthetic spectrum for microwindow #$NW_{pmwPT(VMRxx)t}$	float vector			
Same structure repeated for iterations 2, ..., N_{it}				

Notes:

1. Number of degrees of freedom is equal to number of measured spectral points minus number of retrieved parameters.
2. $\chi 2$ values normalised to number of degrees of freedom

FORTRAN code used to write the file

```
        write (30,*) 'FILE VMR_ANNOT.DAT'
At each iteration wrtites:
        write(30,*) 'Iteration number'
        write (30,*) iterg
        write(30,*) 'lambda damping factor'
        write(30,*) rlambda
        write(30,*) 'Actual values of the unknown parameters'
        write(30,*) (rxpar(j),j=1,itop)
```

```
write(30,*) 'chisquare linear approx'
write(30,*) rlinchisq
write(30,*) 'chisquare '
write(30,*) rchisq(iterg)
do k=1,nselmw
   m(k)=0
end do
write(30,*) 'sweep cycle'
do j=1,ilimb              ! SWEEPS CYCLE (1..ilimb)
  do k=1,nselmw           ! Microwindows CYCLE
     if (lokku(j,k)) then
          m(k)=m(k)+1
          write(30,*) 'Sweep',j,' Microwindow',k
          write(30,*) (rspfov(l,m(k),k),l=1,nsam(k))
     endif
  enddo                   !   end   Microwindows CYCLE
enddo                     ! end SWEEPS CYCLE
```

At the end of the iterations cycle then writes:

```
write(30,*) 'Number of degrees of freedom'
write(30,*) iobs-itop
write(30,*) 'number of iterations, macro+micro'
write(30,*) iterg+iterm
```

2.1.7 Correspondence between data set names and file names

The following cross correlation table establishes the correspondence between data set names and the names of the files accessed by the ORM code.

Notes:

- The column 'Type' specifies whether a given file is Input or Output of the ORM.
- The column 'U/S' specifies whether a given file is currently 'Used' or is a 'Spare' file

#	Type	U/S	Data set name	File name
1	I	u	MP2-L1B-PT (VMRxx)	observ_pt.dat observ_o3.dat observ_h2o.dat observ_hno3.dat observ_n2o.dat observ_ch4.dat
2	I	u	MP2-ANC-01	settings_pt.dat
3	I	u	MP2-ANC-02XX	settings_o3.dat settings_h2o.dat settings_hno3.dat settings_n2o.dat settings_ch4.dat
4	I	u	MP2-ANC-03-PT(VMRxx)	mwoccmat_pt.dat mwoccmat_o3.dat mwoccmat_h2o.dat mwoccmat_hno3.dat mwoccmat_n2o.dat mwoccmat_ch4.dat
5	I	s	MP2-ANC-04-PT (VMRxx)	vcm_fm_pt.dat vcm_fm_o3.dat vcm_fm_h2o.dat vcm_fm_hno3.dat vcm_fm_n2o.dat vcm_fm_ch4.dat
6	I	u	MP2-ANC-05	apod.dat
7	I	u	MP2-ANC-06	vcm_los.dat
8	I	u	MP2-ANC-07-PT(VMRxx)	lookup_[*mwname_gasname*] : :
9	I	u	MP2-ANC-08	in_pres.dat
10	I	u	MP2-ANC-09	in_temp.dat
11	I	u	MP2-ANC-10-PT(VMRxx)	in_cont_pt.dat in_cont_o3.dat in_cont_h2o.dat in_cont_hno3.dat in_cont_n2o.dat in_cont_ch4.dat
12	I	u	MP2-ANC-11	in_vmr_prof.dat
13	I	u	MP2-ANC-12	in_alt.dat
14	I	u	MP2-ANC-13-PT(VMRxx)	spect_db_pt.dat spect_db_o3.dat spect_db_h2o.dat spect_db_hno3.dat spect_db_n2o.dat spect_db_ch4.dat
15	I	s	MP2-ANC-14-PT(VMRxx)	vib_temp_pt.dat

				vib_temp_o3.dat vib_temp_h20.dat vib_temp_hno3.dat vib_temp_n2o.dat vib_temp_ch4.dat
16	I	u	MP2-ANC-15	irrgrid_[*mwname*].dat : :
17	I	s	MP2-ANC-16	fixed_p_grid.dat
18	I	s	MP2-EXT-01	ext_pt.dat
19	I	s	MP2-EXT-02xx	ext_vmr_o3.dat ext_vmr_h2o.dat ext_vmr_n2o.dat ext_vmr_hno3.dat ext_vmr_ch4.dat
20	I	s	MP2-EXT-03-PT(VMRxx)	ext_cont_pt.dat ext_cont_o3.dat ext_cont_h20.dat ext_cont_hno3.dat ext_cont_n2o.dat ext_cont_ch4.dat
21	I	s	MP2-EXT-04-PT(VMRxx)	instr_off_pt.dat instr_off_o3.dat instr_off_h2o.dat instr_off_hno3.dat instr_off_n2o.dat instr_off_ch4.dat
22	I/O	u	MP2-INT-01-PT	pt_dump.dat
23	I/O	u	MP2-INT-02-Rxx	h2o_dump.dat o3_dump.dat hno3_dump.dat ch4_dump.dat n2o_dump.dat
24	O	u	MP2-PRD-PT	pt_out.dat
25	O	u	MP2-PRD-VMRxx	o3_out.dat h2o_out.dat hno3_out.dat n2o_out.dat ch4_out.dat
26	O	u	MP2-ANN-PT	pt_annot.dat
27	O	u	MP2-ANN-VMRxx	o3_annot.dat h2o_annot.dat hno3_annot.dat n2o_annot.dat ch4_annot.dat

In general each data set corresponds to a file. It is however possible that one data set corresponds to more than one file. This happens for example when a multiplicity of files having the same format is needed to distinguish between the different retrievals (p, T and VMRxx retrievals) or to distinguish MWs and gases in the cases of the irregular grids and lookup tables.

3. High level architecture of the ORM

The final goal of the ORM program is to retrieve atmospheric pressure and temperature distributions and Volume Mixing Ratio (VMR) profiles of the five high priority chemical species measured by MIPAS. The code performs six retrievals: the first is called p,T retrieval, the other five retrievals are called VMR retrievals and are performed by the same VMR retrieval module. VMR retrievals must be performed after the p,T retrieval has been completed.
The baseline sequence of operations determined by data flow reported in Fig. 1 is presently assumed. The time sequence of the six retrievals is established by the need of exchanging the internal data sets described in Sect. 2.1.4. In the ORM code Vers.2 (or later) the 6 retrievals are performed by a single main program which contains the calls to p,T and VMR retrieval modules as shown in the logical scheme of Fig.1 (see also AD7).

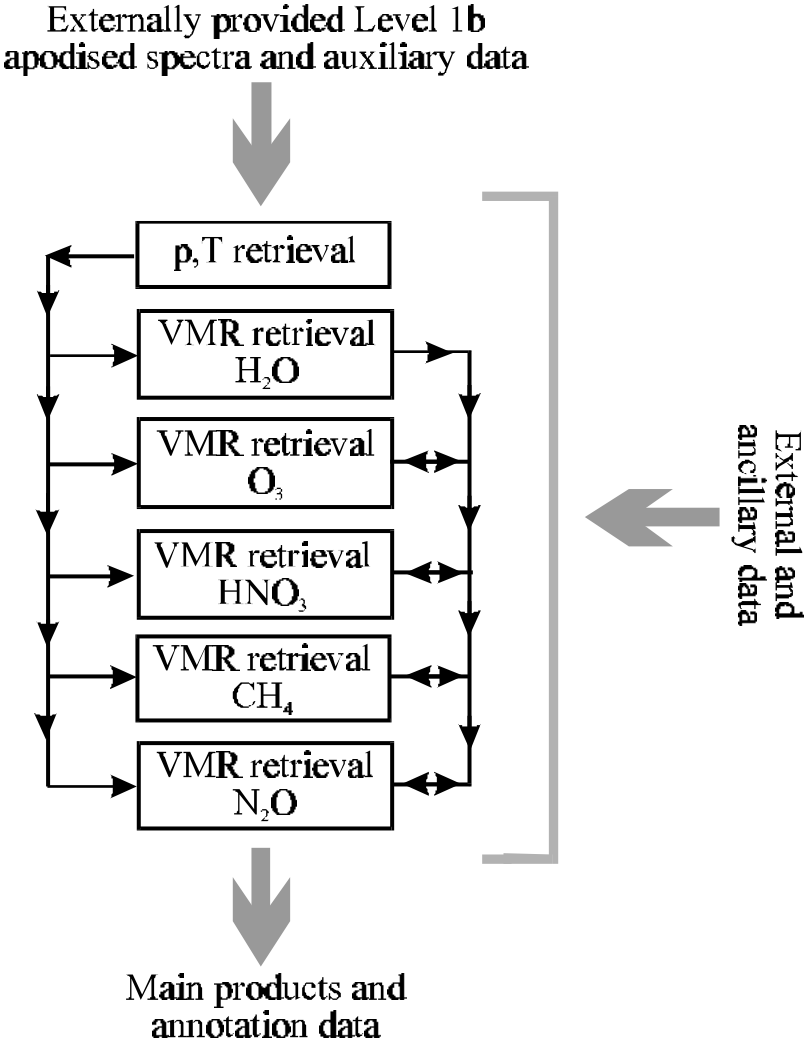

Fig. 1 ORM processor scheme.

The input of the ORM consists of the inputs of its single modules (p,T and VMR retrieval modules), in particular, input data include:

- Auxiliary data
- Externally provided data files from Level 1b
- Ancillary data
- External data

Main products of the ORM:
- Values of tangent pressures of the different sweeps and temperature profile as a function of tangent pressure
- VMR, concentration and column profiles of the five high priority species as a function of tangent pressure
- Profiles of atmospheric continuum cross sections
- Product confidence data.

Each retrieved profile is provided together with its variance-covariance matrix.

Annotation data:
- Intermediate results of the calculations, useful for tests, validation and description of Level 2 main products.

3.1. Interfaces of p, T retrieval module

3.1.1 Input files

The input data of p, T retrieval program are read from files at the beginning of the run by the 'input' subroutine (the source code is reported in Appendix A).
The following files are the inputs of p,T retrieval module:

- observ_pt.dat
- settings_pt.dat
- mwoccmat_pt.dat
- vcm_fm_pt.dat (*)
- apod.dat
- vcm_los.dat
- lookup_[*mw-id_gas*].dat
- in_pres.dat
- in_temp.dat
- in_cont_pt.dat
- in_vmr_prof.dat
- in_alt.dat
- spect_db_pt.dat
- vib_temp_pt.dat (*)
- irrgrid_[*mwname*].dat
- fixed_p_grid.dat (*)
- ext_pt.dat (*)
- ext_cont_pt.dat (*)
- instr_off_pt.dat (*)

(*) This file is not operational in the version 2.3 of the scientific software, not used.

3.1.2 Output files

Output files of p,T retrieval module are:

Main products and annotation data files:
pt_out.dat
pt_annot.dat

Internal data files: pt_dump.dat

3.2. Interfaces of VMR retrieval module

3.2.1 Input files

The input data of VMR retrieval module are read from files at the beginning of the run by the 'input' subroutine (the source code is reported in Appendix B).
The following files are the inputs of VMR retrieval module:

- observ_xx.dat
- settings_xx.dat
- mwoccmat_xx.dat
- vcm_fm_xx.dat (*)
- apod.dat
- lookup_[*mw-id_gas*].dat
- in_pres.dat
- in_temp.dat
- in_cont_xx.dat
- in_vmr_prof.dat
- in_alt.dat
- spect_db_xx.dat
- vib_temp_xx.dat (*)
- irrgrid_[*mwname*].dat
- fixed_p_grid.dat (*)
- ext_vmr_xx.dat (*)
- ext_cont_xx.dat (*)
- instr_off_xx.dat (*)
- pt_dump.dat
- yy_dump.dat

(*) This file is not operational in the version 2.3 of the scientific software, not used.

(Note: in this list of files it has been adopted: xx = H_2O, O, HNO_3, CH_4, N_2O and yy ≠ xx)

3.2.2 Output files

Output files of VMRxx retrieval module are:

Main products and annotation data files:
xx_out.dat
xx_annot.dat

Internal data files: xx_dump.dat

Appendix A

Routine used for reading input files of p,T retrieval: (~/ORM_V2.3/finput_pt.f)

```
      subroutine input_pt
      implicit none
      include 'parameters_pt.inc'
      integer*4 ilimb,nailsdp,nselmw,iept,nswco2,ninterpol,imaingas,
     &          ipro,nrd,imxiterg,imxiterm
      integer*4 nsam(imxmw),nspikes(imxgeo)
      integer*4 ifspmw(imxmw),iline(imxmw)
      integer*4 ilookupmw(imxmw),ngas(imxmw),ih(imxgmw,imxmw)
      integer*4 ioutin(imxlin,imxmw),iiso(imxlin,imxmw),
     &          icode(imxlin,imxmw)
      integer*4 igas,igashi(imxgas),igasmw(imxmw),
     &          igasnr(imxgas,imxmw)
*
      real*8 rzerof,rlat,rearad,rails(imxilc,imxmw),rnoise(imxmw,imxgeo)
     &       ,robs(imxi,imxgeo,imxmw),rconint(imxlmb,imxmw),
     &       rztang(imxgeo)
      real*8 rconvc(3),rmaxtv1,rmaxtv2,rzt12,rhwvar,rincz,redfact,
     &       rhw0ref,rexphref,rwmolref,rulatm,rbase,rsl
      real*8 rcprof(imxpro,imxmw)
      real*8 rzprof(imxpro),rpprof(imxpro),rtprof(imxpro),
     &       rvmrprof(imxpro,imxgas)
      real*8 rint0(imxlin,imxmw),rhw0(imxlin,imxmw),relow(imxlin,imxmw),
     &    rexph(imxlin,imxmw),ruplin(imxlin,imxmw),rlolin(imxlin,imxmw)
      real*4 rapod(imxapo)
      integer*4 napod,ifco
      character smw(imxmw)*6
      logical lmgas(imxgmw,imxmw),lirrgridmw(imxmw),lirrgrid
       real*8 dlatp(imxgeo),dlongp(imxgeo),daltp(imxgeo),dazima(imxgeo),
     &       deleva(imxgeo),dlatt(imxgeo),dlongt(imxgeo),
     &       dlocer(imxgeo)
      real*8 dsilin(imxlin,imxmw),rwmol(imxhit,imxism)
      real*8 delta,dstep,dsigm0,deps
      real*8 rucl,rzc0,rperc,
     &       rlambdain,rlambdadiv,rlambdamul
      real*8 rdzeng(imxlmb),rinvclos(imxlmb,imxlmb)
*
      character squal(imxgeo)*1,slab(imxmw)*25,sname(imxgas)*25,sdol*25,
     &          spare*150,sversn*25,siod*80,sidir*90,sodir*90

      integer*4 iodl
*
* siod = path for the I/O directories
*
      logical lookupc,lfit(imxlmb),lokku(imxgeo,imxmw)
      logical lextinf1,lifend,lextinf2,lextinf3,lextinf4
*
      common/observ_p/ ilimb,nselmw,nailsdp,nsam,nspikes,rzerof,rails,
     &                 rnoise,robs,rlat,rearad,dlatp,dlongp,daltp,dazima,
     &                 deleva,dlatt,dlongt,rztang,dlocer
      common/chobserv_p/ squal,slab
*
      common/settings_p/ iept,nswco2,ninterpol,imaingas,nrd,rconvc,
     &                   imxiterg,imxiterm,rmaxtv1,
     &                   rmaxtv2,rzt12,rhwvar,rincz,redfact,rhw0ref,
     &                   rexphref,rwmolref,rulatm,rbase,rsl,delta,
     &                   dstep,dsigm0,deps,ifco,rucl,rzc0,rperc,
```

```
     &              rlambdain,rlambdadiv,rlambdamul
      common/chsettings_p/ sdol,spare
      common/lsettings_p/ lookupc,lextinf1,lifend,lextinf2,
     &               lextinf3,lextinf4,lfit
*
      common/mwoccmat_p/ ifspmw,iline,rconint
      common/mwlookup_p/ ilookupmw,smw,ngas,ih,lmgas
      common/lmwoccmat_p/ lokku
*
      common/vcmlos_p/ rdzeng,rinvclos
      common/chvcmlos_p/ sversn
*
      common/molec_p/ igas,igashi,igasnr,igasmw
*
      common/molweight_p/rwmol
      common/irrgrid_p/ lirrgridmw,lirrgrid
*
      common/inalt_p/ ipro,rzprof
*
      common/inpres_p/ rpprof
*
      common/intemp_p/ rtprof
*
      common/incont_p/ rcprof
*
      common/invmr_p/ rvmrprof
      common/chinvmr_p/ sname
*
      common/spect_p/ dsilin,rint0,rhw0,relow,rexph,ruplin,rlolin,
     &          ioutin,iiso,icode
      common/apod_f_p/napod,rapod

*                              OPENING FILES
*
* observations
      open(11,file=sidir(1:iodl)//'observ_pt.dat',status='old')
*
* settings
      open(12,file=sidir(1:iodl)//'settings_pt.dat',status='old')
*
* microwindows + occupation matrix
      open(13,file=sidir(1:iodl)//'mwoccmat_pt.dat',status='old')
*
* LOS engineering information:
      open(14,file=sidir(1:iodl)//'vcm_los.dat',status='old')
*
* cross sections look-up table
*
* next open is activated only if lookup tables are used for cross sections
* in that case delete "c" in column 1
*
* initial P profile
      open(16,file=sidir(1:iodl)//'in_pres.dat',status='old')
*
* initial T profile
      open(17,file=sidir(1:iodl)//'in_temp.dat',status='old')
*
* initial continuum profiles
      open(18,file=sidir(1:iodl)//'in_cont_pt.dat',
     &     status='old')
```

```
*
* initial VMR profiles
      open(19,file=sidir(1:iodl)//'in_vmr_prof.dat',status='old')
*
* altitudes of initial profiles
      open(20,file=sidir(1:iodl)//'in_alt.dat',status='old')
*
* spectroscopic data
      open(21,file=sidir(1:iodl)//'spect_db_pt.dat',status='old')
*  Apodisation function in interferogram domain
       open(31,file=sidir(1:iodl)//'apod.dat',status='old')
* output annotations
      open(30,file=sodir(1:iodl)//'pt_annot.dat',status='unknown')
*
      call r_observ_pt
      call r_settings_pt(ilimb)
      call r_mwoccmat_pt(nselmw,ilimb)
      call r_infolos(ilimb,imxlmb,rdzeng,rinvclos,
     &     lextinf1,lifend)
      call r_inalt_pt
      call r_inpres_pt(ipro)
      call r_intemp_pt(ipro)
      call r_incont_pt(nselmw,ipro)
      call uplimit_pt(rzprof,rztang,rbase,rulatm)
      call r_spect_pt(nselmw,iline,rulatm)
      call wmol_pt(rwmol)
      call inigas_pt(nselmw,iline,icode,imaingas,
     &           igas,igashi,igasmw,igasnr)
      call r_invmr_pt(ipro,igas,igashi)
      call r_lookup_pt(nselmw,igasmw,igashi,lookupc,
     &                 ilookupmw,ngas,ih,lmgas)
       call r_apod_pt

*
      return
      end
*
      subroutine r_observ_pt
* subroutine to read file of observations
      implicit none
      include 'parameters_pt.inc'
      integer*4 j,k,l,nf,ilimb,nailsdp,nselmw
      integer*4 nsam(imxmw),nspikes(imxgeo)
      real*8 rzerof,rlat,rearad,rails(imxilc,imxmw),rnoise(imxmw,imxgeo)
     &       ,robs(imxi,imxgeo,imxmw),rztang(imxgeo)
      real*8 dlatp(imxgeo),dlongp(imxgeo),daltp(imxgeo),dazima(imxgeo),
     &       deleva(imxgeo),dlatt(imxgeo),dlongt(imxgeo),
     &       dlocer(imxgeo)
      character squal(imxgeo)*1,slab(imxmw)*25
      common/observ_p/ ilimb,nselmw,nailsdp,nsam,nspikes,rzerof,rails,
     &             rnoise,robs,rlat,rearad,dlatp,dlongp,daltp,dazima,
     &             deleva,dlatt,dlongt,rztang,dlocer
      common/chobserv_p/ squal,slab
      nf=11
      call skip_pt(nf)
      read(nf,100) ilimb
* ilimb = number of sweeps in to the scan which is going to be processed (i5)
      call BLIND_pt(ilimb,'imxgeo','geometries in r_observ')
      call skip_pt(nf)
      read(nf,100) nselmw
```

```
* nselmw = total number of MW's selected for this retrieval
      call BLIND_pt(nselmw,'imxmw','MW,s in r_observ')
      call skip_pt(nf)
      read(nf,110) rzerof

      write(*,*)'rzerof=', rzerof

      if(rzerof .le. 0.1) stop 'rzerof has a wrong value'

* rzerof = zero filling expressed as the ratio between measured and
* transformed interferogram (f10.5)
      call skip_pt(nf)
      read(nf,100) nailsdp
* nailsdp = number of AILS data points (i5)
      call BLIND_pt(nailsdp,'imxilc','points for rails in r_observ')
      do 10 k=1,nselmw
        call skip_pt(nf)
        read(nf,'(a25)') slab(k)
* slab(k) = label for MW k (a25)
        call skip_pt(nf)
        read(nf,100) nsam(k)
* nsam(k) = number of spectral data points for MW k (i5)
        call BLIND_pt(nsam(k),'imxi','observed points in r_observ')
        call skip_pt(nf)
        read(nf,120)(rails(l,k),l=1,nailsdp)
   10 continue
* rails = apodized instrument line shape for all selected MW's (8g10.4)
      do 20 j=1,ilimb
       call skip_pt(nf)
       read(nf,'(a1)') squal(j)
* squal(j) = quality indicator of sweep j (a1)
       call skip_pt(nf)
       read(nf,100) nspikes(j)
* nspikes(j) = number of detected spikes in sweep j (i5)
       call skip_pt(nf)
       read(nf,110) dlatp(j),dlongp(j),daltp(j)
* dlatp(j), dlongp(j), daltp(j) = latitude, longitude, altitude defining the
*                          geolocation of platform at sweep j (3f10.5)
       call skip_pt(nf)
       read(nf,110) dazima(j),deleva(j)
* dazima(j), deleva(j) = azimuth and elevation angles of sweep j (2f10.5)
       call skip_pt(nf)
       read(nf,110) dlatt(j),dlongt(j),rztang(j)
* dlatt(j), dlongt(j), rztang(j) = latitude, longitude, altitude defining
*                          geolocation of tangent point of sweep j (3f10.5)
       call skip_pt(nf)
       read(nf,110) dlocer(j)
* dlocer(j) = local earth's radius curvature for sweep j (f10.5)
       do 30 k=1,nselmw
         call skip_pt(nf)
         read(nf,120) rnoise(k,j)

         if(rzerof.le.0.99) then
          rnoise(k,j)=rnoise(k,j)/rzerof
          write(*,*)'rzerof < 1: the NESR has been corrected '
         end if

* rnoise(k,j) = noise for MW k in sweep j (g10.4)
         call skip_pt(nf)
         read(nf,120) (robs(l,j,k),l=1,nsam(k))
```

```
* robs = observed spectral data points for MW k of sweep j (8g10.4)
   30  continue
   20 continue
      rlat=0.0
      rearad=0.0
      do 40 j=1,ilimb
       rlat=rlat+dlatt(j)
       rearad=rearad+dlocer(j)
40     continue
      rlat=rlat/float(ilimb)
      rearad=rearad/float(ilimb)
  100 format(i5)
  110 format(3f10.5)
  120 format(8g10.4)
      return
      end
*
      subroutine r_settings_pt(ilimb)
* subroutine to read file of settings
      implicit none
      include 'parameters_pt.inc'
      integer*4 nf,j,ilimb,iept,nswco2,ninterpol,imaingas,nrd,
     &          imxiterg,imxiterm,ifcO
      real*8 rconvc(3),rmaxtv1,rmaxtv2,rzt12,rhwvar,rincz,redfact,
     &       rhw0ref,rexphref,rwmolref,rulatm,rbase,rsl
      real*8 delta,dstep,dsigm0,deps,drd,rucl,rzc0,rperc,
     &                rlambdain,rlambdadiv,rlambdamul
      character sdol*25,spare*150
      logical lookupc,lextinf1,lifend,lextinf2,lextinf3,
     &       lextinf4,lfit(imxlmb),lirrgrid,lirrgridmw(imxmw)
      common/settings_p/ iept,nswco2,Ninterpol,imaingas,nrd,rconvc,
     &                imxiterg,imxiterm,rmaxtv1,
     &                rmaxtv2, rzt12,rhwvar,rincz,redfact,rhw0ref,
     &                rexphref,rwmolref,rulatm,rbase,rsl,delta,
     &                dstep,dsigm0,deps,ifco,rucl,rzc0,rperc,
     &                rlambdain,rlambdadiv,rlambdamul
      common/chsettings_p/ sdol,spare
      common/lsettings_p/ lookupc,lextinf1,lifend,lextinf2,lextinf3,
     &                  lextinf4,lfit
      common/irrgrid_p/ lirrgridmw,lirrgrid
      nf=12
      call skip_pt(nf)
      read(nf,100) delta
      write(*,100) delta
* delta = frequency step that simulates infinitesimal spectral resolution
*         in forward simulations (f10.5)
      call skip_pt(nf)
      read(nf,100) dstep
      write(*,100) dstep
* dstep = frequency spacing between observed spectral data points (f10.5)
      call skip_pt(nf)
      read(nf,'(a25)') sdol
      write(*,*)sdol
* sdol = definition of the output level (a25)
      call skip_pt(nf)
      read(nf,100)(rconvc(j),j=1,3)
      write(*,*) rconvc
* rconvc = vector of convergence criteria to stop iterations (3f10.5)
* rconvc(1) = variation of chi**2 between two consecutive iterations
* rconvc(2) = variation between calculated chi**2 and linear chi**2
```

```
* rconvc(3) = largest variation of parameters between consecutive iterations
      call skip_pt(nf)
      read(nf,120) imxiterg
* imxiterg = maximum number of Gauss-Newton macro-iterations
      call skip_pt(nf)
      read(nf,120) imxiterm
* imxiterm = maximum number of Marquard micro-iterations
      call skip_pt(nf)
      read(nf,110) lookupc
      write(*,*) lookupc
*  lextinf1 = switch for using a-priori info on LOS
*  lifenf   = switch for using a-priori info on LOS after the retrieval
*  lextinf2 = switch for using a-priori info on atm. continuum
*  lextinf3 = switch for using a-priori info on instr. offset
*  lextinf4 = switch for using a-priori info on temperature
      call skip_pt(nf)
      read(nf,'(5l2)') lextinf1,lifend,lextinf2,lextinf3,lextinf4
      call skip_pt(nf)
      read(nf,100) rmaxtv1,rmaxtv2,rzt12
* parameters_pt.incor the layering of the atmosphere (3f10.5)
* rmaxtv1 = max. allowed T variation (K) between levels: 0 < altitude < rzt12
* rmaxtv2 = max. allowed T variation (K) between levels: rzt12<altitude<rulatm
* rzt12 = altitude (km) where the thresholds rmaxtv1 and rmaxtv2 are exchanged
      call skip_pt(nf)
      read(nf,100) rhwvar
* rhwvar = max half-width change allowed between levels for ref. transiton (f10.5)
      call skip_pt(nf)
      read(nf,100) rincz
      write(*,100) rincz
* rincz = max thickness of the layers used for modelling the atmosphere (f10.5)
      call skip_pt(nf)
      read(nf,100) redfact
* redfact = reduction factor for rincz to produce thinner layers
      call skip_pt(nf)
      read(nf,100) dsigm0,rhw0ref,rexphref,rwmolref
* data relative to the transition that is used for layering criterion (4f10.5)
* dsigm0 = frequency position
* rhw0ref = half-width at the reference T and P
* rexphref = T dependence of half-width
* rwmolref = molecular weight
      call skip_pt(nf)
      read(nf,100) rulatm
* rulatm = boundary of the atmosphere (km) (f10.5)
      call skip_pt(nf)
      read(nf,100) deps
* deps = convergence criterion for the C.G. integrals (f10.5)
      call skip_pt(nf)
      read(nf,120) iept
* iept = actual number of extra paths (i5)
      call skip_pt(nf)
      read(nf,120) nswco2
* nswco2 = switch for the calculation of the CO2 chi factor (i5)
*        =0 no factor, =1 n2/o2 broadening, =2 n2 broadening only
* NOTE this switch is no longer active but is maintained in the case of future
*      applications
      call skip_pt(nf)
      read(nf,120) ninterpol
*   ninterpol = switch for the interpolation on the cross-sections (i5)
*             = -1: no interpolation, all cross-sections recalculated
*             =  0: all those above the lowest geometry are interpolated
```

```
*           =  1: new calculation only of tangent-layer
*           =  2: new calculation for tangent-layer and the one above
      call skip_pt(nf)
      read(nf,100) rbase,rsl
* parameters_pt.incor the FOV convolution (2f10.5)
* rbase = greater base of the trapetium FOV-function (km)
* rsl   = half difference between bases of the trapetium FOV-function (km)
      call skip_pt(nf)
      read(nf,120)imaingas
* imaingas = HITRAN code of the main molecule of the retrieval (i5)
      call skip_pt(nf)
      read(nf,110)(lfit(j),j=1,ilimb)
* lfit(j) = logical vector that identifies the tangent altitudes which
* correspond to fitted parameters (40l2)

* Switch for fitting atmospheric continuum and offset (ifco):
*      ifco = 2 -->    p,T, continuum and offset are fitted
*      ifco = 1 -->    p,T and continuum are fitted
*      ifco = 0 -->    only p,T are fitted
      call skip_pt(nf)
      read(nf,*) ifco
* Upper continuum limit (rucl): atmospheric continuum is not fitted at the
* sweeps having tangent altitude > rucl. Units: km
      call skip_pt(nf)
      read(nf,*) rucl
* Altitude above which the continuum is set = 0
      call skip_pt(nf)
      read(nf,*) rzc0
* Max. relative distance between MWs to be considered
* as having the same continuum:
      call skip_pt(nf)
      read(nf,*) rperc
* Parameters controlling the evolution of Marquardt damping factor:
*     rlambdain = initial value of rlambda
*     rlambdadiv = factor used to decrease rlambda at each Gauss iteration
*     rlambdamul = factor useed to increase rlambda at each Marquardt iteration
      call skip_pt(nf)
      read(nf,*) rlambdain, rlambdadiv, rlambdamul
      call skip_pt(nf)
      read(nf,*) lirrgrid
      write(*,*)lirrgrid
      call skip_pt(nf)
      read(nf,'(a150)') spare
* spare = spare field for retrieval configuration
      drd=dstep/delta
      nrd=nint(drd)
      if(abs(drd-float(nrd)).gt.0.00001) then
       write(*,*)'sono qui'
       write(*,'(f12.6)')'ratio dstep/deltas not integer program stopped
     & nrd=',dstep/delta
       stop
      end if
  100 format(4f10.5)
  110 format(40l2)
  120 format(i5)
130   format(40l2)
      return
      end
*
      subroutine r_mwoccmat_pt(nselmw,ilimb)
```

```
* subroutine to read file of microwindows and occupation matrix
      implicit none
      include 'parameters_pt.inc'
      integer*4 nf,j,k,nselmw,iordmw,ilimb,ifspmw(imxmw),nsam,
     &          iline(imxmw),ngas(imxmw),ih(imxgmw,imxmw),
     &          ilookupmw(imxmw),i
      real*8 rmin,rmax,rconint(imxlmb,imxmw)
      character slab*10,smw(imxmw)*6,cc*2
      logical lokku(imxgeo,imxmw),lmgas(imxgmw,imxmw),
     &        lirrgridmw(imxmw),lirrgrid
      common/mwoccmat_p/ ifspmw,iline,rconint
      common/mwlookup_p/ ilookupmw,smw,ngas,ih,lmgas
      common/lmwoccmat_p/ lokku
      common/irrgrid_p/ lirrgridmw,lirrgrid
      nf=13
      call skip_pt(nf)
      do 10 j=1,nselmw
        read(nf,130) iordmw,slab,ifspmw(j),nsam,rmin,rmax,
     &               iline(j),ilookupmw(j),ngas(j),
     &               lirrgridmw(j)
        if(ngas(j).ne.0) then
         write(cc,'(i2)')ngas(j)
         read(nf,('//cc//'i3)') (ih(i,j),i=1,ngas(j))
        end if
       smw(j)=slab(5:10)

      call blind_pt(iline(j),'imxlin','transitions in r_mwoccmat')
10    continue
* data for the MW's definition (i5,a10,2i10,2f10.5,i5)
* ifspmw(j) = index of the first sampling point of MW j
* NOTE: the sampling point at frequency=0 has index=1
* iline(j) = number of transitions in the spectroscopic database for MW j
*
* iordmw,slab,nsam,rmin,rmax = are scratch fields reporting:
* iordmw = progressive number
* slab = label of the MW as read in the OBSERV_PT.DAT file
* nsam   = number of sampling points for the MW
* rmin,rmax = minimum and maximum altitude coverage of the MW
      do 20 j=1,nselmw
        call skip_pt(nf)
        read(nf,110)(rconint(k,j),k=1,ilimb)
20    continue
* rconint(k,j) = spectral interval, around MW j, in which atmospheric continuum
* can be assumed to vary linearly at geometry k (8f10.5)
      call skip_pt(nf)
      do 30 j=1,ilimb
      read(nf,120)(lokku(j,k),k=1,nselmw)
30    continue
* lokku(j,k) = ocupation matrix element for geometry j and MW k (40l2)
100   format(2i5)
110   format(8f10.5)
120   format(40l2)
130   format(i5,a10,2i10,2f10.5,i5,2i2,l2)
      return
      end
*************************************************************************
* Subroutine used to read LOS engineering information.
*
      subroutine r_infolos(ilimb,imxlmb,rdzeng,rinvclos,
     &                     lextinf1,lifend)
```

```
*
      implicit none
      integer*4 imxlmb, ilimb, nf, i, j
      real*8 rdzeng(imxlmb),rinvclos(imxlmb,imxlmb)
      logical lextinf1,lifend
* If LOS info ahs to be used the file is read:
      if (lextinf1.or.lifend) then
        nf=14
        call skip_pt(nf)
        read(nf,*)(rdzeng(j),j=1,ilimb-1)
        call skip_pt(nf)
        do i=1,ilimb-1
          read(nf,'(8e12.4)')(rinvclos(i,j),j=1,ilimb-1)
        end do
      else
* If LOS info is NOT to be used the variables are set to zero:
        do i=1,ilimb
          rdzeng(i) = 0.0d0
          do j=1,ilimb
            rinvclos(i,j) = 0.0d0
          end do
        end do
      end if

      return
      end
*
*
      subroutine r_lookup_pt(nselmw,igasmw,igashi,lookupc,
     &                ilookupmw,ngas,ih,lmgas)
* subroutine for defining the variables that are used in cross for using
* or not using look-up tables
      implicit none
      include 'parameters_pt.inc'
      integer*4 j,i,nselmw,mgas,igasmw(imxmw),
     &          ilookupmw(imxmw),ihit,igashi(imxgas),
     &          igasnr(imxgas,imxmw),ih(imxgmw,imxmw),
     &          ngas(imxmw)
      logical lmgas(imxgmw,imxmw),lookupc



       do 10 j=1,nselmw

        do 15 mgas=1,igasmw(j)
          lmgas(mgas,j)=.false.
 15     continue

        if(lookupc) then
         if(ilookupmw(j).eq.1) then
           do 20 mgas=1,igasmw(j)
            ihit=igashi(igasnr(mgas,j))
             do 25 i=1,ngas(j)
               if (ihit .eq. ih(i,j)) goto 20
 25          continue
            ilookupmw(j)=2
            lmgas(mgas,j)= .true.
 20        continue
         end if
        else
```

```
        ilookupmw(j)=0
       end if

 10   continue
      return
      end

      subroutine r_inalt_pt
* subroutine to read file of the altitudes of initial profiles
      implicit none
      include 'parameters_pt.inc'
      integer*4 nf,j,ipro
      real*8 rzprof(imxpro)
      character sident*25
      common/inalt_p/ ipro,rzprof
      nf=20
      call skip_pt(nf)
      read(nf,'(a25)') sident
* sident = alphanumeric identifier of the file (a25)
      call skip_pt(nf)
      read(nf,100) ipro
* ipro = number of elements of the vector of altitudes (i5)
      call blind_pt(ipro,'imxpro','altitudes in r_inalt')
      call skip_pt(nf)
      read(nf,110)(rzprof(j),j=1,ipro)
* rzprof(j) = altitude to which element j of T, P and VMR profiles are
*           referred to (8f10.5)
  100 format(i5)
  110   format(8f10.5)
      return
      end
*
      subroutine r_inpres_pt(ipro)
* subroutine to read file of initial P profile
      implicit none
      include 'parameters_pt.inc'
      integer*4 nf,j,ipro
      real*8 rpprof(imxpro)
      character sident*25
      common/inpres_p/ rpprof
      nf=16
      call skip_pt(nf)
      read(nf,'(a25)') sident
* sident = alphanumeric identifier of the file (a25)
      call skip_pt(nf)
      read(nf,*)(rpprof(j),j=1,ipro)
* rpprof(j) = value of P at altitude j of the initial profile (8f10.5)
      return
      end
*
      subroutine r_intemp_pt(ipro)
* subroutine to read file of initial T profile
      implicit none
      include 'parameters_pt.inc'
      integer*4 nf,j,ipro
      real*8 rtprof(imxpro)
      character sident*25
      common/intemp_p/ rtprof
      nf=17
      call skip_pt(nf)
```

```
      read(nf,'(a25)') sident
* sident = alphanumeric identifier of the file (a25)
      call skip_pt(nf)
      read(nf,100)(rtprof(j),j=1,ipro)
* rpprof(j) = value of T at altitude j of the initial profile (8f10.5)
 100   format(8f10.5)
      return
      end
*
      subroutine r_incont_pt(nselmw,ipro)
* subroutine to read the file of initial continuum profiles
      implicit none
      include 'parameters_pt.inc'
      integer*4 nf, nselmw,j,k,ipro
      real*8 rcprof(imxpro,imxmw)
      character*25 smwlabel
      common /incont_p/ rcprof
      nf=18
      call skip_pt(nf)
      do 10 j=1,nselmw
        read (nf,'(a25)') smwlabel
* smwlabel = scratch label of the microwindow (a25)
        read(nf,110)(rcprof(k,j),k=1,ipro)
* rcprof(k,j) = continuum cross section for MW j at altitude k (8e10.3)
   10 continue
  110 format(8E10.3)
      return
      end
*
      subroutine r_spect_pt(nselmw,iline,rulatm)
* subroutine to read file of spectroscopic data
      implicit none
      include 'parameters_pt.inc'
      integer*4 nf,j,k,nselmw,iline(imxmw)
      integer*4 ioutin(imxlin,imxmw),iiso(imxlin,imxmw),
     &          icode(imxlin,imxmw)
      real*8 rint0(imxlin,imxmw),rhw0(imxlin,imxmw),relow(imxlin,imxmw),
     &    rexph(imxlin,imxmw),ruplin(imxlin,imxmw),rlolin(imxlin,imxmw)
      real*8 dsilin(imxlin,imxmw),rulatm
      character smwlabel*25
      common/spect_p/ dsilin,rint0,rhw0,relow,rexph,ruplin,rlolin,
     &             ioutin,iiso,icode
      nf=21
      do 10 j=1,nselmw
       call skip_pt(nf)
       read(nf,'(a25)') smwlabel
* smwlabel = scratch label of the microwindow (a25)
       do 20 k=1,iline(j)
        read(nf,100) dsilin(k,j),rint0(k,j),rhw0(k,j),relow(k,j),
     &              rexph(k,j),ioutin(k,j),iiso(k,j),icode(k,j),
     &              ruplin(k,j),rlolin(k,j)

         if (rlolin(k,j).le.8.)then
          rlolin(k,j)=0.
         else
          rlolin(k,j)=rlolin(k,j)-.5
         end if

        if(abs(ruplin(k,j)-70.) .le. 1.e-4)then
          ruplin(k,j)=rulatm
```

```
        else
          ruplin(k,j)=ruplin(k,j)+10.
        end if


* spectroscopic data for transition k of MW j
* (f11.5,2e11.3,2f11.4,2x,i1,2x,i2,1x,i2,2f7.2)
*
* dsilin = line position
* rint0  = line strength
* rhw0   = pressure broadening
* relow  = energy of starting level
* rexph  = exponent for T dependence of rhw0
* ioutin = control on continuum treatment
* iiso   = isotope code
* icode  = hitran code of the molecule
* ruplin = highest altitude where the line has to be used
* rlolin = lowest altitude where the line has to be used
*
20    continue
10    continue
100   format(f12.6,(1pe11.3),0pe11.3,2f11.4,2x,i1,2x,i2,1x,i2,2f7.2)
      return
      end
*
      subroutine r_invmr_pt(ipro,igas,igashi)
* subroutine to read file of initial VMR profiles
      implicit none
      include 'parameters_pt.inc'
      integer*4 nf,j,k,l,igc,ipro,igas,igashi(imxgas)
      real*8 rvmrprof(imxpro,imxgas)
      character sident*25,sname(imxgas)*25
      common/invmr_p/ rvmrprof
      common/chinvmr_p/ sname
      nf=19
      do 10 j=1,igas
        rewind nf
        call skip_pt(nf)
        read(nf,'(a25)') sident
        call skip_pt(nf)
        do 10 k=1,igashi(j)
         read(nf,100) igc,sname(j)
* igc   = progressive number denoting hitran gas code
* sname = name of the gas
         read(nf,110)(rvmrprof(l,j),l=1,ipro)
   10 continue
100   format(i5,1x,a25)
  110 format(8e10.3)
      return
      end
subroutine skip_pt(nf)
* subroutine to skip comment lines on read files
* convention is that: at least one comment line appears before a read statement
*                     last comment line starts with a character # in column 1
*
      implicit none
      character*1 sendc
      integer*4 nf
      sendc=' '
      do while (sendc.ne.'#')
```

```
      read(nf,'(a1)') sendc
     end do
     return
     end
*******************************************************************************
*  SUBROUTINE              : uplimit
*  CREATED BY              : michael hoepfner
*  DATE OF CREATION        : 5.6.96
*  DATE OF LAST MODIFICATION  :
*  MODIFIED BY             :
*  LAST MODIFICATION BY    :
*
*  DESCRIPTION             :
*  Determination of the upper atmospheric limit to be calculated.
*  Changes the input-variable 'rulatm' if the highest altitude to be
*  considered from the line-database is lower than the input 'rulatm'
*
*  INPUTS:
*   nselmw         total number of selected microwindows for the retrieval
*   iline(imxmw)    number of lines in each microwindow
*   ruplin(imxlin,imxmw) upper limit where the line has to be considered
*   rzprof(imxpro)  vector of altitudes Z to which rtprof, rpprof
*                and rvmrprof are referred [km]
*   rztang(imxgeo)  vector containing the engineering values of
*                tangent altitudes
*   rbase          greater base of trapezium of Field of View function
*   rulatm          max. upper limit of the atmosphere (km) from settings
*
*  OUTPUTS:
*   rulatm          upper limit of the atmosphere (km) (for radiative
*                transfer calculations)
*
*******************************************************************************
*******************************************************************************
     subroutine uplimit_pt(rzprof,rztang,rbase,rulatm)
     implicit none
     include 'parameters_pt.inc'
     real*8 rzprof(imxpro),rztang(imxgeo),
    &       rbase,rulatm
*
* if the new rulatm is less than the highest observation geometry
*  (+2*rbase) it is changed
*
     if (rulatm.lt.rztang(1)+2.*rbase) rulatm=rztang(1)+2.*rbase
*
* if the new rulatm is higher than the highest point of the inital profiles
*  it is changed and a warning is written.
*
     if (rulatm.gt.rzprof(1)) then
       rulatm=rzprof(1)
       write(*,*) 'WARNING: The upper limit of the atmosphere'
       write(*,*) ' (rulatm) was higher than the highest altitude of'
       write(*,*) ' the initial profiles. This was corrected, but'
       write(*,*) ' check rulatm in SETTINGS.DAT! '
     end if
*
* test output
*
     write(*,*) '*****************************************************'
     write(*,*) 'Test output from uplimit:'
```

```
      write(*,*) 'The upper limit for the radiative transfer'
     &          //' calculations (rulatm) =',rulatm
      write(*,*) '*****************************************************'
      end
*******************************************************************************
*  SUBROUTINE              : inigas
*  CREATED BY              : michael hoepfner
*  DATE OF CREATION        : 21.5.96
*  DATE OF LAST MODIFICATION  :
*  MODIFIED BY             :
*  LAST MODIFICATION BY    :
*
*  DESCRIPTION             :
*  Initialisation of the variables 'igas, igashi,igasmw,igasnr' that
*  define the two internal numberings of the gases
*
*
*  INPUTS:
*   nselmw         total number of selected microwindows
*   iline(imxmw)    number of lines in each microwindow
*   icode(imxlin,imxmw)   HITRAN molecular code for each line of each MW
*   imaingas           HITRAN code of the main gas of the retrieval
*
*  OUTPUTS:
*   igas          number of different gases for actual retrieval
*   igashi(imxgas)  HITRAN code nr for each gas-number of actual retrieval
*   igasmw(imxmw)   number of gases to be considered for each mw.
*   igasnr(imxgas,imxmw)  retrieval gas number for each internal gas
*                    number of each NW
*
*******************************************************************************
      subroutine inigas_pt(nselmw,iline,icode,imaingas,
     &                igas,igashi,igasmw,igasnr)
      implicit none
      include 'parameters_pt.inc'
      integer*4 nselmw,iline(imxmw),icode(imxlin,imxmw),
     &         igas,igashi(imxgas),igasmw(imxmw),
     &         igasnr(imxgas,imxmw),imaingas,
     &         jmw,kline,mgas,m1,j
      logical lmaingas
*
* initialization, so that the main gas has the retrieval numbering =1
*
      igas=1
      igashi(1)=imaingas
*
*  begin loop over microwindows
*
      do 10 jmw=1,nselmw
*
* initialization, so that the main gas has the microwindow numbering =1
*
       igasmw(jmw)=1
       igasnr(1,jmw)=1
*
* initialization of the test variable, if the main gas has a line in the MW
*
       lmaingas=.false.
*
*  begin loop over the lines of one mircowindow
```

```
*
      do 20 kline=1,iline(jmw)
*
*  test, it the line is a line of the main gas
*
       if (icode(kline,jmw).eq.imaingas) lmaingas=.true.
*
*  the retrieval gas numbering is performed
*
       do 30 mgas=1,igas
        if (igashi(mgas).eq.icode(kline,jmw)) then
          m1=mgas
          goto 40
        end if
30      continue
       igas=igas+1
       call BLIND_pt(igas,'imxgas','Blind stop in inigas')
       igashi(igas)=icode(kline,jmw)
       m1=igas
*
* the internal microwindow gas numbering is performed
*
40      continue
       do 50 mgas=1,igasmw(jmw)
        if (igashi(igasnr(mgas,jmw)).eq.icode(kline,jmw))
     &      goto 20
50      continue
       igasmw(jmw)=igasmw(jmw)+1
       igasnr(igasmw(jmw),jmw)=m1
*
*  end loop over the lines of one mircowindow
*
20     continue
*
*  test, if the main gas is at least one line of the mw
*
      if (.not.lmaingas) then
        print*,'No line of the main gas in microwindow',jmw
        stop
      end if
*
*  test the number of gases per Mw
*
       call blind_pt(igasmw(jmw),'imxgmw','Blind stop in inigas')
*
*  end loop over microwindows
*
10   continue
*
* Test -output
*
     write(*,*) '*****************************************'
     write(*,*) 'Test output of inigas:'
     write(*,*)
     write(*,*) 'Total number of gases:',igas
     write(*,'(a,32i3)') ' HITRAN codes:',(igashi(j),j=1,igas)
     do jmw=1,nselmw
     write(*,*)
     write(*,'(a,i4)') ' Microwindow:',jmw
     write(*,'(a,32i3)') ' Local mw gas number:',(j,j=1,igasmw(jmw))
```

```
      write(*,'(a,32i3)') ' Global gas number  :',
     &              (igasnr(j,jmw),j=1,igasmw(jmw))
      write(*,'(a,32i3)') ' HITRAN gas number  :',
     &              (igashi(igasnr(j,jmw)),j=1,igasmw(jmw))
      end do
      write(*,*)
      write(*,*) '*****************************************'
      end
*******************************************************************************
*******************************************************************************
      subroutine wmol_pt(x)
      implicit none
      include 'parameters_pt.inc'
      real*8 rwmol(imxhit,imxism),x(imxhit,imxism)
      integer*4 j,k
      data
     &(rwmol( 1,j),j=1,4) / 18., 20., 19., 19./
      data
     &(rwmol( 2,j),j=1,8) / 44., 45., 46., 45., 47., 46.,48.,47./
      data
     &(rwmol( 3,j),j=1,5) / 48., 50., 50., 49., 49./
      data
     &(rwmol( 4,j),j=1,5) / 44., 45., 45., 46., 45./
      data
     &(rwmol( 5,j),j=1,6) / 28., 29., 30., 29., 31., 30./
      data
     &(rwmol( 6,j),j=1,3) / 16., 17., 17./
      data
     &(rwmol( 7,j),j=1,3) / 32., 34., 33./
      data
     &(rwmol( 8,j),j=1,3) / 30., 31., 32./
      data
     &(rwmol( 9,j),j=1,2) / 64., 66./
      data
     &(rwmol(10,j),j=1,1) / 46./
      data
     &(rwmol(11,j),j=1,2) / 17., 18./
      data
     &(rwmol(12,j),j=1,1) / 63./
      data
     &(rwmol(13,j),j=1,3) / 17., 19., 18./
      data
     &(rwmol(14,j),j=1,1) / 20./
      data
     &(rwmol(15,j),j=1,2) / 36., 38./
      data
     &(rwmol(16,j),j=1,2) / 80., 82./
      data
     &(rwmol(17,j),j=1,1) /128./
      data
     &(rwmol(18,j),j=1,2) / 51., 53./
      data
     &(rwmol(19,j),j=1,4) / 60., 62., 61., 62./
      data
     &(rwmol(20,j),j=1,3) / 30., 31., 32./
      data
     &(rwmol(21,j),j=1,2) / 52., 54./
      data
     &(rwmol(22,j),j=1,1) / 28./
      data
```

```
     &(rwmol(23,j),j=1,3) / 27., 28., 28./
      data
     &(rwmol(24,j),j=1,2) / 50., 52./
      data
     &(rwmol(25,j),j=1,1) / 34./
      data
     &(rwmol(26,j),j=1,2) / 26., 27./
      data
     &(rwmol(27,j),j=1,1) / 30./
      data
     &(rwmol(28,j),j=1,1) / 28./
      data
     &(rwmol(29,j),j=1,1) / 66./
      data
     &(rwmol(30,j),j=1,1) / 146./
      data
     &(rwmol(31,j),j=1,3) / 34.,36.,35./
      data
     &(rwmol(32,j),j=1,1) / 46./
      data
     &(rwmol(33,j),j=1,1) / 33./
      data
     &(rwmol(34,j),j=1,1) / 16./
      data
     &(rwmol(35,j),j=1,2) / 97.,99./
      data
     &(rwmol(36,j),j=1,1) / 30./
      do 10 j=1,imxhit
        do 20 k=1,imxism
          x(j,k)=rwmol(j,k)
20      continue
10    continue
      end

      subroutine r_apod_pt
* subroutine to read file of apodisation function in the interferogram
* domain
      implicit none
      include 'parameters_pt.inc'
      integer*4 nf,j,napod,n
      real*4 rapod(imxapo)
      real*8 rx
      common/apod_f_p/napod,rapod
      nf=31
      call skip_pt(nf)
      read(nf,*) napod

       call blind_pt(napod,'imxapo','blind-stop in finput_pt.f')

*   Control on the value of napod: IT HAS TO BE (2^n+1), with n=integer
      rx=(log(dble(napod-1)))/log(2.)
      write(*,*)'rx=',rx
      n=nint(rx)
      write(*,*)'n=',n
      if(abs(rx-dble(n)).gt.1.e-6) then
       write(*,*)'program stopped in finput_pt.f'
       write(*,*)'(napod-1) has to be 2^n, with n integer'
       write(*,*)'(napod-1) =',(napod-1)
       stop
      end if
```

```
read(nf,*) (rapod(j),j=1,napod)
close(nf)

return
end
```

Appendix B

Routine used for reading input files of VMR retrieval: (~/ORM_V2.3/finput_vmr.f)

```
      subroutine input_vmr(im)
      implicit none
      include 'parameters_vmr.inc'
      integer*4 ilimb,nailsdp,nselmw,iept,nswco2,ninterpol,imaingas,
     &          ipro,nrd,imxiterg,imxiterm
      integer*4 nsam(imxmw),nspikes(imxgeo)
      integer*4 ifspmw(imxmw),iline(imxmw)
      integer*4 ilookupmw(imxmw),ngas(imxmw),ih(imxgmw,imxmw)
      integer*4 ioutin(imxlin,imxmw),iiso(imxlin,imxmw),
     &          icode(imxlin,imxmw)
      integer*4 igas,igashi(imxgas),igasmw(imxmw),
     &          igasnr(imxgas,imxmw),j,im,j1
*
      real*8 rzerof,rlat,rearad,rails(imxilc,imxmw),
     &       rnoise(imxmw,imxgeo)
     &       ,robs(imxi,imxgeo,imxmw),rconint(imxlmb,imxmw),
     &       rztang(imxgeo)
      real*8 rconvc(3),rmaxtv1,rmaxtv2,rzt12,rhwvar,rincz,redfact,
     &       rhw0ref,rexphref,rwmolref,rulatm,rbase,rsl
      real*8 rvcmlos(imxlmb,imxlmb),rcprof(imxpro,imxmw)
      real*8 rzprof(imxpro),rpprof(imxpro),rtprof(imxpro),
     &       rvmrprof(imxpro,imxgas)
      real*8 rint0(imxlin,imxmw),rhw0(imxlin,imxmw),relow(imxlin,imxmw),
     &    rexph(imxlin,imxmw),ruplin(imxlin,imxmw),rlolin(imxlin,imxmw)
      real*4 rapod(imxapo)
      integer*4 napod,ifco
*
      real*8 dlatp(imxgeo),dlongp(imxgeo),daltp(imxgeo),dazima(imxgeo),
     &       deleva(imxgeo),dlatt(imxgeo),dlongt(imxgeo),
     &       dlocer(imxgeo)
      real*8 dsilin(imxlin,imxmw),rwmol(imxhit,imxism)
      real*8 delta,dstep,dsigm0,deps,rucl,rzc0,rperc,
     &                rlambdain,rlambdadiv,rlambdamul

*
      character squal(imxgeo)*1,slab(imxmw)*25,sname(imxgas)*25,sdol*25,
     &       spare*150,sversn*25
      character*6 smw(imxmw), sidir*90, sodir*90
      integer*4 iodl
      logical lmgas(imxgmw,imxmw),lirrgridmw(imxmw),lirrgrid
*
      logical lookupc,lextinf1,lfit(imxlmb),lokku(imxgeo,imxmw),
     &       lifptret,lifvmret,lextinf2,lextinf3
*
      character*6 sgas(imxhit)
*

      common/observ_v/ ilimb,nselmw,nailsdp,nsam,nspikes,rzerof,rails,
     &             rnoise,robs,rlat,rearad,dlatp,dlongp,daltp,dazima,
     &             deleva,dlatt,dlongt,rztang,dlocer
      common/chobserv_v/ squal,slab
*
      common/settings_v/ iept,nswco2,ninterpol,imaingas,nrd,rconvc,
     &              imxiterg,imxiterm,rmaxtv1,
     &              rmaxtv2,rzt12,rhwvar,rincz,redfact,rhw0ref,
     &              rexphref,rwmolref,rulatm,rbase,rsl,delta,
```

```
     &              dstep,dsigm0,deps,ifco,rucl,rzc0,rperc,
     &              rlambdain,rlambdadiv,rlambdamul
*
      common/chsettings_v/ sdol,spare,sgas
      common/lsettings_v/ lookupc,lextinf1,lextinf2,
     &               lextinf3,lfit,lifptret,lifvmret
*
      common/mwoccmat_v/ ifspmw,iline,rconint
      common/mwlookup_v/ ilookupmw,smw,ngas,ih,lmgas
      common/lmwoccmat_v/ lokku
*
      common/vcmlos_v/ rvcmlos
      common/chvcmlos_v/ sversn
*
      common/molec_v/ igas,igashi,igasnr,igasmw
*
      common/molweight_v/rwmol
      common/irrgrid_v/ lirrgridmw,lirrgrid
*
      common/inalt_v/ ipro,rzprof
*
      common/inpres_v/ rpprof
*
      common/intemp_v/ rtprof
*
      common/incont_v/ rcprof
*
      common/invmr_v/ rvmrprof
      common/chinvmr_v/ sname
*
      common/spect_v/ dsilin,rint0,rhw0,relow,rexph,ruplin,rlolin,
     &            ioutin,iiso,icode
      common/apod_f_v/napod,rapod
*
       common/iopaths/ sidir,sodir,iodl

      data sgas/'h2o','pt','o3','n2o','co','ch4','o2','no','so2','no2',
     *'nh3','hno3','oh','hf','hcl','hbr','hi','clo','ocs','h2co',
     *'hocl','n2','hcn','ch3cl','h2o2','c2h2','c2h6','ph3',
     *'cof2','sf6','h2s','hcooh','ho2','o','clono2','no+'/

*                              OPENING I/O FILES
*
      do j=1,6
       if(sgas(im)(j:j).ne.' ') j1=j
      end do
*
* observations
      open(11,file=sidir(1:iodl)//'observ_'//sgas(im)(1:j1)//'.dat'
     &     ,status='old')
*
* settings
      open(12,file=sidir(1:iodl)//'settings_'//sgas(im)(1:j1)//'.dat'
     &     ,status='old')
*
* microwindows + occupation matrix
      open(13,file=sidir(1:iodl)//'mwoccmat_'//sgas(im)(1:j1)//'.dat'
     &     ,status='old')
*
* P profile
```

```
      open(16,file=sidir(1:iodl)//'in_pres.dat',status='old')
*
* T profile
      open(17,file=sidir(1:iodl)//'in_temp.dat',status='old')
*
* initial continuum profiles
      open(18,file=sidir(1:iodl)//'in_cont_'//sgas(im)(1:j1)//'.dat'
     &     ,status='old')
*
* initial VMR profiles
      open(19,file=sidir(1:iodl)//'in_vmr_prof.dat',status='old')
*
* altitudes of the profiles
      open(20,file=sidir(1:iodl)//'in_alt.dat',status='old')
*
* spectroscopic data
      open(21,file=sidir(1:iodl)//'spect_db_'//sgas(im)(1:j1)//'.dat'
     &     ,status='old')
*  Apodisation function in interferogram domain
       open(31,file=sidir(1:iodl)//'apod.dat',status='old')
*
* output results
      open(29,file=sodir(1:iodl)//sgas(im)(1:j1)//'_out.dat'
     &    ,status='unknown')
*
* output annotations
      open(30,file=sodir(1:iodl)//sgas(im)(1:j1)//'_annot.dat'
     &     ,status='unknown')
*
*
      call r_observ_vmr
      call r_settings_vmr(ilimb)
      if (im.ne.imaingas) then
       write(*,*)'The HITRAN codes of the main gas in settings_.dat'
       write(*,*)'and in call statement are not identical. STOP.'
       stop
      end if
      call r_mwoccmat_vmr(nselmw,ilimb)
*
* next call is activated only if lookup tables are used for cross sections
* in that case delete "c" in column 1
*
      call r_inalt_vmr
      call r_inpres_vmr(ipro)
      call r_intemp_vmr(ipro)
      call r_incont_vmr(nselmw,ipro)
      call uplimit_vmr(rzprof,rztang,rbase,rulatm)
      call r_spect_vmr(nselmw,iline,rulatm)
      call wmol_vmr(rwmol)
      call inigas_vmr(nselmw,iline,icode,imaingas,
     &           igas,igashi,igasmw,igasnr)
      call r_invmr_vmr(ipro,igas,igashi)
      call r_lookup_vmr(nselmw,igasmw,igashi,lookupc,
     &                  ilookupmw,ngas,ih,lmgas)
      call r_apod_vmr

*
      return
      end
*
```

```
      subroutine r_observ_vmr
* subroutine to read file of observations
      implicit none
      include 'parameters_vmr.inc'
      integer*4 j,k,l,nf,ilimb,nailsdp,nselmw
      integer*4 nsam(imxmw),nspikes(imxgeo)
      real*8 rzerof,rlat,rearad,rails(imxilc,imxmw),rnoise(imxmw,imxgeo)
     &      ,robs(imxi,imxgeo,imxmw),rztang(imxgeo)
      real*8 dlatp(imxgeo),dlongp(imxgeo),daltp(imxgeo),dazima(imxgeo),
     &      deleva(imxgeo),dlatt(imxgeo),dlongt(imxgeo),
     &      dlocer(imxgeo)
      character squal(imxgeo)*1,slab(imxmw)*25
      common/observ_v/ ilimb,nselmw,nailsdp,nsam,nspikes,rzerof,rails,
     &            rnoise,robs,rlat,rearad,dlatp,dlongp,daltp,dazima,
     &            deleva,dlatt,dlongt,rztang,dlocer
      common/chobserv_v/ squal,slab
      nf=11
      call skip_vmr(nf)
      read(nf,100) ilimb
* ilimb = number of sweeps in to the scan which is going to be processed (i5)
      call blind_vmr(ilimb,'imxgeo','geometries in r_observ')
      call skip_vmr(nf)
      read(nf,100) nselmw
* nselmw = total number of MW's selected for this retrieval
      call blind_vmr(nselmw,'imxmw','MW,s in r_observ')
      call skip_vmr(nf)
      read(nf,110) rzerof

      write(*,*)'rzerof=', rzerof

      if(rzerof .le. 0.1) stop 'rzerof has a wrong value'

* rzerof = zero filling expressed as the ratio between measured and
* transformed interferogram (f10.5)
      call skip_vmr(nf)
      read(nf,100) nailsdp
* nailsdp = number of AILS data points (i5)
      call blind_vmr(nailsdp,'imxilc','points for rails in r_observ')
      do 10 k=1,nselmw
        call skip_vmr(nf)
        read(nf,'(a25)') slab(k)
* slab(k) = label for MW k (a25)
        call skip_vmr(nf)
        read(nf,100) nsam(k)
* nsam(k) = number of spectral data points for MW k (i5)
        call blind_vmr(nsam(k),'imxi','observed points in r_observ')
        call skip_vmr(nf)
        read(nf,120)(rails(l,k),l=1,nailsdp)
   10 continue
* rails = apodized instrument line shape for all selected MW's (8g10.4)
      do 20 j=1,ilimb
       call skip_vmr(nf)
       read(nf,'(a1)') squal(j)
* squal(j) = quality indicator of sweep j (a1)
       call skip_vmr(nf)
       read(nf,100) nspikes(j)
* nspikes(j) = number of detected spikes in sweep j (i5)
       call skip_vmr(nf)
       read(nf,110) dlatp(j),dlongp(j),daltp(j)
* dlatp(j), dlongp(j), daltp(j) = latitude, longitude, altitude defining the
```

```
*                       geolocation of platform at sweep j (3f10.5)
      call skip_vmr(nf)
      read(nf,110) dazima(j),deleva(j)
* dazima(j), deleva(j) = azimuth and elevation angles of sweep j (2f10.5)
      call skip_vmr(nf)
      read(nf,110) dlatt(j),dlongt(j),rztang(j)
* dlatt(j), dlongt(j), rztang(j) = latitude, longitude, altitude defining
*                       geolocation of tangent point of sweep j (3f10.5)
      call skip_vmr(nf)
      read(nf,110) dlocer(j)
* dlocer(j) = local earth's radius curvature for sweep j (f10.5)
      do 30 k=1,nselmw
        call skip_vmr(nf)
        read(nf,120) rnoise(k,j)
* rnoise(k,j) = noise for MW k in sweep j (g10.4)

        if(rzerof.le.0.99) then
         rnoise(k,j)=rnoise(k,j)/rzerof
         write(*,*)'rzerof < 1: the NESR has been corrected '
        end if

        call skip_vmr(nf)
        read(nf,120) (robs(l,j,k),l=1,nsam(k))
* robs = observed spectral data points for MW k of sweep j (8g10.4)
   30  continue
   20 continue
      rlat=0.0
      rearad=0.0
      do 40 j=1,ilimb
       rlat=rlat+dlatt(j)
       rearad=rearad+dlocer(j)
40     continue
      rlat=rlat/float(ilimb)
      rearad=rearad/float(ilimb)
  100 format(i5)
  110 format(3f10.5)
  120 format(8g10.4)
      return
      end
*
      subroutine r_settings_vmr(ilimb)
* subroutine to read file of settings
      implicit none
      include 'parameters_vmr.inc'
      integer*4 nf,j,ilimb,iept,nswco2,ninterpol,imaingas,nrd,
     &          imxiterg,imxiterm,ifco
      real*8 rconvc(3),rmaxtv1,rmaxtv2,rzt12,rhwvar,rincz,redfact,
     &       rhw0ref,rexphref,rwmolref,rulatm,rbase,rsl
      real*8 delta,dstep,dsigm0,deps,drd,rucl,rzc0,rperc,
     &                rlambdain,rlambdadiv,rlambdamul

      character sdol*25,spare*150,sgas(imxhit)*6
      logical lookupc,lextinf1,lextinf2,lextinf3,
     &        lfit(imxlmb),lifptret,lifvmret,
     &        lirrgrid,lirrgridmw(imxmw)
      common/settings_v/ iept,nswco2,ninterpol,imaingas,nrd,rconvc,
     &                imxiterg,imxiterm,rmaxtv1,
     &                rmaxtv2, rzt12,rhwvar,rincz,redfact,rhw0ref,
     &                rexphref,rwmolref,rulatm,rbase,rsl,delta,
     &                dstep,dsigm0,deps,ifco,rucl,rzc0,rperc,
```

```
     &              rlambdain,rlambdadiv,rlambdamul

      common/chsettings_v/ sdol,spare,sgas
      common/lsettings_v/ lookupc,lextinf1,lextinf2,lextinf3,
     &                lfit,lifptret,lifvmret
      common/irrgrid_v/ lirrgridmw,lirrgrid
      nf=12

      call skip_vmr(nf)
* lifptret = switch for the use of p,T retrieved data
      read(nf,*)lifptret
      call skip_vmr(nf)
* lifvmret = switch for the use of last VMR retrieved data
      read(nf,*)lifvmret
      call skip_vmr(nf)
      read(nf,100) delta
* delta = frequency step that simulates infinitesimal spectral resolution
*         in forward simulations (f10.5)
      call skip_vmr(nf)
      read(nf,100) dstep
* dstep = frequency spacing between observed spectral data points (f10.5)
      call skip_vmr(nf)
      read(nf,'(a25)') sdol
* sdol = definition of the output level (a25)
      call skip_vmr(nf)
      read(nf,100)(rconvc(j),j=1,3)
* rconvc = vector of convergence criteria to stop iterations (3f10.5)
* rconvc(1) = variation of chi**2 between two consecutive iterations
* rconvc(2) = variation between calculated chi**2 and linear chi**2
* rconvc(3) = largest variation of parameters between consecutive iterations
      call skip_vmr(nf)
      read(nf,120) imxiterg
* imxiterg = maximum number of Gauss-Newton macro-iterations
      call skip_vmr(nf)
      read(nf,120) imxiterm
* imxiterm = maximum number of Marquard micro-iterations
      call skip_vmr(nf)
      read(nf,110) lookupc
* lookupc = switch for the use of cross sections look-up tables (l1)
      call skip_vmr(nf)
      read(nf,'(3l2)') lextinf1,lextinf2,lextinf3
* lextinf = switches for the combination of results with external information (l1)
      call skip_vmr(nf)
      read(nf,100) rmaxtv1,rmaxtv2,rzt12
* parameters_vmr.incor the layering of the atmosphere (3f10.5)
* rmaxtv1 = max. allowed T variation (K) between levels: 0 < altitude < rzt12
* rmaxtv2 = max. allowed T variation (K) between levels: rzt12<altitude<rulatm
* rzt12 = altitude (km) where the thresholds rmaxtv1 and rmaxtv2 are exchanged
      call skip_vmr(nf)
      read(nf,100) rhwvar
* rhwvar = max half-width change allowed between levels for ref. transiton (f10.5)
      call skip_vmr(nf)
      read(nf,100) rincz
* rincz = max thickness of the layers used for modelling the atmosphere (f10.5)
      call skip_vmr(nf)
      read(nf,100) redfact
* redfact = reduction factor for rincz to produce thinner layers
      call skip_vmr(nf)
      read(nf,100) dsigm0,rhw0ref,rexphref,rwmolref
* data relative to the transition that is used for layering criterion (4f10.5)
```

```
* dsigm0 = frequency position
* rhw0ref = half-width at the reference T and P
* rexphref = T dependence of half-width
* rwmolref = molecular weight
      call skip_vmr(nf)
      read(nf,100) rulatm
* rulatm = boundary of the atmosphere (km) (f10.5)
      call skip_vmr(nf)
      read(nf,100) deps
* deps = convergence criterion for the C.G. integrals (f10.5)
      call skip_vmr(nf)
      read(nf,120) iept
* iept = actual number of extra paths (i5)
      call skip_vmr(nf)
      read(nf,120) nswco2
* nswco2 = switch for the calculation of the CO2 chi factor (i5)
*        =0 no factor, =1 n2/o2 broadening, =2 n2 broadening only
* NOTE this switch is no longer active but is maintained in the case of future
*      applications
      call skip_vmr(nf)
      read(nf,120) ninterpol
*   ninterpol = switch for the interpolation on the cross-sections (i5)
*             = -1: no interpolation, all cross-sections recalculated
*             =  0: all those above the lowest geometry are interpolated
*             =  1: new calculation only of tangent-layer
*             =  2: new calculation for tangent-layer and the one above
      call skip_vmr(nf)
      read(nf,100) rbase,rsl
* parameters_vmr.incor the FOV convolution (2f10.5)
* rbase = greater base of the trapetium FOV-function (km)
* rsl   = half difference between bases of the trapetium FOV-function (km)
      call skip_vmr(nf)
      read(nf,120)imaingas
* imaingas = HITRAN code of the main molecule of the retrieval (i5)
      call skip_vmr(nf)
      read(nf,110)(lfit(j),j=1,ilimb)
* lfit(j) = logical vector that identifies the tangent altitudes which
* correspond to fitted parameters (40l2)

* Switch for fitting atmospheric continuum and offset (ifco):
*      ifco = 2 -->    p,T, continuum and offset are fitted
*      ifco = 1 -->    p,T and continuum are fitted
*      ifco = 0 -->    only p,T are fitted
      call skip_vmr(nf)
      read(nf,*) ifco
* Upper continuum limit (rucl): atmospheric continuum is not fitted at the
* sweeps having tangent altitude > rucl. Units: km
      call skip_vmr(nf)
      read(nf,*) rucl
* Altitude above which the continuum is set = 0
      call skip_vmr(nf)
      read(nf,*) rzc0
* Max.relative distance between MWs (with respect to the umbrella radius)
* so that the continuum is considered equal in the two MWs.
      call skip_vmr(nf)
      read(nf,*) rperc
* Parameters controlling the evolution of Marquardt damping factor:
*     rlambdain = initial value of rlambda
*     rlambdadiv = factor used to decrease rlambda at each Gauss iteration
*     rlambdamul = factor useed to increase rlambda at each Marquardt iteration
```

```
      call skip_vmr(nf)
      read(nf,*) rlambdain, rlambdadiv, rlambdamul
      call skip_vmr(nf)
      read(nf,*) lirrgrid
      write(*,*)lirrgrid
      call skip_vmr(nf)
      read(nf,'(a150)') spare
* spare = spare field for retrieval configuration
      drd=dstep/delta
      nrd=nint(drd)
      if(abs(drd-float(nrd)).gt.0.00001) then
       write(*,'(f12.6)')'ratio dstep/deltas not integer program stopped
     & nrd=',dstep/delta
       stop
      end if
  100 format(4f10.5)
  110 format(40l2)
  120 format(i5)
  130   format(40l2)
      return
      end
*
      subroutine r_mwoccmat_vmr(nselmw,ilimb)
* subroutine to read file of microwindows and occupation matrix
      implicit none
      include 'parameters_vmr.inc'
      integer*4 nf,j,k,nselmw,iordmw,ilimb,ifspmw(imxmw),nsam,
     &          iline(imxmw)
      integer*4 ngas(imxmw),ih(imxgmw,imxmw),
     &          ilookupmw(imxmw),i
      real*8 rmin,rmax,rconint(imxlmb,imxmw)
      character slab*10
      character smw(imxmw)*6,cc*2
      logical lokku(imxgeo,imxmw),lmgas(imxgmw,imxmw),
     &        lirrgridmw(imxmw),lirrgrid
      common/mwoccmat_v/ ifspmw,iline,rconint
      common/mwlookup_v/ ilookupmw,smw,ngas,ih,lmgas
      common/lmwoccmat_v/ lokku
      common/irrgrid_v/ lirrgridmw,lirrgrid
      nf=13
      call skip_vmr(nf)
      do 10 j=1,nselmw
        read(nf,130) iordmw,slab,ifspmw(j),nsam,rmin,rmax,
     &               iline(j),ilookupmw(j),ngas(j),
     &               lirrgridmw(j)
        if(ngas(j).ne.0) then
         write(cc,'(i2)')ngas(j)
         read(nf,('//cc//'i3)') (ih(i,j),i=1,ngas(j))
        end if
       smw(j)=slab(5:10)

      call blind_vmr(iline(j),'imxlin','transitions in r_mwoccmat')
   10   continue
* data for the MW's definition (i5,a10,2i10,2f10.5,i5)
* ifspmw(j) = index of the first sampling point of MW j
* NOTE: the sampling point at frequency=0 has index=1
* iline(j) = number of transitions in the spectroscopic database for MW j
*
* iordmw,slab,nsam,rmin,rmax = are scratch fields reporting:
* iordmw = progressive number
```

```
* slab = label of the MW as read in the observ_h2o.dat file
* nsam   = number of sampling points for the MW
* rmin,rmax = minimum and maximum altitude coverage of the MW
      do 20 j=1,nselmw
        call skip_vmr(nf)
        read(nf,110)(rconint(k,j),k=1,ilimb)
20    continue
* rconint(k,j) = spectral interval, around MW j, in which atmospheric continuum
* can be assumed to vary linearly at geometry k (8f10.5)
      call skip_vmr(nf)
      do 30 j=1,ilimb
      read(nf,120)(lokku(j,k),k=1,nselmw)
C       write(*,120)(lokku(j,k),k=1,nselmw)
30    continue
* lokku(j,k) = ocupation matrix element for geometry j and MW k (40l2)
100   format(2i5)
110   format(8f10.5)
120   format(40l2)
130   format(i5,a10,2i10,2f10.5,i5,2i2,l2)
      return
      end
*
* next subroutine is activated only if lookup tables are used for cross sections
* in that case delete all "c" in column 1
      subroutine r_lookup_vmr(nselmw,igasmw,igashi,lookupc,
     &                ilookupmw,ngas,ih,lmgas)
* subroutine for defining the variables that are used in cross for using
* or not using look-up tables
      implicit none
      include 'parameters_vmr.inc'
      integer*4 j,i,nselmw,mgas,igasmw(imxmw),
     &          ilookupmw(imxmw),ihit,igashi(imxgas),
     &          igasnr(imxgas,imxmw),ih(imxgmw,imxmw),
     &          ngas(imxmw)
      logical lmgas(imxgmw,imxmw),lookupc

       do 10 j=1,nselmw

       do 15 mgas=1,igasmw(j)
         lmgas(mgas,j)=.false.
15     continue

       if(lookupc) then
        if(ilookupmw(j).eq.1) then
          do 20 mgas=1,igasmw(j)
           ihit=igashi(igasnr(mgas,j))
            do 25 i=1,ngas(j)
              if (ihit .eq. ih(i,j)) goto 20
25          continue
           ilookupmw(j)=2
           lmgas(mgas,j)= .true.
20         continue
        end if
       else
        ilookupmw(j)=0
       end if
10    continue

      return
      end
```

```
*
      subroutine r_inalt_vmr
* subroutine to read file of the altitudes of initial profiles
      implicit none
      include 'parameters_vmr.inc'
      integer*4 nf,j,ipro
      real*8 rzprof(imxpro)
      character sident*25
      common/inalt_v/ ipro,rzprof
      nf=20
      call skip_vmr(nf)
      read(nf,'(a25)') sident
* sident = alphanumeric identifier of the file (a25)
      call skip_vmr(nf)
      read(nf,100) ipro
* ipro = number of elements of the vector of altitudes (i5)
      call blind_vmr(ipro,'imxpro','altitudes in r_inalt')
      call skip_vmr(nf)
      read(nf,110)(rzprof(j),j=1,ipro)
* rzprof(j) = altitude to which element j of T, P and VMR profiles are
*             referred to (8f10.5)
  100 format(i5)
110   format(8f10.5)
      return
      end
*
      subroutine r_inpres_vmr(ipro)
* subroutine to read file of initial P profile
      implicit none
      include 'parameters_vmr.inc'
      integer*4 nf,j,ipro
      real*8 rpprof(imxpro)
      character sident*25
      common/inpres_v/ rpprof
      nf=16
      call skip_vmr(nf)
      read(nf,'(a25)') sident
* sident = alphanumeric identifier of the file (a25)
      call skip_vmr(nf)
      read(nf,*)(rpprof(j),j=1,ipro)
* rpprof(j) = value of P at altitude j of the initial profile (8f10.5)
      return
      end
*
      subroutine r_intemp_vmr(ipro)
* subroutine to read file of initial T profile
      implicit none
      include 'parameters_vmr.inc'
      integer*4 nf,j,ipro
      real*8 rtprof(imxpro)
      character sident*25
      common/intemp_v/ rtprof
      nf=17
      call skip_vmr(nf)
      read(nf,'(a25)') sident
* sident = alphanumeric identifier of the file (a25)
      call skip_vmr(nf)
      read(nf,100)(rtprof(j),j=1,ipro)
* rpprof(j) = value of T at altitude j of the initial profile (8f10.5)
 100   format(8f10.5)
```

```
      return
      end
*
      subroutine r_incont_vmr(nselmw,ipro)
* subroutine to read the file of initial continuum profiles
      implicit none
      include 'parameters_vmr.inc'
      integer*4 nf, nselmw,j,k,ipro
      real*8 rcprof(imxpro,imxmw)
      character*25 smwlabel
      common /incont_v/ rcprof
      nf=18
      call skip_vmr(nf)
      do 10 j=1,nselmw
        read (nf,'(a25)') smwlabel
* smwlabel = scratch label of the microwindow (a25)
        read(nf,110)(rcprof(k,j),k=1,ipro)
* rcprof(k,j) = continuum cross section for MW j at altitude k (8e10.3)
   10 continue
  110 format(8e10.3)
      return
      end
*
      subroutine r_spect_vmr(nselmw,iline,rulatm)
* subroutine to read file of spectroscopic data
      implicit none
      include 'parameters_vmr.inc'
      integer*4 nf,j,k,nselmw,iline(imxmw)
      integer*4 ioutin(imxlin,imxmw),iiso(imxlin,imxmw),
     &          icode(imxlin,imxmw)
      real*8 rint0(imxlin,imxmw),rhw0(imxlin,imxmw),relow(imxlin,imxmw),
     &    rexph(imxlin,imxmw),ruplin(imxlin,imxmw),rlolin(imxlin,imxmw)
      real*8 dsilin(imxlin,imxmw),rulatm
      character smwlabel*25
      common/spect_v/ dsilin,rint0,rhw0,relow,rexph,ruplin,rlolin,
     &             ioutin,iiso,icode
      nf=21
      do 10 j=1,nselmw
       call skip_vmr(nf)
       read(nf,'(a25)') smwlabel
* smwlabel = scratch label of the microwindow (a25)
       do 20 k=1,iline(j)
        read(nf,100) dsilin(k,j),rint0(k,j),rhw0(k,j),relow(k,j),
     &              rexph(k,j),ioutin(k,j),iiso(k,j),icode(k,j),
     &              ruplin(k,j),rlolin(k,j)


         if (rlolin(k,j).le.8.)then
          rlolin(k,j)=0.
         else
          rlolin(k,j)=rlolin(k,j)-.5
         end if

        if(abs(ruplin(k,j)-70.) .le. 1.e-4)then
          ruplin(k,j)=rulatm
        else
          ruplin(k,j)=ruplin(k,j)+10.
        end if

* spectroscopic data for transition k of MW j
```

```
* (f11.5,2e11.3,2f11.4,2x,i1,2x,i2,1x,i2,2f7.2)
*
* dsilin = line position
* rint0  = line strength
* rhw0   = pressure broadening
* relow  = energy of starting level
* rexph  = exponent for T dependence of rhw0
* ioutin = control on continuum treatment
* iiso   = isotope code
* icode  = hitran code of the molecule
* ruplin = highest altitude where the line has to be used
* rlolin = lowest altitude where the line has to be used
*
20    continue
10    continue
100   format(f12.6,(1pe11.3),0pe11.3,2f11.4,2x,i1,2x,i2,1x,i2,2f7.2)
      return
      end
*
      subroutine r_invmr_vmr(ipro,igas,igashi)
* subroutine to read file of initial VMR profiles
      implicit none
      include 'parameters_vmr.inc'
      integer*4 nf,j,k,l,igc,ipro,igas,igashi(imxgas)
      real*8 rvmrprof(imxpro,imxgas)
      character sident*25,sname(imxgas)*25
      common/invmr_v/ rvmrprof
      common/chinvmr_v/ sname
      nf=19
      do 10 j=1,igas
        rewind nf
        call skip_vmr(nf)
        read(nf,'(a25)') sident
        call skip_vmr(nf)
        do 10 k=1,igashi(j)
         read(nf,100) igc,sname(j)
* igc   = progressive number denoting hitran gas code
* sname = name of the gas
         read(nf,110)(rvmrprof(l,j),l=1,ipro)
   10 continue
100   format(i5,1x,a25)
  110 format(8e10.3)
      return
      end

      subroutine skip_vmr(nf)
* subroutine to skip comment lines on read files
* convention is that: at least one comment line appears before a read statement
*                     last comment line starts with a character # in column 1
*
      implicit none
      character*1 sendc
      integer*4 nf
      sendc=' '
      do while (sendc.ne.'#')
        read(nf,'(a1)') sendc
      end do
      return
      end
*******************************************************************************
```

```
*******************************************************************************
*  SUBROUTINE              : uplimit
*  CREATED BY              : michael hoepfner
*  DATE OF CREATION        : 5.6.96
*  DATE OF LAST MODIFICATION  :
*  MODIFIED BY             :
*  LAST MODIFICATION BY    :
*
*  DESCRIPTION             :
*  Determination of the upper atmospheric limit to be calculated.
*  Changes the input-variable 'rulatm' if the highest altitude to be
*  considered from the line-database is lower than the input 'rulatm'
*
*  INPUTS:
*   nselmw         total number of selected microwindows for the retrieval
*   iline(imxmw)    number of lines in each microwindow
*   ruplin(imxlin,imxmw) upper limit where the line has to be considered
*   rzprof(imxpro)  vector of altitudes Z to which rtprof, rpprof
*                and rvmrprof are referred [km]
*   rztang(imxgeo)  vector containing the engineering values of
*                tangent altitudes
*   rbase          greater base of trapezium of Field of View function
*   rulatm          max. upper limit of the atmosphere (km) from settings
*
*  OUTPUTS:
*   rulatm          upper limit of the atmosphere (km) (for radiative
*                transfer calculations)
*
*******************************************************************************
*******************************************************************************
      subroutine uplimit_vmr(rzprof,rztang,rbase,rulatm)
      implicit none
      include 'parameters_vmr.inc'
      real*8 rzprof(imxpro),rztang(imxgeo),
     &       rbase,rulatm
*
* if the new rulatm is less than the highest observation geometry
*  (+2*rbase) it is changed
*
      if (rulatm.lt.rztang(1)+2.*rbase) rulatm=rztang(1)+2.*rbase
*
* if the new rulatm is higher than the highest point of the inital profiles
*  it is changed and a warning is written.
*
      if (rulatm.gt.rzprof(1)) then
        rulatm=rzprof(1)
        write(*,*) 'WARNING: The upper limit of the atmosphere'
        write(*,*) ' (rulatm) was higher than the highest altitude of'
        write(*,*) ' the initial profiles. This was corrected, but'
        write(*,*) ' check rulatm in SETTINGS.DAT! '
      end if
*
* test output
*
      write(*,*) '*************************************************'
      write(*,*) 'Test output from uplimit:'
      write(*,*) 'The upper limit for the radiative transfer'
     &          //' calculations (rulatm) =',rulatm
      write(*,*) '*************************************************'
      end
```

```
*******************************************************************************
*  SUBROUTINE              : inigas
*  CREATED BY              : michael hoepfner
*  DATE OF CREATION        : 21.5.96
*  DATE OF LAST MODIFICATION  :
*  MODIFIED BY             :
*  LAST MODIFICATION BY    :
*
*  DESCRIPTION             :
*  Initialisation of the variables 'igas, igashi,igasmw,igasnr' that
*  define the two internal numberings of the gases
*
*
*  INPUTS:
*   nselmw         total number of selected microwindows
*   iline(imxmw)    number of lines in each microwindow
*   icode(imxlin,imxmw)   HITRAN molecular code for each line of each MW
*   imaingas           HITRAN code of the main gas of the retrieval
*
*  OUTPUTS:
*   igas          number of different gases for actual retrieval
*   igashi(imxgas)  HITRAN code nr for each gas-number of actual retrieval
*   igasmw(imxmw)   number of gases to be considered for each mw.
*   igasnr(imxgas,imxmw)  retrieval gas number for each internal gas
*                    number of each NW
*
*******************************************************************************
      subroutine inigas_vmr(nselmw,iline,icode,imaingas,
     &                igas,igashi,igasmw,igasnr)
      implicit none
      include 'parameters_vmr.inc'
      integer*4 nselmw,iline(imxmw),icode(imxlin,imxmw),
     &         igas,igashi(imxgas),igasmw(imxmw),
     &         igasnr(imxgas,imxmw),imaingas,
     &         jmw,kline,mgas,m1,j
      logical lmaingas
*
* initialization, so that the main gas has the retrieval numbering =1
*
      igas=1
      igashi(1)=imaingas
*
*  begin loop over microwindows
*
      do 10 jmw=1,nselmw
*
* initialization, so that the main gas has the microwindow numbering =1
*
        igasmw(jmw)=1
        igasnr(1,jmw)=1
*
* initialization of the test variable, if the main gas has a line in the MW
*
        lmaingas=.false.
*
*  begin loop over the lines of one mircowindow
*
        do 20 kline=1,iline(jmw)
*
*  test, it the line is a line of the main gas
```

```
*
      if (icode(kline,jmw).eq.imaingas) lmaingas=.true.
*
*  the retrieval gas numbering is performed
*
      do 30 mgas=1,igas
        if (igashi(mgas).eq.icode(kline,jmw)) then
          m1=mgas
          goto 40
        end if
30      continue
      igas=igas+1
      call blind_vmr(igas,'imxgas','Blind stop in inigas')
      igashi(igas)=icode(kline,jmw)
      m1=igas
*
* the internal microwindow gas numbering is performed
*
40      continue
      do 50 mgas=1,igasmw(jmw)
        if (igashi(igasnr(mgas,jmw)).eq.icode(kline,jmw))
     &      goto 20
50      continue
      igasmw(jmw)=igasmw(jmw)+1
      igasnr(igasmw(jmw),jmw)=m1
*
*  end loop over the lines of one mircowindow
*
20    continue
*
*  test, if the main gas is at least one line of the mw
*
      if (.not.lmaingas) then
        print*,'No line of the main gas in microwindow',jmw
        stop
      end if
*
*  test the number of gases per Mw
*
      call blind_vmr(igasmw(jmw),'imxgmw','Blind stop in inigas')
*  end loop over microwindows
*
10   continue
*
* Test -output
*
    write(*,*) '*****************************************'
    write(*,*) 'Test output of inigas:'
    write(*,*)
    write(*,*) 'Total number of gases:',igas
    write(*,'(a,32i3)') ' HITRAN codes:',(igashi(j),j=1,igas)
    do jmw=1,nselmw
    write(*,*)
    write(*,'(a,i4)') ' Microwindow:',jmw
    write(*,'(a,32i3)') ' Local mw gas number:',(j,j=1,igasmw(jmw))
    write(*,'(a,32i3)') ' Global gas number  :',
   &              (igasnr(j,jmw),j=1,igasmw(jmw))
    write(*,'(a,32i3)') ' HITRAN gas number  :',
   &              (igashi(igasnr(j,jmw)),j=1,igasmw(jmw))
    end do
```

```
      write(*,*)
      write(*,*) '*****************************************'
      end
*******************************************************************************
*******************************************************************************
      subroutine wmol_vmr(x)
      implicit none
      include 'parameters_vmr.inc'
      real*8 rwmol(imxhit,imxism),x(imxhit,imxism)
      integer*4 j,k
      data
     &(rwmol( 1,j),j=1,4) / 18., 20., 19., 19./
      data
     &(rwmol( 2,j),j=1,8) / 44., 45., 46., 45., 47., 46.,48.,47./
      data
     &(rwmol( 3,j),j=1,5) / 48., 50., 50., 49., 49./
      data
     &(rwmol( 4,j),j=1,5) / 44., 45., 45., 46., 45./
      data
     &(rwmol( 5,j),j=1,6) / 28., 29., 30., 29., 31., 30./
      data
     &(rwmol( 6,j),j=1,3) / 16., 17., 17./
      data
     &(rwmol( 7,j),j=1,3) / 32., 34., 33./
      data
     &(rwmol( 8,j),j=1,3) / 30., 31., 32./
      data
     &(rwmol( 9,j),j=1,2) / 64., 66./
      data
     &(rwmol(10,j),j=1,1) / 46./
      data
     &(rwmol(11,j),j=1,2) / 17., 18./
      data
     &(rwmol(12,j),j=1,1) / 63./
      data
     &(rwmol(13,j),j=1,3) / 17., 19., 18./
      data
     &(rwmol(14,j),j=1,1) / 20./
      data
     &(rwmol(15,j),j=1,2) / 36., 38./
      data
     &(rwmol(16,j),j=1,2) / 80., 82./
      data
     &(rwmol(17,j),j=1,1) /128./
      data
     &(rwmol(18,j),j=1,2) / 51., 53./
      data
     &(rwmol(19,j),j=1,4) / 60., 62., 61., 62./
      data
     &(rwmol(20,j),j=1,3) / 30., 31., 32./
      data
     &(rwmol(21,j),j=1,2) / 52., 54./
      data
     &(rwmol(22,j),j=1,1) / 28./
      data
     &(rwmol(23,j),j=1,3) / 27., 28., 28./
      data
     &(rwmol(24,j),j=1,2) / 50., 52./
      data
     &(rwmol(25,j),j=1,1) / 34./
```

```
      data
     &(rwmol(26,j),j=1,2) / 26., 27./
      data
     &(rwmol(27,j),j=1,1) / 30./
      data
     &(rwmol(28,j),j=1,1) / 28./
      data
     &(rwmol(29,j),j=1,1) / 66./
      data
     &(rwmol(30,j),j=1,1) / 146./
      data
     &(rwmol(31,j),j=1,3) / 34.,36.,35./
      data
     &(rwmol(32,j),j=1,1) / 46./
      data
     &(rwmol(33,j),j=1,1) / 33./
      data
     &(rwmol(34,j),j=1,1) / 16./
      data
     &(rwmol(35,j),j=1,2) / 97.,99./
      data
     &(rwmol(36,j),j=1,1) / 30./
      do 10 j=1,imxhit
        do 20 k=1,imxism
          x(j,k)=rwmol(j,k)
20      continue
10    continue
      end

      subroutine r_apod_vmr
* subroutine to read file of apodisation function in the interferogram
* domain
      implicit none
      include 'parameters_vmr.inc'
      integer*4 nf,j,napod,n
      real*4 rapod(imxapo)
      real*8 rx
      common/apod_f_v/napod,rapod
      nf=31
      call skip_vmr(nf)
      read(nf,*) napod

       call blind_vmr(napod,'imxapo','blind-stop in finput_vmr.f')

*   Control on the value of napod: IT HAS TO BE (2^n+1), with n=integer
      rx=(log(dble(napod-1)))/log(2.)
      write(*,*)'rx=',rx
      n=nint(rx)
      write(*,*)'n=',n
      if(abs(rx-dble(n)).gt.1.e-6) then
       write(*,*)'program stopped in finput_vmr.f'
       write(*,*)'(napod-1) has to be 2^n, with n integer'
       write(*,*)'(napod-1) =',(napod-1)
       stop
      end if

      read(nf,*) (rapod(j),j=1,napod)
      close(nf)

      return
```

```
end
```