Module, subroutine and variable listing and description

M. Höpfner

1 Module Listing and Description

- Modules for fixed data and parameters:
  - ckdcoe_m
    Parameters for the calculation of the H_2O-continuum.
    Subroutines:
    -/-
  - param_m
    Definition of mathematical-physical and dimensioning parameters.
    Subroutines:
    -/-
  - precis_m Definition of precision parameters for integer and real variables.
    Subroutines:
    -/-
- Module for type-definition:
  - types_m
    Definition of types for all derived-type variables in kopra.
    Subroutines:
    -/-
- Modules for variable-interfaces:
  - inpdat_m
    Definition of input variables.
    Subroutines:
    -/-
  - modat_m
    Definition of internal model variables.
    Subroutines:
    -/-
  - outdat_m
    Definition of output variables.
    Subroutines:
    -/-
- Modules with mathematical libraries:
  - linpac_m
    Subroutine collection for linear algebra.
    (Translated from the slatec library on http://www.netlib.org/slatec/)
    Subroutines:
    public :: balanc, cbal, balbak, cgedi, cgefa, cginv, corth, elmhes
    eltran, compqr2, hqr, cg, rg, qsort
  - recipe_m
    Subroutine collection from numerical recipes.
    Subroutines:
    public :: polcoe, hunt, locate, spline, splint, splint1, linint1, isort

- Modules for often used subroutines:
  - give_m
    Collection of subroutines which return the value of atmospheric parameters for a given location in the atmosphere (latitude, longitude, altitude). The interpolation is performed either in the input profiles or in the retrieval parameter profiles.
    Subroutines:
    public :: give_p,give_T,give_vmr,give_Tvib, give_aerabs,give_aersca
    private :: give_pgrad, give_Tgrad, give_Tvibgrad, geodistance
  - varsub_m
    Collection of subroutines that are often use by different modules.
    Subroutines:
    public :: delete_element_i, multiple_cut, go_next, g, press, alti, seconds, yinterpol1, yinterpol2
- Modules for specialized tasks:
  - abco_m
    Calculates absorption coefficients [cm^2] for all layers and observation geometries.
    Subroutines:
    public :: absco_calc
    private :: absco_branch, allocate_geo_mw, line_strength, cho2chi
  - addlin_m
    Provides subroutines for the efficient calculation of absorption coefficient on a line-by-line basis.
    The basic idea the use of look-up-tables for determining the optimal set of sampling points for each spectral line.
    Subroutines:
    public :: allocate_cutoff, deallocate_cutoff, allocate_grid, deallocate_grid, add_lines_chilm, add_lines_chi, add_lines_lm, add_lines, interpolate_grid
    private :: voigt
  - gascon_m
    Calculation of N_2,O_2, H_2O, CO_2 continua.
    Subroutines:
    public :: calc_h2ocon, calc_n2con, calc_o2con
  - ifnlte_m
    Interface to the nlte-model
    Subroutines:
    public :: interface_nlte
  - ilsfov_m
    Calculate ails and convolve fine-grid spectrum. Determine field-of-view weighted average spectrum.
    Subroutines:
    public :: envils, envfovils, ilsapo, fovils1, ilsapo_fixed, ilsfov_fixed
    private :: value_test, vertinterpol, fovmeanspec, scndderfov, fovilsspec, derive,
    convolution, makeifg, makeils, shiftwvnr, makecifg, fovtrm, makecils, addnoise

- inider_m
  Initialization of variable deri% for derivative calculation.
  Subroutines:
  public :: ini_deri
- inipar_m
  Initialization of the vector para% which contains the parameterization of atmospheric profiles.
  Subroutines:
  public :: ini_para
  private :: ini_para_vmr, ini_para_T, ini_para_Tvib,
  ini_para_aerabs, ini_para_Tgrad, ini_para_vmrgrad, ini_para_p
- inismw_m
  Determination of internal forward model sub-microwindows.
  Subroutines:
  public :: ini_sub_mw
- iniout_m
  Initialization and determination of the output variable outdat%.
  Subroutines:
  public :: ini_output
- input_m
  Initialization and definition of all input variables.
  Subroutines:
  public :: input
  private :: input_hitmol, input_main, input_mwdef, input_isoprof, input_pTprof,
  input_vmrprof,
  input_pTgradprof, input_vmrgradprof, input_contprof, input_Tvibprof,
  input_Tvibgradprof,input_ails,
  make_hydroequi, ils_radius, extend_mw, speciorder, isomult
- inspec_m
  Read spectroscopic data and built up speci% variable.
  Subroutines:
  public :: input_spectroscopy
  private :: numdata, readlines, isoabun, linein, allocopy1, allocno_nlte,
  allocno_lm, alloc_nlte,
  allocopy2, pointmw, check_mwspeci, read_lmdata, alloc_lm, delete_lmlines,
- linmix_m
  Provides subroutines for line-mixing.
  Subroutines:
  public :: y_calc_rk, y_calc_dd, corr_y_coeffs,
- miemod_m
  Performs the Mie-calculations.
  Subroutines:
  public :: mie
  private :: miecoe,mieampli,set_miepar,set_gauher
- modgeo_m
  Introduces additional geometries for the simulation of FOV and defines the new occupation matrix for all the simulated
  geometries ('simulated geometries' are called all geometries to be calculated in the forward model i.e. the

observed plus the additional geometries for FOV.) Subroutines:
public :: make_modelgeo
private :: addsim_a, addsim_b, addgeo, make_occusim,

- modlev_m
  Defines the model altitude levels for the forward calculation.
  Subroutines:
  public :: make_modelgrid
  private :: min_distance, calc_ztang, base, grid_t_hw, grid_tang, delete_level, add_level,
- nltder_m
  Post derivatives in case of use of nlte-model
  Subroutines:
  public :: nlte_postderi
  private :: nlte_vmr_postderi, nlte_x_postderi
- offsca_m
  Add offset and scale spectra. Calculate offset and scale derivatives.
  Subroutines:
  public :: offset_scale
  private :: sca, off, sca_deri, off_deri
- parchk_m
  Checks if variation of a distinct retrieval parameter influences the values of the atmospheric quantity
  inside an altitude region of the atmosphere.
  Subroutines:
  public :: para_aerabs_change, para_vmr_change, para_T_change, para_Tvib_change, para_Tgrad_change, para_vmrgrad_change
- radtra_m
  Calculation of the radiative transfer through the atmosphere and determination of the fine-grid spectra and their derivatives.
  Convolution of the fine-grid spectra with the ails to get the spectra on the coarse-grid (measurement grid) and
  calculation of the field-of-view weighting.
  Subroutines:
  public :: radtrans
  private :: radtrans_mw, tausrc, optsrcdTvib, derivmr_calc, deriaer_calc, deriT_calc, deriTvib_calc,
  deriTgrad_calc, derivmrgrad_calc, writeabco, writespec, lsimobs, finewrk, ilsapofov_calc, sub_mw_minmax,
  alloc_Sails
- rayctl_m
  Controls ray-tracing, calculation of path integrated values (Curtis-Godson values and column amounts) and the
  derivatives of path integrated values wrt retrieval parameters.
  Subroutines:
  public :: raytrace_ctrl
  private :: allocgeo1, homog_path, para_dcol_ne0, para_dabsopt_ne0, para_dT_ne0, para_dTgrad_ne0,
  para_dTvib_ne0, para_dcolgrad_ne0, calc_nlte_ratios, calc_nlte_dratios
- ray_m Calculate ray-tracing, path integrated values (Curtis-Godson values and column amounts) and the

derivatives of path integrated values wrt retrieval parameters.
Subroutines:
public :: raytra, tangalt
private :: leveltang, leveltrans, integrate, nmax_calc, index_nlte, tangpara, height, findtop, facds, cross,
unitvector, refin, epsi, gradient, observer, tnew, latlon, vektorsin, alloc_geo, pathcopy, ray_out

- transf_m Collection of subroutines specialized for radiative transfer calculation. (Interpolation between different grids.)
  Calculation of planck/source functions and N_2 and O_2 continua.
  Subroutines:
  n2calc, o2calc, h2ocalc, planck, alpha, dalphadTvib, source, dsourcedTvib, absco1, absco2, scal_add_fabsco,
  fabsco_add_scal, scal_add_absco, absco_add_scal, scal_sub_fabsco, fabsco_sub_scal, scal_sub_absco,
  absco_sub_scal, scal_addsub_absco, scal_mul_fabsco, fabsco_mul_scal, absco_mul_scal, scal_mul_absco,
  absco_mul_complex, complex_mul_absco, cut_fabsco, cut_absco, ipos, fine_absco, fine_fabsco, exp_fabsco,
  exp_absco, absco_to_absco, fabsco_to_absco, fabsco_sub_fabsco, absco_sub_fabsco, fabsco_sub_absco,
  absco_sub_absco, fabsco_add_fabsco, absco_add_fabsco, fabsco_add_absco, absco_add_absco, absco_addsub_absco,
  fabsco_mul_fabsco, absco_mul_fabsco, fabsco_mul_absco, absco_mul_absco, Re_absco, Re_fabsco, Im_absco, Im_fabsco,
  deallocate0, deallocate1, deallocate2, deallocate3, init0_absco, init1_absco, init2_absco, init3_absco

- wriout_m
  Write output spectrum outdat%... to file.
  Subroutines:
  public :: writeout
  private :: writespec

- xinput_m
  Reads measured heavy molecule cross-section data and determines measuring range for every microwindow/xsct.-gas.
  Subroutines:
  public :: input_xsection
  private :: deallocate_range, make_x, readxfilenames, readxbody, readxheader, check_mwXspeci

- xintpl_m
  Pressure, temperature and wavenumber grid interpolation of heavy molecule cross-section measurements.
  Subroutines:
  public :: allocate_x, deallocate_x, interpl_xpt
  private :: interpl_pt1, interpl_pt2, interpl_pt3, interpl_pt4, interpl_x, quadrant, quad_occ, distance1, distance2

- Modules for control tasks:

  - kopfwd_m
    Controls forward model run (without in/output) and determines numerical p-derivatives.

Subroutines:
public :: kopra_forwrd
private :: kopra_derip

- kopra Main program.
  Subroutines:
  -/-

2 Subroutine Listing and Description

```
pri...private
pub...public

Subroutines from:______________________________________________________________
-------------------------------------------------------------------------------
abco_m.f90   (module abco_m)
addlin_m.f90 (module addlin_m)
give_m.f90   (module give_m)
ifnlte_m.f90 (module ifnlte_m)
ilsfov_m.f90 (module ilsfov_m)
inider_m.f90 (module inider_m)
iniout_m.f90 (module iniout_m)
inipar_m.f90 (module inipar_m)
inismw_m.f90 (module inismw_m)
input_m.f90  (module input_m)
inspec_m.f90 (module inspec_m)
kopfwd_m.f90 (module kopfwd_m)
linmix_m.f90 (module linmix_m)
miemod_m.f90 (module miemod_m)
modgeo_m.f90 (module modgeo_m)
modlev_m.f90 (module modlev_m)
nltder_m.f90 (module nltder_m)
offsca_m.f90 (module offsca_m)
parchk_m.f90 (module parchk_m)
radtra_m.f90 (module radtra_m)
ray_m.f90    (module ray_m)
rayctl_m.f90 (module rayctl_m)
varsub_m.f90 (module varsub_m)
wriout_m.f90 (module wriout_m)
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
absco_calc@abco_m            pub Calculates absorption coefficients [cm**2]
                                 for all layers and observation geometries

-------------------------------------------------------------------------------
absco_branch@abco_m          pri Calculates the absorption coefficient of
                                 branch

-------------------------------------------------------------------------------
accuracy_calc@abco_m         pri Calculates the accuracy for absortion cross
                                 section calculation

-------------------------------------------------------------------------------
addlev@modlev_m              pri  Add additional levels between two base-levels
                                  if temperature or half-width conditions are
                                  valid
```

```
-----------------------------------------------------------------------------
add_level@modlev_m          pri Add one altitude level to a given profile.
-----------------------------------------------------------------------------
add_lines@addlin_m          pub add the individual lines on the grid made up
                                of different tiers
-----------------------------------------------------------------------------
add_lines_chi@addlin_m      pub add the individual lines on the grid made up
                                of different tiers
-----------------------------------------------------------------------------
add_lines_chilm@addlin_m    pub add the individual lines on the grid made up
                                of different tiers
-----------------------------------------------------------------------------
add_lines_lm@addlin_m       pub add the individual lines on the grid made up
                                of different tiers
-----------------------------------------------------------------------------
addgeo@modgeo_m             pri adds one geometry to the list of simulated
                                geometries
-----------------------------------------------------------------------------
addnoise@ilsfov_m           pri Adds Gaussian noise to the coarse spectra
-----------------------------------------------------------------------------
addsim_a@modgeo_m           pri adds simulated geometries to the observed ones
                                for criterion accu%ifov <= 0 (criterion 1 in
                                input-file)
-----------------------------------------------------------------------------
addsim_b@modgeo_m           pri adds simulated geometries to the observed ones
                                for criterion accu%ifov > 0 (criterion 2 in
                                input-file)
-----------------------------------------------------------------------------
allocate_cutoff@addlin_m    pub allocates memory and reads cutoff-files
                                "cutdop.dat" and "cutlor.dat"
-----------------------------------------------------------------------------
allocate_geo_mw@abco_m      pri allocates geo%...%mw part of geo% variable
-----------------------------------------------------------------------------
allocate_grid@addlin_m      pub allocates and initializes the grid tiers
-----------------------------------------------------------------------------
alloc_geo@ray_m             pri allocate geo()%...%lay and geo()%...%lay%speci
                                geo%...%lay%speci%iso , geo%... %iso%state
-----------------------------------------------------------------------------
allocgeo1@rayctl_m          pri allocation of geo%() and geo%()par()%
-----------------------------------------------------------------------------
alloc_lm@inspec_m           pri allocate the line mixing branches of vector
                                speci and copy the line data of the line mxing
                                lines from the vector lm into
                                speci()%iso()%band()%branch()%line and
                                speci()%iso()%band()%branch()%lmline
-----------------------------------------------------------------------------
alloc_nlte@inspec_m         pri determine number of different isotopes for
                                each species where nlte has to be considered
                                and the number of nlte bands.
                                Allocate speci%iso and speci%iso%band
-----------------------------------------------------------------------------
alloc_Sails@radtra_m        pri allocate data vector where the spectrum and
                                derivatives are stored be careful: the
                                sub-microwindow index is on the internal!
```

```
                                 sub-microwidows
-------------------------------------------------------------------------------
allocno_lm@inspec_m          pri allocate speci%iso%band%branch in the case no
                                 line mixing is considered
-------------------------------------------------------------------------------
allocno_nlte@inspec_m        pri allocate speci%iso and speci%iso%band in the
                                  case no nlte is considered
-------------------------------------------------------------------------------
allocopy1@inspec_m           pri allocate speci()% vector and copy line data in
                                 case no nlte and no line mixing is considered
-------------------------------------------------------------------------------
allocopy2@inspec_m           pri allocate the vectors
                                 speci()%iso()%band()%branch(0)%line  and copy
                                 the line data from vector spe()%line into this
                                 vectors.
-------------------------------------------------------------------------------
alti@varsub_m                pri Calculation of the altitude using mean
                                 temperature of the layer and g at the lower
                                 boundary level
-------------------------------------------------------------------------------
altitest@input_m             pri Test if altitudes in profile file are equal to
                                 altitudes in kopra.inp
-------------------------------------------------------------------------------
base@modlev_m                pri determine index of highest atmospheric
                                 base_level: nbaselev so that the highest
                                 altitudelevel is lower or equal accu%upatm and
                                 make the base-levels
-------------------------------------------------------------------------------
brdaero_gridinv_calc@radtra_m pri determine for each gridpoint of the
                                 irregular fine grid the corresponding grid
                                 point in the broadband-aerosol grid
-------------------------------------------------------------------------------
brdaero_inter@radtra_m       pri interpolate the broadband-aerosol wavenumber
                                 dependence on the irregular fine grid on
                                 which the radiative transfer is performed
-------------------------------------------------------------------------------
calc_fq@input_m              pri calculate the fq-nlte-factors if sw%fq_equ1=
                                 .false.
-------------------------------------------------------------------------------
calc_new_abcos@abco_m        pri determines if the absorption cross sections
                                 for a layer of a geometry will be calculated
                                 new or if
                                 they will be taken from the lowest geometry
-------------------------------------------------------------------------------
calc_nlte_ratios@rayctl_m    pri the derivative of the ratios of population
                                 geo()%...%state()%r with respect to Tkin is
                                 calculated
-------------------------------------------------------------------------------
calc_ztang@modlev_m          pri determination of the tangent altitudes if the
                                 nadir angles of a limb scan  are given
                                 (sw%modeobs = 2 or 4)
-------------------------------------------------------------------------------
check_mwspeci@inspec_m       pri deletes species from the microwindow species
                                 list if there are no lines of the species in
```

```
                                the microwindow
-------------------------------------------------------------------------------
check_T_hw@modlev_m          pri  Check if conditions on temperature variation
                                  and half-width variation are fulfilled so
                                  that an additional level must be added

-------------------------------------------------------------------------------
convolution@ilsfov_m         pri convolution of spectrum with ILS and gridpoint
                                 density reduction

-------------------------------------------------------------------------------
corr_y_coefs@linmix_m        pri corrects the y-coefficients

-------------------------------------------------------------------------------
co2chi@abco_m                pri Calculates the chi-factor for the correction
                                 of the co2-lineshape. The chi-factor is
                                 calculated for the n2- and the o2-broadening
                                 of co2-lines using the parametrizations from:

                                 C. Cousin, R. Le Doucen, C. Boulet, and
                                 A. Henry,
                                 'Temperature dependence of the absorption in
                                 the region beyond the 4.3-um band head of CO2.
                                 2: N2 and O2 broadening', Appl. Opt., 24,
                                 3899-3907, 1985.

                                 V. Menoux, R. Le Doucen, J. Boissoles, and
                                 C. Boulet,
                                 'Line shape in the low frequency wing of
                                 self- and N2 broadened v3 CO2 lines:
                                 temperature dependence of the asymmetrie',
                                 Appl. Opt., 30, 281-286, 1991.

                                 The chi-factors are linearily interpolated in
                                 the ranges 193-238K and 238-296K and linearily
                                 extrapolated to lower (higher) temperatures
                                 from these ranges..
                                 The chi-factor is then calculated by weighting
                                 of the N2 and O2 factors according to their
                                 atmospheric abundance.
-------------------------------------------------------------------------------
cross@ray_m                  pri cross - product

-------------------------------------------------------------------------------
dangle_dztang@rayctl_m       pri calculates sim()%dangle_dztang: the derivative
                                 of the nadir angle wrt the tangent altitude

-------------------------------------------------------------------------------
deallocate_cutoff@addlin_m   pub deallocates space used for the cutoff-tables

-------------------------------------------------------------------------------
deallocate_grid@addlin_m     pub deallocates the grid tiers

-------------------------------------------------------------------------------
delete_element_i@varsub_m    pri delete integer element i from an integer
                                 vector

-------------------------------------------------------------------------------
delete_level@modlev_m        pri Delete one level from a given profile.

-------------------------------------------------------------------------------
delete_lmlines@inspec_m      pri Deletes lines from the %branch(0)%line lines
                                 if they are also included in the line mixing
```

```
                                branches, so that these lines are not
                                calculated twice.
-------------------------------------------------------------------------------
deriaer_calc@radtra_m       pri Calculation of derivatives with respect to
                                aerosol absorption parameters
-------------------------------------------------------------------------------
derinum_calc@radtra_m       pri Calculation of derivatives with respect to
                                broadband aerosol number density parameters
-------------------------------------------------------------------------------
deriMie_calc@radtra_m       pri Calculation of derivatives with respect to
                                Mie parameters
-------------------------------------------------------------------------------
derip_refr@ray_m            pri  Calculate derivatives of the cg-pressure
                                 of air and the air-column with respect
                                 to p-parameters
                                 (including refraction dependence)
-------------------------------------------------------------------------------
deriT_calc@radtra_m         pri Calculation of derivatives with respect to T
                                parameters
-------------------------------------------------------------------------------
deriTgrad_calc@radtra_m     pri Calculation of derivatives with respect to
                                Tgrad parameters
-------------------------------------------------------------------------------
deriTvib_calc@radtra_m      pri Calculation of derivatives with respect to vmr
                                parameters
-------------------------------------------------------------------------------
derive@ilsfov_m             pri derivative of spectrum with respect to
                                abszissa
-------------------------------------------------------------------------------
derivmr_calc@radtra_m       pri Calculation of derivatives with respect to vmr
                                parameters
-------------------------------------------------------------------------------
derivmrgrad_calc@radtra_m   pri Calculation of derivatives with respect to
                                vmr-gradient parameters
-------------------------------------------------------------------------------
envfovils@ilsfov_m          pub AILS and FOV convolution for ENVISAT
-------------------------------------------------------------------------------
envils@ilsfov_m             pub AILS convolution for Envisat
-------------------------------------------------------------------------------
epsi@ray_m                  pri calculate refraction index-1 at cartesian
                                point r in atmosphere
-------------------------------------------------------------------------------
error@varsub_m              pub stop program
-------------------------------------------------------------------------------
extend_mw@input_m           pri Extend the microwindows on each side by the
                                ilsradius.
-------------------------------------------------------------------------------
facds@ray_m                 pri find, along a direction vector dr, with
                                endpoint heights h1 and h2 the point with
                                height h
-------------------------------------------------------------------------------
findtop@ray_m               pri find (coming from outside) the crossing point
                                with top level of atmosphere
-------------------------------------------------------------------------------
```

```
finewrk@radtra_m           pri determine the vector wrkrad which contains the
                               spectra for all geometries to the used for
                               field-of-view
------------------------------------------------------------------------------
fovils1@ilsfov_m           pub AILS and FOV convolution for linear aperture
------------------------------------------------------------------------------
fovils_fixed@ilsfov_m      pub AILS and FOV convolution for read-in
                               AILS-function
------------------------------------------------------------------------------
fovilsspec@ilsfov_m        pri determines finite FOV taking 2nd derivative of
                               vertically varying spectrum into account
                               (thus modifying resulting lineshape)
------------------------------------------------------------------------------
fovmeanspec@ilsfov_m       pri determines spectrum representative for FOV
                               FOV is assumed to extend vertically around
                               elevcenter from
                               -vertfovradius...+vertfovradius
                               FOV is described by weights of nvertfov
                               horizontal stripes of equal width covering the
                               FOV. Therefore the used elevations extend over
                               a slightly narrower range than
                               -vertfovradius...+vertfovradius
------------------------------------------------------------------------------
g@varsub_m                 pri Calculation of earth gravitational
                               acceleration depending on altitude and
                               latitude: slightly modified version of:
                                 - Redemann (1984)
                                 - Clarmann (1986)
------------------------------------------------------------------------------
gauher@recipe_m            pri numerical recipe to calculate abscissas and
                               weights for Gauss-Hermite integration
------------------------------------------------------------------------------
geodistance@give_m         pri determine the distance (in terms of dx[km] and
                               dy[km] of the point rlat,rlon
                               from the profile location
------------------------------------------------------------------------------
give_aerabs@give_m         pub give the aerosol absorption for a distinct
                               altitude
------------------------------------------------------------------------------
give_aersca@give_m         pub give the aerosol scattering for a distinct
                               altitude
------------------------------------------------------------------------------
give_altbase@give_m        pub give the altitude value to a given index

------------------------------------------------------------------------------
give_brdaero_prf           pub give the broadband aerosol parameters for a
                               certain altitude
------------------------------------------------------------------------------
give_brdaero_wave          pub give the broadband-aerosol absorption and
                               extinction cross-sections (no Mie-model)
                               or the real and
                               imaginary refraction indices for a distinct
                               altitude (Mie-model)
------------------------------------------------------------------------------
```

```
give_p@give_m              pub give the pressure for a distinct altitude and
                               temperature at this altitude
-----------------------------------------------------------------------------
give_pgrad@give_m          pri give the p-gradients for a given altitude
-----------------------------------------------------------------------------
give_t@give_m              pub give the temperature for a distinct altitude
-----------------------------------------------------------------------------
give_Tgrad@give_m          pri give the T-gradients for a given altitude
-----------------------------------------------------------------------------
give_tvib@give_m           pub give the vibrational temperature for a
                               distinct altitude
-----------------------------------------------------------------------------
give_Tvibgrad@give_m       pri give the vmr-gradients for a given altitude
-----------------------------------------------------------------------------
give_vmr@give_m            pub give the vmr for a distinct altitude
-----------------------------------------------------------------------------
give_vmrgrad@give_m        pri give the vmr-gradients for a given altitude
-----------------------------------------------------------------------------
go_next@varsub_m           pri goto next '§' in input file ifile
-----------------------------------------------------------------------------
gradient@ray_m             pri calculate gradient of (refraction index-1)
                               around point r
-----------------------------------------------------------------------------
grid_t_hw@modlev_m         pri fine-level gridding using T-differences and
                               half-width changes
-----------------------------------------------------------------------------
grid_tang@modlev_m         pri fine-level gridding due to levels above
                               tangent points
-----------------------------------------------------------------------------
height@ray_m               pri calculate distance to surface of ellipse
-----------------------------------------------------------------------------
homog_path@rayctl_m        pri determine the path parameters for homogeneous
                               path calculation
-----------------------------------------------------------------------------
ilsapo@ilsfov_m            pub AILS convolution for circular aperture
-----------------------------------------------------------------------------
ilsapo_fixed@ilsfov_m      pub AILS convolution for read-in AILS-function
-----------------------------------------------------------------------------
ilsapofov_calc@radtra_m    pri convolution of the fine-grid spectra add their
                               derivatives with the apodized instrumental
                               fine shape and field of view calculation
-----------------------------------------------------------------------------
ils_radius@input_m         pri Determination of the radius (in multiples of
                               the fine grid distance wgrid%fine) where the
                               chosen apodisation function 'accu%iapo' is
                               decreased to 1, 0.1, or 0.01% of it's centre
                               value.
-----------------------------------------------------------------------------
index_nlte@ray_m           pri gives for species ispeci and state istate the
                               index in 'tot' vector where Tvibs are stored
-----------------------------------------------------------------------------
ini_deri@inider_m          pub initialize deri% variable
-----------------------------------------------------------------------------
ini_output@iniout_m        pub initialize data vector where the output
```

```
                             spectrum is stored be careful: the
                             sub-microwindow index is on the external!
                             sub-microwidows
-------------------------------------------------------------------------------
ini_para@inipar_m        pub Initialization of parameter vector para%.
-------------------------------------------------------------------------------
ini_para_aerabs@inipar_m pri initialization of parameter vector for aerosol
                             absorption coefficient parameters
-------------------------------------------------------------------------------
ini_para_Mie@inipar_m    pri initialization of parameter vector for
                             Mie model
-------------------------------------------------------------------------------
ini_para_num@inipar_m    pri initialization of parameter vector for
                             broadband aerosol number density
-------------------------------------------------------------------------------
ini_para_p@inipar_m      pri initialization of parameter vector for
                             pressure parameters
-------------------------------------------------------------------------------
ini_para_T@inipar_m      pri initialization of parameter vector for
                             temperature parameters
-------------------------------------------------------------------------------
ini_para_Tgrad@inipar_m  pri initialisation of parameter vector for
                             temperature derivative parameters
-------------------------------------------------------------------------------
ini_para_Tvib@inipar_m   pri initialization of parameter vector for Tvib
                             parameters
-------------------------------------------------------------------------------
ini_para_vmr@inipar_m    pri initialization of parameter vector for vmr
                             parameters
-------------------------------------------------------------------------------
ini_para_vmrgrad@inipar_m  pri initialisation of parameter vector for vmr
                             parameters
-------------------------------------------------------------------------------
ini_sub_mw@inismw_m      pub determines internal forward model sub-
                             microwindows
-------------------------------------------------------------------------------
input@input_m            pub controls the input from files and defines
                             input variables
-------------------------------------------------------------------------------
input_brdaero_prf@input_m  pri read broadband-aerosol altitude profiles of
                             number density, mode-radii and mode-widths
-------------------------------------------------------------------------------
input_brdaero_wave@input_m pri read broadband-aerosol wavenumber and altitude
                             dependent absorption and extinction
                             cross-sections or real and imaginary
                             refraction indices
-------------------------------------------------------------------------------
input_contprof@input_m   pri reads the continuum profiles from file ifile.
-------------------------------------------------------------------------------
input_hitmol@input_m     pri reads the hitran molecule information from
                             file
-------------------------------------------------------------------------------
input_isoprof@input_m    pri reads the isotope abundance profiles from
                             file ifile. These profiles are used to
```

```
                                scale the vmr profiles if single isotopes
                                have to be calculated
-------------------------------------------------------------------------------
input_main@input_m          pri -reads the data from the main input file
                                -defines the species
                                -extends the microwindows due to ils
-------------------------------------------------------------------------------
input_mwdef@input_m         pri reads microwindow definition section of main
                                input file
-------------------------------------------------------------------------------
input_pTgradprof@input_m    pri reads the p,T - gradient profiles from file
                                ifile.
-------------------------------------------------------------------------------
input_pTprof@input_m        pri reads the p,T profiles from file ifile.
-------------------------------------------------------------------------------
input_spectroscopy@inspec_m pub Controls the input of the spectroscopic data,
                                i.e. the determination of the spectroscopic
                                data type construction
                                speci%iso%band%branch%line.
                                This is performed for different options:
                                - no line mixing, no nlte
                                - no line mixing, nlte
                                - line mixing, no nlte
                                - line mixing, nlte
-------------------------------------------------------------------------------
input_Tvibgradprof@input_m  pri reads the Tvibr profiles from file ifile.
-------------------------------------------------------------------------------
input_Tvibprof@input_m      pri reads the Tvib profiles from file ifile.
-------------------------------------------------------------------------------
input_vmrgradprof@input_m   pri reads the vmr - gradient profiles from file
                                ifile.
-------------------------------------------------------------------------------
input_vmrprof@input_m       pri reads the vmr profiles from file ifile.
-------------------------------------------------------------------------------
integrate@ray_m             pri explicit integration for layer values and
                                derivatives
                                (columns, Curtis-Godson-T, -p, -Tvib, ...)
-------------------------------------------------------------------------------
interface_nlte@ifnlte_m     pub Interface to nlte-model
-------------------------------------------------------------------------------
interpolate_grid@addlin_m   pub interpolates the intervals of the lower tiers
                                to the finest tier used in the summation of
                                lines
-------------------------------------------------------------------------------
isoabun@inspec_m            pri multiply line strength with isotope abundancy
                                if species is single isotope
-------------------------------------------------------------------------------
isomult@input_m             pri multiplication of the vmr input profiles
                                of one molecule with the isotope abundance
                                profile in the case a species is a single
                                isotope
-------------------------------------------------------------------------------
kopra_derip@kopfwd_m        pri Perform numeric p-derivatives
-------------------------------------------------------------------------------
```

```
kopra_forwrd@kopfwd_m        pub Performs Kopra forward calculation and
                                 controls numeric p-derivatives

-----------------------------------------------------------------------------
latlon@ray_m                 pri calculate geographic latitude and longitude
                                 (first point) of cartesian point r

-----------------------------------------------------------------------------
leveltang@ray_m              pri perform integration for tangent layer and find
                                 exactly tangent position

-----------------------------------------------------------------------------
leveltrans@ray_m             pri perform integration for all layers except
                                 tangent layer and find exact level positions

-----------------------------------------------------------------------------
linein@inspec_m              pri determines if the line should be read or not
-----------------------------------------------------------------------------
line_strength@abco_m         pri Calculates the line intensities and optionally
                                 the T-derivatives for a bundle of lines

-----------------------------------------------------------------------------
lsimobs@radtra_m             pri determine lsim and lobs, the activated
                                 simulated and observed geometries i.e. the
                                 ones which are influenced by the actual
                                 retrieval parameter lpara

-----------------------------------------------------------------------------
next_para@ray_m              pri  calculate the index for the next p-parameter
                                  which influences the lowest part of the
                                  actual geometry (which is influenced most
                                  by refraction)

-----------------------------------------------------------------------------
nmax_calc@ray_m              pri determination of the max. number of
                                 integration variables

-----------------------------------------------------------------------------
numdata@inspec_m             pri Determination of the spectroscopic data files
                                 to be opened and allocation of dummy vector
                                 spe()%... where the line data are stored
                                 temporarily

-----------------------------------------------------------------------------
makecils@ilsfov_m            pri Generation of ILS from complex interferogram
-----------------------------------------------------------------------------
makeenvifg@ilsfov_m          pri generates complex IFG (MIPAS-ENVISAT)
-----------------------------------------------------------------------------
make_hydroequi@input_m       pri Transfer of the input profiles into
                                 hydrostatic equilibrium: (the pressures or the
                                 altitudes are adjusted)

-----------------------------------------------------------------------------
makeifg@ilsfov_m             pri generates a real, single sided modulation
                                 efficiency interferogram

                                 kind of apodisation
                                 1= boxcar in ifg == sinc in spectrum
                                 2= triangle in ifg == sinc^2 in spectrum
                                 3= Hamming (Happ-Genzel)
                                 4= 3 term Blackmann-Harris
                                 5= 4 term Blackmann-Harris
                                 6= Norton-Beer weak
                                 7= Norton-Beer medium
```

```
                             8= Norton-Beer strong

                             selfapo=T/F with/without selfapodisation by
                             evenly illuminated FOV
                             ifg(1):= ZPD
-------------------------------------------------------------------------------
makeils@ilsfov_m         pri generates ILS from given interferogram
-------------------------------------------------------------------------------
make_mainspeci@input_m   pri determine main species number sw%mainspeci
-------------------------------------------------------------------------------
make_modelgeo@modgeo_m   pub introduces additional geometries for the
                             simulation of FOV and defines the new
                             occupation matrix for all the simulated
                             geometries ('simulated geometries' are called
                             all geometries to be calculated in the forward
                             model i.e. the observed plus the additional
                             geometries for FOV)
-------------------------------------------------------------------------------
make_modelgrid@modlev_m  pri Defines the model altitude levels for the
                             forward calculation. Starting from a base
                             grid, levels are added in order to fulfill
                             criteria on the T difference and half-width
                             change between levels. Then a criterium for
                             smaller levels distances directly over tangent
                             altitudes is applied. At the end all levels
                             which are less distant than a threshold are
                             deleted.
-------------------------------------------------------------------------------
make_occusim@modgeo_m    pri determines the microwindow occupation matrix
                             for the simulated geometries


-------------------------------------------------------------------------------
mie@miemod_m             pub calculate
                             extinciton and absorption cross sections
                             for broadband-aerosol and their derivatives
                             with respect to the distribution parameters
-------------------------------------------------------------------------------
mieampli@miemod_m        pri calculation of the Mie-amplitude function
-------------------------------------------------------------------------------
Mie_call@abco_m          pri call the Mie model in order to calculate
                             extinciton and absorption cross sections
                             for broadband-aerosol and their derivatives
                             with respect to the distribution parameters
-------------------------------------------------------------------------------
miecoe@miemod_m          pri calculate the Mie coefficients for given
                             refraction indices
-------------------------------------------------------------------------------
min_distance@modlev_m    pri levels which are less distant than accu%dmin
                             are selected out (only the ones that do not
                             belong to the base-grid)
-------------------------------------------------------------------------------
multiple_cut@varsub_m    pri multiple (direct following) elements of an
                             integer vector are deleted
-------------------------------------------------------------------------------
```

```
nlte_postderi@nltder_m       pub calculate post-derivatives in case of
                                 use of nlte-model
--------------------------------------------------------------------------------
nlte_vmr_postderi@nltder_m   pri calculate post-derivatives for vmr
                                 derivatives (dependence on nlte)
--------------------------------------------------------------------------------
nlte_x_postderi@nltder_m     pri calculate post-derivatives for
                                 nlte-model parameters
--------------------------------------------------------------------------------
observer@ray_m               pri calculate position and viewing direction of
                                 observer in cartesian coordinates
--------------------------------------------------------------------------------
off@offsca_m                 pri add offset to spectrum
--------------------------------------------------------------------------------
off_deri@offsca_m            pri calculate offset derivatives
--------------------------------------------------------------------------------
offset_scale@offsca_m        pub multiply spectrum and derivatives by factor
                                 add offset to spectrum calculate offset and
                                 scale derivatives
--------------------------------------------------------------------------------
openerror@varsub_m           pub stop program if error occurs while opening a
                                 file
--------------------------------------------------------------------------------
optsrcdTvib@radtra_m         pri Calculation of derivative of optical thickness
                                 and source function with respect to Tvib
--------------------------------------------------------------------------------
para_aerabs_change@parchk_m  pub checks if variation of a distinct aerosol
                                 absorption parameter influences the absorption
                                 values inside an altitude region of the
                                 atmosphere (for all microwindows)
--------------------------------------------------------------------------------
para_dabsopt_ne0@rayctl_m    pri determine the parameters which do not
                                 influence the aerosop absorption optical depth
                                 i.e. for which the derivative daeropt = 0
--------------------------------------------------------------------------------
para_dbrdaerocol_ne0@rayctl_m pri determine the broadband aerosol number density
                                 parameters which do not
                                 influence the aerosol column
--------------------------------------------------------------------------------
para_dcol_ne0@rayctl_m       pri determine the parameters which do not
                                 influence the partial columns i.e. for which
                                 the derivative dcol = 0
--------------------------------------------------------------------------------
para_dcolgrad_ne0@rayctl_m   pri determine the vmr gradient parameters which do
                                 not influence the partial columns i.e. for
                                 which the derivative dcol = 0
--------------------------------------------------------------------------------
para_dMie_ne0@rayctl_m       pri determine the Mie model-
                                 parameters which do not
                                 influence the CG-Mie layer value
--------------------------------------------------------------------------------
para_dp_ne0@rayctl_m         pri determine the p parameters which do not
                                 influence the cg-p of air i.e. for which the
                                 derivative dp = 0
```

```
------------------------------------------------------------------------------
para_dT_ne0@rayctl_m         pri determine the T parameters which do not
                                 influence the cg-T of air i.e. for which the
                                 derivative dT = 0
------------------------------------------------------------------------------
para_dTgrad_ne0@rayctl_m     pri determine the Tgrad parameters which do not
                                 influence the cg-T of air i.e. for which the
                                 derivative dTgrad = 0
------------------------------------------------------------------------------
para_dTvib_ne0@rayctl_m      pri determine the parameters which do not
                                 influence the Tvib's i.e. for which the
                                 derivative dTvib_cg_dT = 0
------------------------------------------------------------------------------
para_Mie_change@parchk_m     pub checks if variation of a distinct Mie model
                                 parameter influences the
                                 values inside an altitude region of the
                                 atmosphere
------------------------------------------------------------------------------
para_num_change@parchk_m     pub checks if variation of a distinct broadband
                                 aerosol number density parameter influences the
                                 number density values inside an altitude
                                 region of the atmosphere
------------------------------------------------------------------------------
para_p_change@parchk_m       pub checks if variation of a distinct p parameter
                                 influences the p values inside an altitude
                                 region of the atmosphere
------------------------------------------------------------------------------
para_T_change@parchk_m       pub checks if variation of a distinct T parameter
                                 influences the T values inside an altitude
                                 region of the atmosphere
------------------------------------------------------------------------------
para_Tgrad_change@parchk_m   pub checks if variation of a distinct Tgrad
                                 parameter influences the T values inside an
                                 altitude region of the atmosphere
------------------------------------------------------------------------------
para_Tvib_change@parchk_m    pub checks if variation of a distinct Tvib
                                 parameter influences the Tvib values inside an
                                 altitude region of the atmosphere
------------------------------------------------------------------------------
para_vmr_change@parchk_m     pub checks if variation of a distinct vmr
                                 parameter influences the vmr values inside an
                                 altitude region of the atmosphere
------------------------------------------------------------------------------
para_vmrgrad_change@parchk_m pub checks if variation of a distinct vmr gradient
                                 parameter influences the vmr values inside an
                                 altitude region of the atmosphere
------------------------------------------------------------------------------
pathcopy@ray_m               pri copy path parameters into vector geo%...
                                 for geometry part ipart
------------------------------------------------------------------------------
pointmw@inspec_m             pri determine for each microwindow i the range of
                                 lines which will be used: speci()%...%mw_l1(i)
                                 speci()%...%mw_l2(i).
------------------------------------------------------------------------------
```

```
press@varsub_m              pri Calculation of the pressure using mean
                                temperature of the layer and g at the lower
                                boundary level
-----------------------------------------------------------------------------
radtrans@radtra_m           pub Calculation of the radiative transfer through
                                the atmosphere and determination of the
                                fine-grid spectra and their derivatives.
                                Convolution of the fine-grid spectra with the
                                ails to get the spectra on the coarse-grid
                                (measurement grid) and calculation of the
                                field-of-view weighting.
-----------------------------------------------------------------------------
radtrans_mw@radtra_m        pri perform radiative transfer for spectra and
                                calculate derivatives on the non-equidistant
                                fine-grid
-----------------------------------------------------------------------------
ray_out@ray_m               pri prepare output variable geo()%... which
                                contains all path parameters
-----------------------------------------------------------------------------
raytra@ray_m                pub calculation of ray-tracing in inhomogeneous
                                atmosphere and path integration
-----------------------------------------------------------------------------
raytrace_ctrl@rayctl_m      pub Controls ray-tracing, calculation of path
                                integrated values
                                (Curtis-Godson values and column amounts) and
                                the derivatives of path integrated values
                                wrt retrieval parameters.
-----------------------------------------------------------------------------
readlines@inspec_m          pri read in spectroscopic data
-----------------------------------------------------------------------------
read_lmdata@inspec_m        pri read the line mixing data from file
                                fil%linemix into vector 'lm'. Only the
                                branches which are 'near' a microwindow
                                (distance w_linemix) are read.
-----------------------------------------------------------------------------
refin@ray_m                 pri calculate refraction index for pressure px and
                                temperatute tx
-----------------------------------------------------------------------------
sca@offsca_m                pri multiply spectrum and derivatives by scale
-----------------------------------------------------------------------------
sca_deri@offsca_m           pri calculate scale derivatives
-----------------------------------------------------------------------------
scndderfov@ilsfov_m         pri second derivative with respect to elevation
                                representative for FOV
                                (notice: unit of angle is vertfovradius)
-----------------------------------------------------------------------------
set_gauher@miemod_m         pri determine the Gauss-Hermite abscissas and
                                weights for the particle distribution and
                                its derivative with respect to the
                                distribution parameters
-----------------------------------------------------------------------------
set_miepar@miemod_m         pri set the discretization for the scattering
                                angles
-----------------------------------------------------------------------------
```

```
shiftwvnr@ilsfov_m          pri performs shift of spectrum along abszissa
-------------------------------------------------------------------------------
speciorder@input_m          pri determination of species describing variables
-------------------------------------------------------------------------------
sub_mw_minmax@radtra_m      pri deternine first and last index in equidistant
                                fine grid spectrum for one sub_microwindow
-------------------------------------------------------------------------------
tangpara@ray_m              pri determine tangent height and distances by
                                parabolic interpolation
-------------------------------------------------------------------------------
tangalt@ray_m               pub calculate tangent altitude (not exact)
-------------------------------------------------------------------------------
tausrc@radtra_m             pri Calculation of layer transmission tau and
                                layer source function src (nlte considered)
-------------------------------------------------------------------------------
tnew@ray_m                  pri calculation of new tangent vector along LOS
-------------------------------------------------------------------------------
unitvector@ray_m            pri calculate unit vector of vector vr
-------------------------------------------------------------------------------
value_test@ilsfov_m         pri Test for change of variable
-------------------------------------------------------------------------------
vektorsin@ray_m             pri sine of angle between two vectors
-------------------------------------------------------------------------------
vertinterpol@ilsfov_m       pri interpolates linear in elevation between given
                                spectra
-------------------------------------------------------------------------------
voigt@addlin_m              pri Calculates the Voigt-Function times a
                                user-definied value fac with a relative
                                accuracy better than 2*10-4. The algorithm
                                switches automatically from the calculation of
                                the Voigt-Function to the calculation of the
                                Lorentz-Function.
                                If this subroutine is called several times
                                with similar values y, the numerically
                                expensive coefficents a1..t8 are only
                                calculated once. The coefficients are only
                                recalculated if the relative change in y is
                                grater than the internal parameter rel, which
                                is set to 1e-4.
                           ref: A new implementation of the Humlicek algorithm
                                for the calculation of the Voigt profile
                                function, J. Quant. Spectrosc. Radiat.
                                Transfer, 57, 819-824, 1997
-------------------------------------------------------------------------------
warning@varsub_m            pub write warning
-------------------------------------------------------------------------------
writeabco@abco_m            pri printout of absorption coefficient
-------------------------------------------------------------------------------
writeabco@radtra_m          pri write in file absco_type spectra
-------------------------------------------------------------------------------
writeout@wriout_m           pub write the spectra and derivatives on the
                                coarse wavenumber grid on file
-------------------------------------------------------------------------------
writespec@radtra_m          pri write in file spectrum stored in vector
```

```
------------------------------------------------------------------------------
writespec@wriout_m           pri write one spectrum on file
------------------------------------------------------------------------------
y_calc_dd@linmix_m           pri calculates the y-coefficients for direct
                                 diagonalisation
------------------------------------------------------------------------------
y_calc_rk@linmix_m           pri calculates the y-coefficients for the
                                 Rosenkranz-approximation
------------------------------------------------------------------------------
yinterpol1@varsub_m          pri returns the linearly interpolated and
                                 extrapolated y value when the x value and a
                                 grid y(n) and x(n) is given
------------------------------------------------------------------------------
------------------------------------------------------------------------------
```

3 Variable Listing and Description

```
==============================================================================

accu%
       accuracy_type : computational accuracy
accu%absolute_absco
                real : absolute accuracy for optical depth
              Origin : input_main@input_m
accu%basealt()
                real : altitude of the base-levels [km]
           Dimension : n%baselev
              Origin : input_main@input_m
accu%dif
                real : maximum layer thickness above the tangent altitude [km]
              Origin : input_main@input_m
accu%dmin
                real : minimal layer thickness [km]
              Origin : input_main@input_m
accu%iapo
             integer : accuracy of apodisation (defines mw extension)
                       1= apodisation function decreases to  1%    of center value
                       2= apodisation function decreases to  0.1%  of center value
                       3= apodisation function decreases to  0.01% of center value
              Origin : input_main@input_m
accu%iexpath
             integer : number of extra paths where the absorption
                       coefficients are recalculated/interpolated
              Origin : input_main@input_m
accu%ifov()
             integer : criteria for addition of simulated
                       geometries for field of view calculation
           Dimension : 2
              Origin : input_main@input_m
```

```
accu%ifov_sep
            integer : number of the geometry up to which ifov(1)
                      is used as criterium for addition of
                      simulated geometries
             Origin : input_main@input_m
accu%raytrace
               real : integration step length [km]
             Origin : input_main@input_m
accu%retain_absco
            integer : =0 lines my be rejected and truncated
                      =1 all lines retained, no truncation
             Origin : input_main@input_m
accu%Tvar1
               real : max. temperature variation between
                      levels in the lower altitude region
             Origin : input_main@input_m
accu%Tvar2
               real : max. temperature variation between
                      levels in the higher altitude region
             Origin : input_main@input_m
accu%w_useline
               real : wavenumber range outside mw where lines
                      are taken into account for
                      calculation of each line
             Origin : input_main@input_m
accu%wvar
               real : maximum variation of the line width
                      between two model levels
             Origin : input_main@input_m
accu%upto
               real : altitude above the tangent up to which
                      this maximum thickness (accu%dif) is valid [km]
             Origin : input_main@input_m
accu%zTvar
               real : altitude [km] which divides the regions
                      for temperature variation between levels
             Origin : input_main@input_m
===============================================================================
brdaero%
       brdaero_type : broadband-aerosol definition

brdaero%alt_prf()
               real : altitudes of broadband-aerosol profile levels [km]
          Dimension : brdaero%nlev_prf
             Origin : input_brdaero_prf@input_m
brdaero%alt_wave()
               real : altitudes of levels of broadband-aerosol
                      refraction indices or cross sections [km]
          Dimension : brdaero%nlev_wave
             Origin : input_brdaero_wave@input_m
brdaero%iwave1()
            integer : begin (1,:)  wavenumber index for broadband-aerosol
                      refraction indices or cross sections for each mw
          Dimension : n%mw
```

```
            Origin : input_brdaero_wave@input_m
brdaero%nlev_prf
           integer : number of levels for the broadband-aerosol
                     profiles
            Origin : input_brdaero_prf@input_m
brdaero%nlev_wave
           integer : number of altitude levels for the broadband-aerosol
                     refraction indices or cross sections
            Origin : input_brdaero_wave@input_m
brdaero%nMie
           integer : number of Mie-parameter sets
            Origin : input_main@input_m
brdaero%nmode
           integer : number of modes for broadband-aerosol
                     particle distribution function
            Origin : input_brdaero_prf@input_m
brdaero%nwave()
           integer : number of wavenumber indices for broadband-aerosol
         Dimension : n%mw
            Origin : input_brdaero_wave@input_m
==============================================================================

brdaero_prf(:,:)
              real : broadband-aerosol profiles of particle number density
                     [particles/cm3] and mode-radii [micrometer] and
                     mode-widths and ratio of mode1 to mode1+mode2
         Dimension :(brdaero%nlev_prf,6) if sw%Miemod (Mie-model)
                    (brdaero%nlev_prf,1) if not sw%Miemod (no Mie-model)
            Origin : input_brdaero_prf@input_m
==============================================================================

brdaero_wave(:,:,:,:)
              real : broadband-aerosol wavenumber
                     dependent refraction index (if sw%Miemod)
                     or cross section [cm2/particle] (if not sw%Miemod)
         Dimension : (i,2,brdaero%nlev_wave,n%mw) with
                     i is max number of aerosol grid points for all mw's
==============================================================================

deri%
   derivative_type : derivative definition
deri%igasiso()
           integer : gas/isotope identifier for deriv. species
         Dimension : deri%nspeci
            Origin : input_main@input_m
deri%igasiso_grad()
           integer : gas/isotope identifier for vmr-gradientderiv. species
         Dimension : deri%n_vmrgrad_speci
            Origin : input_main@input_m
deri%igasiso_nlte
           integer : Tvib species for derivative
            Origin : input_main@input_m
deri%i_nlte_iso
           integer : global nlte_isotope number
```

```
                       for which Tvib derivatives are calculated
              Origin : ini_deri@inider_m
deri%i_nlte_speci
             integer : global species number for which Tvib
                       derivatives are calculated
              Origin : ini_deri@inider_m
deri%i_nlte_state()
             integer : pointer to the global nlte state number
           Dimension : deri%n_nlte_state
              Origin : ini_deri@inider_m
deri%istate()
             integer : states for which T-vib derivatives
                       should be calcualted
           Dimension : deri%n_nlte_state
              Origin : input_main@input_m
deri%i_vmrgrad_speci()
             integer : pointer to global species numbering
           Dimension : deri%n_vmrgrad_speci
              Origin : ini_deri@inider_m
deri%i_vmr_speci()
             integer : pointer to global species numbering
           Dimension : deri%n_vmr_speci
              Origin : ini_deri@inider_m
deri%nlte_speci
             integer : pointer to the nlte-species of nlte()%..input variable
              Origin : ini_deri@inider_m
deri%n_nlte_state
             integer : number of T-vib states for which derivatives
                       should be calculated
              Origin : input_main@input_m
deri%n_vmrgrad_speci
             integer : number of species for vmr-gradient derivative
              Origin : input_main@input_m
deri%n_vmr_speci
             integer : number of species for vmr derivative
              Origin : input_main@input_m
===============================================================================

fil%
      filenames_type : input file names and directories
fil%ails
      character(200) : apodized instrumental line shape
              Origin : input_main@input_m
fil%brdaero_cross
      character(200) : particle cross sections for broadband-aerosol
                       absorption and extinction
              Origin : input_main@input_m
fil%brdaero_prf
      character(200) : profiles for broadband-aerosol number
                       density and size distribution parameters
              Origin : input_main@input_m
fil%brdaero_refind
      character(200) : wavenumber dependent refraction indice
                       profiles for broadband-aerosol
```

```
             Origin : input_main@input_m
fil%contprof
      character(200) : continuum profiles
             Origin : input_main@input_m
fil%cutdop
      character(200) : doppler cut-off for abco calculation
             Origin : input_main@input_m
fil%cutlor
      character(200) : lorentz cut-off for abco calculation
             Origin : input_main@input_m
fil%err
      character(200) : output error file
             Origin : input_main@input_m
fil%errunit
            integer : output error file unit number
             Origin : input_main@input_m
fil%hitmol
      character(200) : data on hitran molecules, isotopes, states
             Origin : input_main@input_m
fil%isoprof
      character(200) : isotope abundance profiles
             Origin : input_main@input_m
fil%linedatadir
      character(200) : spectrospcopic data directory
             Origin : input_main@input_m
fil%linemix
      character(200) : line-mixing data
             Origin : input_main@input_m
fil%log
      character(200) : output log file
             Origin : input_main@input_m
fil%logunit
            integer : output log file unit number
             Origin : input_main@input_m
fil%outcoarse
      character(200) : spectra on coarse wavenumber grid
             Origin : input_main@input_m
fil%outfine
      character(200) : spectra on fine wavenumber grid
             Origin : input_main@input_m
fil%ptgradprof
      character(200) : p,T - gradient - profiles
             Origin : input_main@input_m
fil%ptprof
      character(200) : p,T - profiles
             Origin : input_main@input_m
fil%Tvibgradprof
      character(200) : Tvib-gradient profiles
             Origin : input_main@input_m
fil%Tvibprof
      character(200) : Tvib profiles
             Origin : input_main@input_m
fil%vmrgradprof
      character(200) : vmr - gradient - profiles
```

```
             Origin : input_main@input_m
fil%vmrprof
      character(200) : vmr - profiles
             Origin : input_main@input_m
fil%Xdatadir
      character(200) : cross-section data directory
             Origin : input_main@input_m
==============================================================================

geo()%
           geo_type : path definition
          Dimension : n%simgeo
geo()%nparts
            integer : number of parts of the geometry
             Origin : allocgeo1@rayctl_m, homog_path@rayctl_m

geo()%par()%
           par_type : parts of geometry (1:far and 2:near observer)
          Dimension : geo()%nparts
geo()%par()%l_high
            integer : highest layer number
             Origin : ray_out@ray_m, homog_path@rayctl_m
geo()%par()%l_low
            integer : lowest layer number
             Origin : ray_out@ray_m, homog_path@rayctl_m

geo()%par()%lay()%
           lay_type : layers description
          Dimension : (geo()%par()%l_low,geo()%par()%l_high)
geo()%par()%lay()%absopt()
               real : aerosol absorption optical depth
          Dimension : n%mw
             Origin : pathcopy@ray_m, homog_path@rayctl_m
geo()%par()%lay()%brdaerocol
               real : broadband-aerosol column [particles/cm2]
             Origin : pathcopy@ray_m
geo()%par()%lay()%col
               real : total air column in layer [molecules/cm2]
             Origin : pathcopy@ray_m, homog_path@rayctl_m
geo()%par()%lay()%cross(:,:,:)
               real : broadband-aerosol absorption and extinction
                      cross-sections [cm2/particle]
          Dimension : size(brdaero_wave,1),2,n%mw
             Origin : pathcopy@ray_m
geo()%par()%lay()%dabsopt()
               real : derivative of aerosol absorption optical
                      depth with respect to each aerosol
                      absorption coefficient parameter for each mw
          Dimension : (n%mw,para%n_aerabs_para)
             Origin : pathcopy@ray_m
geo()%par()%lay()%dbrdaerocol()
               real : derivative of broadband aerosol column
                      with respect to each broadband aerosol
                      number density parameter
```

```
           Dimension : para%n_brdaeronum_para
              Origin : pathcopy@ray_m
geo()%par()%lay()%dcol()
                real : derivative of air column
                       with respect to each p-parameter
           Dimension : para%n_p_para, deri_p_calc@ray_m
              Origin : pathcopy@ray_m
geo()%par()%lay()%dcross(:,:,:,:)
                real : derivatives of broadband-aerosol absortion
                       and extinction cross-sections with respect to
                       C.G. Mie layer values
           Dimension : size(brdaero_wave,1),2,brdaero%nMie,n%mw
              Origin : pathcopy@ray_m
geo()%par()%lay()%dMie(:,:)
                real : derivative of Mie C.-G. parameters
           Dimension : brdaero%nMie, para%n_Mie_para
              Origin : pathcopy@ray_m
geo()%par()%lay()%dp()
                real : derivative of cg-p of air with
                       respect to each p-parameter
           Dimension : para%n_p_para, deri_p_calc@ray_m
              Origin : pathcopy@ray_m
geo()%par()%lay()%dT()
                real : derivative of cg-T of air with
                       respect to each T-parameter
           Dimension : para%n_T_para
              Origin : pathcopy@ray_m
geo()%par()%lay()%dTgrad()
                real : derivative of cg-T of air with
                       respect to each Tgrad-parameter
           Dimension : para%n_Tgrad_para
              Origin : pathcopy@ray_m
geo()%par()%lay()%extopt()
                real : aerosol extinction optical depth
           Dimension : n%mw
              Origin : pathcopy@ray_m, homog_path@rayctl_m
geo()%par()%lay()%i_dabsopt_ne0()
             integer : dabsopt index for dabsopt /= 0
           Dimension : geo()%par()%lay()%n_dabsopt_ne0
              Origin : para_dabsopt_ne0@rayctl_m
geo()%par()%lay()%i_dbrdaerocol_ne0()
             integer : dbrdaerocol index for dbrdaerocol /= 0
           Dimension : geo()%par()%lay()%n_dbrdaerocol_ne0
              Origin : para_dbrdaerocol_ne0@rayctl_m
geo()%par()%lay()%i_dMie_ne0
             integer : dMie index for dMie /= 0
              Origin : para_dMie_ne0@rayctl_m
geo()%par()%lay()%i_dp_ne0()
             integer : dp index for dp /= 0
           Dimension : geo()%par()%lay()%n_dp_ne0
              Origin : para_dp_ne0@rayctl_m
geo()%par()%lay()%i_dTgrad_ne0()
             integer : dTgrad index for dTgrad /= 0
           Dimension : geo()%par()%lay()%n_dTgrad_ne0
```

```
              Origin : para_dTgrad_ne0@rayctl_m
geo()%par()%lay()%i_dT_ne0()
             integer : dT index for dT /= 0
           Dimension : geo()%par()%lay()%n_dT_ne0
              Origin : para_dT_ne0@rayctl_m
geo()%par()%lay()%n_dabsopt_ne0
             integer : number of dabsopt values which are /= 0
              Origin : para_dabsopt_ne0@rayctl_m
geo()%par()%lay()%n_dbrdaerocol_ne0
             integer : number of dbrdaerocol values which are /= 0
              Origin : para_dbrdaerocol_ne0@rayctl_m
geo()%par()%lay()%n_dMie_ne0
             integer : number of dMie values which are /= 0
              Origin : para_dMie_ne0@rayctl_m
geo()%par()%lay()%n_dp_ne0
             integer : number of dp values which are /= 0
              Origin : para_dp_ne0@rayctl_m
geo()%par()%lay()%n_dTgrad_ne0
             integer : number of dTgrad values which are /= 0
              Origin : para_dTgrad_ne0@rayctl_m
geo()%par()%lay()%n_dT_ne0
             integer : number of dT values which are /= 0
              Origin : para_dT_ne0@rayctl_m
geo()%par()%lay()%path
                real : path length in layer [km]
              Origin : pathcopy@ray_m, homog_path@rayctl_m
geo()%par()%lay()%pcg
                real : curtis godson pressure for air [hPa]
              Origin : pathcopy@ray_m, homog_path@rayctl_m
geo()%par()%lay()%refind(:,:,:)
                real : broadband-aerosol CG refraction index values
           Dimension : size(brdaero_wave,1),2,n%mw
              Origin : pathcopy@ray_m
geo()%par()%lay()%Tcg
                real : curtis godson temperature for air [K]
              Origin : pathcopy@ray_m, homog_path@rayctl_m
geo()%par()%lay()%vn
                real : quotient of number density of mode 1 to total
                       nuber density for broadband-aerosol particle
                       distribution function
              Origin : pathcopy@ray_m
geo()%par()%lay()%vr()
                real : mode-radius for broadband-aerosol distribution
                       function [micrometer]
           Dimension : 2
              Origin : pathcopy@ray_m
geo()%par()%lay()%vs()
                real : mode-width for broadband-aerosol distribution
                       function
           Dimension : 2
              Origin : pathcopy@ray_m

geo()%par()%lay()%speci()%
         speci_type2 : layerspecies
```

```
          Dimension : n%speci
geo()%par()%lay()%speci()%col
               real : partial column amount [molecules/cm2]
             Origin : pathcopy@ray_m, homog_path@rayctl_m
geo()%par()%lay()%speci()%dcol()
               real : derivative of partial column amount with
                      respect to each vmr-parameter
          Dimension : para%speci()%n_vmr_para
             Origin : pathcopy@ray_m
geo()%par()%lay()%speci()%dcolgrad()
               real : derivative of partial column amount with
                      respect to each vmr-gradient parameter
          Dimension : para%speci()%n_vmrgrad_para
             Origin : pathcopy@ray_m
geo()%par()%lay()%speci()%i_dcolgrad_ne0()
            integer : dcolgrad index for dcol /= 0
          Dimension : geo()%par()%lay()%speci()%n_dcolgrad_ne0
             Origin : para_dcolgrad_ne0@rayctl_m
geo()%par()%lay()%speci()%i_dcol_ne0()
            integer : dcol index for dcol /= 0
          Dimension : geo()%par()%lay()%speci()%n_dcol_ne0
             Origin : para_dcol_ne0@rayctl_m
geo()%par()%lay()%speci()%n_dcolgrad_ne0
            integer : number of dcolgrad values which are /= 0
             Origin : para_dcolgrad_ne0@rayctl_m
geo()%par()%lay()%speci()%n_dcol_ne0
            integer : number of dcol values which are /= 0
             Origin : para_dcol_ne0@rayctl_m
geo()%par()%lay()%speci()%n_nlte_iso
            integer : number of nlte isotopes
             Origin : alloc_geo@ray_m, homog_path@rayctl_m
geo()%par()%lay()%speci()%pcg
               real : curtis godson pressure [hPa]
             Origin : pathcopy@ray_m, homog_path@rayctl_m
geo()%par()%lay()%speci()%Tcg
               real : curtis godson temperature [K]
             Origin : pathcopy@ray_m, homog_path@rayctl_m

geo()%par()%lay()%speci()%iso()%
          iso_type3 : isotope
          Dimension : speci()%n_nlte_iso
geo()%par()%lay()%speci()%iso()%i
            integer : hitran isotope number
             Origin : alloc_geo@ray_m, homog_path@rayctl_m
geo()%par()%lay()%speci()%iso()%nlte_speci
            integer : pointer to the nlte-species
             Origin : alloc_geo@ray_m, homog_path@rayctl_m
geo()%par()%lay()%speci()%iso()%nstate
            integer : number of nlte vib states
             Origin : alloc_geo@ray_m, homog_path@rayctl_m

geo()%par()%lay()%speci()%iso()%state()%
        state_type2 : band vibrational state
          Dimension : 0:nlte(speci()%iso()%nlte_speci)%nstate
```

```
geo()%par()%lay()%speci()%iso()%state()%i
             integer : HITRAN vibr. state-number
              Origin : alloc_geo@ray_m, homog_path@rayctl_m
geo()%par()%lay()%speci()%iso()%state()%i_dTvib_ne0()
             integer : index for dTvib_cg_dT /= 0
           Dimension : geo()%par()%lay()%speci()%iso()%state()%n_dTvib_ne0
              Origin : para_dTvib_ne0@rayctl_m
geo()%par()%lay()%speci()%iso()%state()%dr_dTkin
                real : derivative of r with respect to Tkin - cg
              Origin : calc_nlte_ratios@rayctl_m
geo()%par()%lay()%speci()%iso()%state()%dTvib_cg_dT()
                real : derivative of Tvib with respect to
                       each Tvib parameter
           Dimension : para%state()%n_Tvib_para
              Origin : pathcopy@ray_m
geo()%par()%lay()%speci()%iso()%state()%n_dTvib_ne0
             integer : number of dTvib_cg_dT values which are /= 0
              Origin : para_dTvib_ne0@rayctl_m
geo()%par()%lay()%speci()%iso()%state()%r
                real : ratio of populations between nlte and lte
                       case for actual state
              Origin : pathcopy@ray_m, homog_path@rayctl_m

geo()%par()%lay()%speci()%mw()%
       mw_absco_type : microwindow (for absorption coefficients)
           Dimension : n%mw
geo()%par()%lay()%speci()%mw()%l_absco_calc
             logical : =.true. if absorption coeff's in
                       mw_absco_type are calculated
                       =.false. if absorption coeff's are
                       mw_absoc type is only a pointer
                       to already calculated abscos
              Origin : allocate_geo_mw@abco_m, absco_calc@abco_m

geo()%par()%lay()%speci()%mw()%absco%
          absco_type : absorption coefficient lte
geo()%par()%lay()%speci()%mw()%absco%abco()
                real : absorption coefficients [cm2/molecule]
           Dimension : geo()%par()%lay()%speci()%mw()%absco%iabcomx
              Origin : absco_calc@abco_m
geo()%par()%lay()%speci()%mw()%absco%dabcodT()
                real : derivative of absorption coefficients
                        with respect to temperature
                        [cm2/molecule/K]
           Dimension : geo()%par()%lay()%speci()%mw()%absco%iabcomx
              Origin : absco_calc@abco_m
geo()%par()%lay()%speci()%mw()%absco%fdel
                real : wavenumber fine grid distance [cm-1]
              Origin : absco_calc@abco_m
geo()%par()%lay()%speci()%mw()%absco%fmax
                real : highest wavenumber [cm-1]
              Origin : absco_calc@abco_m
geo()%par()%lay()%speci()%mw()%absco%fmin
                real : lowest wavenumber [cm-1]
```

```
             Origin : absco_calc@abco_m
geo()%par()%lay()%speci()%mw()%absco%iabco()
            integer : index for the point in the wavenumber
                      fine grid where the absorption coefficient
                      is stored
          Dimension : geo()%par()%lay()%speci()%mw()%absco%iabcomx
             Origin : absco_calc@abco_m
geo()%par()%lay()%speci()%mw()%absco%iabcomx
            integer : number of absorption coeffients
             Origin : absco_calc@abco_m
geo()%par()%lay()%speci()%mw()%absco%Tflag
            logical : =.true. if absorption coefficient
                      T-derivative is calculated
             Origin : absco_calc@abco_m

geo()%par()%lay()%speci()%mw()%iso()%
      iso_absco_type : nlte isotope for abs. coef.
          Dimension : speci()%n_nlte_iso

geo()%par()%lay()%speci()%mw()%iso()%band()%
     band_absco_type : band for absorption coefficients
          Dimension : speci()%iso()%n_nlte_bands
geo()%par()%lay()%speci()%mw()%iso()%band()%l_absco_calc
            logical : =.true. if absorption coeff's in
                      mw_absco_type are calculated
                      =.false. if absorption coeff's are
                      mw_absoc type is only a pointer
                      to already calculated abscos
             Origin : allocate_geo_mw@abco_m, absco_calc@abco_m

geo()%par()%lay()%speci()%mw()%iso()%band()%absco%
          absco_type : absorption coefficient nlte
geo()%par()%lay()%speci()%mw()%iso()%band()%absco%abco()
               real : absorption coefficients [cm2/molecule]
          Dimension : geo()%par()%lay()%speci()%mw()%iso()%band()%absco%iabcomx
             Origin : absco_calc@abco_m
geo()%par()%lay()%speci()%mw()%iso()%band()%absco%dabcodT()
               real : derivative of absorption coefficients
                       with respect to temperature
                       [cm2/molecule/K]
          Dimension : geo()%par()%lay()%speci()%mw()%iso()%band()%absco%iabcomx
             Origin : absco_calc@abco_m
geo()%par()%lay()%speci()%mw()%iso()%band()%absco%fdel
               real : wavenumber fine grid distance [cm-1]
             Origin : absco_calc@abco_m
geo()%par()%lay()%speci()%mw()%iso()%band()%absco%fmax
               real : highest wavenumber [cm-1]
             Origin : absco_calc@abco_m
geo()%par()%lay()%speci()%mw()%iso()%band()%absco%fmin
               real : lowest wavenumber [cm-1]
             Origin : absco_calc@abco_m
geo()%par()%lay()%speci()%mw()%iso()%band()%absco%iabco()
            integer : index for the point in the wavenumber
                      fine grid where the absorption coefficient
```

```
                        is stored
           Dimension : geo()%par()%lay()%speci()%mw()%iso()%band()%absco%iabcomx
              Origin : absco_calc@abco_m
geo()%par()%lay()%speci()%mw()%iso()%band()%absco%iabcomx
             integer : number of absorption coeffients
geo()%par()%lay()%speci()%mw()%iso()%band()%absco%Tflag
             logical : =.true. if absorption coefficient
                       T-derivative is calculated
              Origin : absco_calc@abco_m
===============================================================================

inprof()%
     inprofiles_type : input profiles
           Dimension : n%inlev
inprof()%alt
                real : altitudes [km]
              Origin : input_main@input_m
inprof()%aerabs()
                real : continuum absorption coefficient [km-1]
           Dimension : n%mw
              Origin : input_contprof@input_m
inprof()%aersca()
                real : continuum scattering coefficient [km-1]
           Dimension : n%mw
              Origin : input_contprof@input_m
inprof()%lat
                real : latitude of the profile [rad]
              Origin : input_main@input_m
inprof()%lon
                real :  longitude of the profile [rad]
                        (increasing in easterly direction)
              Origin : input_main@input_m
inprof()%p
                real : pressure [hPa]
              Origin : input_pTprof@input_m
inprof()%T
                real : temperature [K]
              Origin : input_pTprof@input_m
inprof()%Tvib()
                real : vibrational temperatures [K]
           Dimension : (nlte()%nstate,n%nlte_speci)
              Origin : input_Tvibprof@input_m
inprof()%vmr()
                real : vmr [ppmv]
           Dimension : n%speci
              Origin : input_vmrprof@input_m
              Modify : isomult@input_m

inprof()%latgrad%
  ingrad_profiles_type : gradient profiles along latitude circles
                          (positive to east)
inprof()%latgrad%p
                real : pressure gradient [hPa/km]
              Origin : input_pTgradprof@input_m
```

```
inprof()%latgrad%T
                real : temperature gradient [K/km]
              Origin : input_pTgradprof@input_m
inprof()%latgrad%Tvib()
                real : vibrational temperature gradient [K/km]
           Dimension : (nlte()%nstate,n%nlte_speci)
              Origin : input_Tvibgradprof@input_m
inprof()%latgrad%vmr()
                real : vmr gradient [ppmv/km]
           Dimension : n%speci
              Origin : input_vmrgradprof@input_m
              Modify : isomult@input_m

inprof()%longrad%
  ingrad_profiles_type : gradient profiles along longitude circles
                         (positive to south)
inprof()%longrad%p
                real : pressure gradient [hPa/km]
              Origin : input_pTgradprof@input_m
inprof()%longrad%T
                real : temperature gradient [K/km]
              Origin : input_pTgradprof@input_m
inprof()%longrad%Tvib()
                real : vibrational temperature gradient [K/km]
           Dimension : (nlte()%nstate,n%nlte_speci)
              Origin : input_Tvibgradprof@input_m
inprof()%longrad%vmr()
                real : vmr gradient [ppmv/km]
           Dimension : n%speci
              Origin : input_vmrgradprof@input_m
              Modify : isomult@input_m
==============================================================================

inst%
     instrument_type : instrumental specifications
inst%ilsradius
             integer : radius of the apodisation (or ils) function
                       [integer multiples of wgrid%fine]
              Origin : input_main@input_m
inst%noise_seed
             integer : seed value for the noise:
                         >= 0: same noise in two runs
                         < 0: noise always changes
              Origin : input_main@input_m
inst%rms_sinc()
                real : rms value of the noise in the sinc-spectrum
                       (independent grid-values of distance 1/2*opdmax)
           Dimension : n%mw
              Origin : input_main@input_m

inst%gen%
   general_inst_type : ils-model parameters for general instrument
inst%gen%apolin
                real : Part of linear Apodisation
```

```
                         1.0= perfect instrument, 0.0= triangle
               Origin : input_main@input_m
inst%gen%coneifm
                 real : maximum inclination of the ray in the
                        interferometer [rad]
               Origin : input_main@input_m
inst%gen%iapokind
              integer : kind of apodisation
               Origin : input_main@input_m
inst%gen%opdmax
                 real : maximum path difference [cm]
               Origin : input_main@input_m
inst%gen%phaserr
                 real : Phase error [rad]
               Origin : input_main@input_m

inst%envi%
         envisat_type : MIPAS-envisat ils-model parameters
inst%envi%airy
                 real : infrared misalignment y-direction [rad]
               Origin : input_main@input_m
inst%envi%airz
                 real : infrared misalignment z-direction [rad]
               Origin : input_main@input_m
inst%envi%aly
                 real : laser misalignment along y [rad]
               Origin : input_main@input_m
inst%envi%alz
                 real : laser misalignment along z [rad]
               Origin : input_main@input_m
inst%envi%by
                 real : blur angular width along y [rad]
               Origin : input_main@input_m
inst%envi%bz
                 real : blur angular width along z [rad]
               Origin : input_main@input_m
inst%envi%dalphy
                 real : full interferometer divergence along y [rad]
               Origin : input_main@input_m
inst%envi%dalphz
                 real : full interferometer divergence along z [rad]
               Origin : input_main@input_m
inst%envi%deltat
                 real : mismatched delay between electronic response
                        and ADC trigger [s]
inst%envi%delxy
                 real : linear shear variation along y [-]
               Origin : input_main@input_m
inst%envi%delxz
                 real : linear shear variation along z [-]
               Origin : input_main@input_m
inst%envi%dely
                 real : retroreflector linear shear along y [cm]
               Origin : input_main@input_m
```

```
inst%envi%delz
               real : retroreflector linear shear along z [cm]
             Origin : input_main@input_m
inst%envi%dvnull
               real : initial relative speed fluctuation at
                      beginning of scan [cm/s]
             Origin : input_main@input_m
inst%envi%dxnull
               real : initial sampling pertubation [dxnull]
             Origin : input_main@input_m
inst%envi%iapokind
            integer : kind of apodisation
             Origin : input_main@input_m
inst%envi%lfnoise
               real : bandwith laser 1/f noise [Hz]
             Origin : input_main@input_m
inst%envi%lwnoise
               real : bandwith laser white noise [Hz]
             Origin : input_main@input_m
inst%envi%lwvdrift
               real : relative drift rate of laser wvnr [-]
             Origin : input_main@input_m
inst%envi%lwvnr
               real : laser wvnr [cm-1]
             Origin : input_main@input_m
inst%envi%opdmax
               real : maximum path difference [cm]
             Origin : input_main@input_m
inst%envi%pgain
               real : gain slope of IR electrical response [-]
             Origin : input_main@input_m
inst%envi%taus
               real : time constant of exponential attentuation of
                      initial sampling pertubation [s]
             Origin : input_main@input_m
inst%envi%tauv
               real : time constant of exponential attentuation of
                      initial speed fluctuation [s]
             Origin : input_main@input_m
inst%envi%vscan
               real : optical speed of interferometer [cm/s]
             Origin : input_main@input_m

inst%mw()%
        inst_mw_type : microwindow dependent instrument parameters
           Dimension : n%mw
inst%mw()%ails()
               real : read-in apodised instrumental line shape
           Dimension : inst%mw()%n_ails_pts
             Origin : input_ails@input_m
inst%mw()%n_ails_pts
            integer : number of spectral grid points of read-in ails
             Origin : input_ails@input_m
inst%mw()%off()
```

```
                real : instrumental offset per geometry [nW/cm2Srcm-1]
           Dimension : n%obsgeo
              Origin : input_main@input_m
inst%mw()%sca()
                real : instrumental scale per geometry
           Dimension : n%obsgeo
              Origin : input_main@input_m
inst%mw()%shift
                real :  wavenumber shift [cm-1]
              Origin : input_main@input_m
inst%mw()%vertconefov
                real : half vertical fov extension [rad]
              Origin : input_main@input_m
inst%mw()%vertfov()
                real : weighting function for vertical fov
           Dimension : nvertfov
              Origin : input_main@input_m
===============================================================================

modprof()%
  modelprofiles_type : model altitudes
           Dimension : n%modlev
modprof()%alt
                real : altitude of model levels [km]
              Origin : make_modelgrid@modlev_m
===============================================================================

mol()%
         hitmol_type : HITRAN molecule information
           Dimension : nlingas
mol()%abunalt()
                real : altitude levels of the isotope profiles
           Dimension : mol()%nlev
              Origin : input_isoprof@input_m
mol()%n
             integer : number of isotopes
              Origin : input_hitmol@input_m
mol()%nlev
             integer : number of levels for the isotope profiles
              Origin : input_isoprof@input_m

mol()%iso()%
           iso_type1 : information on isotopes
           Dimension : mol()%n
mol()%iso()%abunhit
                real : hitran isotope abundance
              Origin : input_hitmol@input_m
mol()%iso()%abunprof()
                real : isotope abundance profiles
           Dimension : mol()%nlev
              Origin : input_isoprof@input_m
mol()%iso()%iground
             integer : hitran number of each vibration ground state
           Dimension : mol()%iso()%nground
```

```
             Origin : input_hitmol@input_m
mol()%iso()%nground
            integer : number of vibrational ground states
             Origin : input_hitmol@input_m
mol()%iso()%nstate
            integer : number of vibr. states
             Origin : input_hitmol@input_m
mol()%iso()%qcoef1
               real : Gamache coefficients for partiton-sum
                      calculation(70-500K)
          Dimension : 4
             Origin : input_hitmol@input_m
mol()%iso()%qcoef2
               real : Gamache coefficients for partiton-sum
                      calculation(>500K)
          Dimension : 4
             Origin : input_hitmol@input_m
mol()%iso()%q296
               real : partition sums at 296 K
             Origin : input_hitmol@input_m
mol()%iso()%wmol
               real : molecular weight [g/mol]
             Origin : input_hitmol@input_m

mol()%iso()%state()%
        state_type1 : state vector
          Dimension : mol()%iso()%nstate
mol()%iso()%state()%E
               real : state energy
             Origin : input_hitmol@input_m
===============================================================================

mw()%
    microwindow_type : mw definition
          Dimension : n%mw
mw()%igasiso()
            integer : molecule/isotope identifier of each species
          Dimension : mw()%nspeci
             Origin : input_mwdef@input_m
mw()%ispeci()
            integer : general species number of each species in mw
          Dimension : mw()%nspeci
             Origin : speciorder@input_m
mw()%nsmw
            integer : number of sub-microwindows
             Origin : ini_sub_mw@inismw_m
mw()%nspeci
            integer : number of species per mw
             Origin : input_mwdef@input_m
mw()%occuobs()
            logical : occupation matrix for the observations
          Dimension : n%obsgeo
             Origin : ini_sub_mw@inismw_m
mw()%occusim()
```

```
             logical : occupation matrix for the simulations
           Dimension : n%simgeo
              Origin : make_occusim@modgeo_m
mw()%shifthitwvnr()
                real : shift vs HITRAN data for each species and microwindow
           Dimension : mw()%nspeci
              Origin : input_mwdef@input_m
mw()%sim_obs()
             logical : which simulations are necessary for which observations
           Dimension : (n%simgeo,n%obsgeo)
              Origin : make_occusim@modgeo_m
mw()%w1
                real : begin wavenumber of mw [cm-1]
              Origin : input_mwdef@input_m
mw()%w2
                real : end wavenumber of mw  [cm-1]
              Origin : input_mwdef@input_m
mw()%w1_ils
                real : begin wavenumber of ils-extended mw [cm-1]
              Origin : extend_mw@input_m
mw()%w2_ils
                real : end wavenumber of ils-extended mw [cm-1]
              Origin : extend_mw@input_m

mw()%geo()%
         mw_geo_type : geometries for external sub-mws
           Dimension : n%obsgeo
mw()%geo()%nsmw
             integer : number of external sub-mws
              Origin : input_mwdef@input_m

mw()%geo()%smw()%
      mw_geosmw_type : external sub-mws
           Dimension : mw()%geo()%nsmw
mw()%geo()%smw()%w1
                real : begin wavenumber of external sub-mw [cm-1]
              Origin : input_mwdef@input_m
mw()%geo()%smw()%w2
                real : end wavenumber of external sub-mw [cm-1]
              Origin : input_mwdef@input_m
mw()%geo()%smw()%w1_ils
                real : begin wavenumber of ils-extended external mw [cm-1]
              Origin : extend_mw@input_m
mw()%geo()%smw()%w2_ils
                real : end wavenumber of ils-extended external mw [cm-1]
              Origin : extend_mw@input_m

mw()%smw()%
         sub_mw_type : internal sub-microwindows
mw()%smw()%occuobs()
             logical : occupation matrix for the observations
           Dimension : n%obsgeo
              Origin : ini_sub_mw@inismw_m
mw()%smw()%occusim()
```

```
             logical : occupation matrix for the simulations
           Dimension : n%simgeo
              Origin : make_occusim@modgeo_m
mw()%smw()%sim_obs()
             logical : which simulations are necessary for which observations
           Dimension : (n%simgeo,n%obsgeo)
              Origin : make_occusim@modgeo_m
mw()%smw()%w1
                real : begin wavenumber of internal sub-mw [cm-1]
              Origin : ini_sub_mw@inismw_m
mw()%smw()%w2
                real : end wavenumber of internal sub-mw  [cm-1]
              Origin : ini_sub_mw@inismw_m
mw()%smw()%w1_ils
                real : begin wavenumber of ils-extended internal sub-mw [cm-1]
              Origin : ini_sub_mw@inismw_m
mw()%smw()%w2_ils
                real : end wavenumber of ils-extended internal sub-mw [cm-1]
              Origin : ini_sub_mw@inismw_m
===============================================================================

n%
        numbers_type : numbers
n%baselev
             integer : number of base levels
              Origin : input_main@input_m
n%inlev
             integer : number of input levels
              Origin : input_main@input_m
n%lspeci
             integer : number of line-data species
              Origin : speciorder@input_m
n%modlev
             integer : number of model levels
              Origin : make_modelgrid@modlev_m
n%mw
             integer : number of microwindows
              Origin : input_mwdef@input_m
n%nlte_speci
             integer : number of nlte-species
              Origin : input_main@input_m
n%obsgeo
             integer : number of observation geometries
              Origin : input_main@input_m
n%simgeo
             integer : number of simulated geometries
              Origin : make_modelgeo@modlev_m
n%speci
             integer : number of species
              Origin : speciorder@input_m
n%tot_nlte_state
             integer : total number of nlte-states in input
              Origin : input_main@input_m
n%xspeci
```

```
             integer : number of cross-section data species
              Origin : speciorder@input_m
===============================================================================

nlte()%
        non_lte_type : nlte definition
           Dimension : n%nlte_speci
nlte()%igasiso
             integer : nlte species identifier (10*mo+iso)
              Origin : input_main@input_m
nlte()%istate()
             integer : hitran vibrational state number
           Dimension : nlte()%nstate
              Origin : input_main@input_m
nlte()%itrans()
             integer : transition
           Dimension : (nlte()%ntrans,2)
              Origin : input_main@input_m
nlte()%model_speci
             integer : =1 if the species is a nlte-model species
                       =0 if not
              Origin : input_main@input_m
nlte()%nstate
             integer : number of vibrational states
              Origin : input_main@input_m
nlte()%ntrans
             integer : number of transitions for which
                       nlte should be considered
              Origin : input_main@input_m
===============================================================================

obs%
   obs_geometry_type : observation geometry
obs%alt()
                real : observer altitude
           Dimension : n%obsgeo
              Origin : input_main@input_m
obs%angle()
                real : nadir/elevation angles [rad]
           Dimension : n%obsgeo
              Origin : input_main@input_m, make_modelgeo@modgeo_m,
                       raytrace_ctrl@rayctl_m
obs%aziview()
                real : viewing azimuth angle [rad] (south=0,
                       direction S->O->N->W)
           Dimension : n%obsgeo
              Origin : input_main@input_m
obs%lat()
                real : latitudes of obverver/tangent points [rad]
           Dimension : n%obsgeo
              Origin : input_main@input_m
obs%length
                real : path length for homogeneous path
                       calculation [km]
```

```
            Origin : input_main@input_m
obs%lon()
              real : longitude of obverver/tangent points [rad]
                     (increasing in easterly direction)
         Dimension : n%obsgeo
            Origin : input_main@input_m
obs%Tback
              real : background temperature
                     (if negative: only transmission calculation)
            Origin : input_main@input_m
obs%ztang()
              real : tangent altitudes [km]
         Dimension : n%obsgeo
            Origin : input_main@input_m, raytrace_ctrl@rayctl_m
===============================================================================

outdat%
      outdata_type : output data vector
outdat%n_mw
           integer : number of microwindows
            Origin : ini_output@iniout_m
outdat%wdel
              real : wavenumber grid distance
            Origin : ini_output@iniout_m

outdat%mw()%
         outmw_type : output data for each microwindow
          Dimension : n%mw
outdat%mw()%n_geo
           integer : number geometries
            Origin : ini_output@iniout_m

outdat%mw()%geo()%
        outgeo_type : output data for each geometry
          Dimension : outdat%mw()%n_geo
outdat%mw()%geo()%n_smw
           integer : number of external sub-microwindows for each geometry
            Origin : ini_output@iniout_m

outdat%mw()%geo()%smw()%
           out_type : output data for each sub-microwindow
         Dimension : outdat%mw()%geo()%n_smw
outdat%mw()%geo()%smw()%dspec_dapo()
              real : derivative of coarse grid
                     spectrum with respect to linear apodisation
         Dimension : outdat%mw()%geo()%smw()%n_pts
            Origin : ini_output@iniout_m
outdat%mw()%geo()%smw()%dspec_dlos()
              real : derivative of coarse grid
                     spectrum with respect to line of sight
                     elevation angle
         Dimension : outdat%mw()%geo()%smw()%n_pts
            Origin : ini_output@iniout_m
outdat%mw()%geo()%smw()%dspec_doff()
```

```
                real : derivative of coarse grid
                       spectrum with respect to offset
           Dimension : outdat%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m
outdat%mw()%geo()%smw()%dspec_dpha()
                real : derivative of coarse grid
                       spectrum with respect to phase
           Dimension : outdat%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m
outdat%mw()%geo()%smw()%dspec_dsca()
                real : derivative of coarse grid
                       spectrum with respect to scale
           Dimension : outdat%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m
outdat%mw()%geo()%smw()%dspec_dshift()
                real : derivative of coarse grid
                       spectrum with respect to shift
           Dimension : outdat%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m
outdat%mw()%geo()%smw()%n_pts
             integer : number of spectral points
outdat%mw()%geo()%smw()%spec()
                real : spectrum on coarse grid
           Dimension : outdat%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m
outdat%mw()%geo()%smw()%w1
                real : first wavenumber of output spectrum
              Origin : ini_output@iniout_m

outdat%mw()%geo()%smw()%dspec_daerabs()%
          dspec_type : derivative of coarse grid
                       spectrum with respect to continuum
                       absorption coefficient
           Dimension : para%n_aerabs_para
              Origin : ini_output@iniout_m
outdat%mw()%geo()%smw()%dspec_daerabs()%spec()
                real : derivative spectra
           Dimension : outdat%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m

outdat%mw()%geo()%smw()%dspec_dbrdaeronum()%
          dspec_type : derivative of coarse grid
                       spectrum with respect to broadband aerosol
                       number density coefficients
           Dimension : para%n_brdaeronum_para
              Origin : ini_output@iniout_m
outdat%mw()%geo()%smw()%dspec_dbrdaeronum()%spec()
                real : derivative spectra
           Dimension : outdat%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m

outdat%mw()%geo()%smw()%dspec_dMie(:,:)%
          dspec_type : derivative of coarse grid
                       spectrum with respect to Mie parameters
```

```
           Dimension : brdaero%nMie,para%n_Mie_para
              Origin : ini_output@iniout_m
outdat%mw()%geo()%smw()%dspec_dbrdaeronum()%spec()
                real : derivative spectra
           Dimension : outdat%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m

outdat%mw()%geo()%smw()%dspec_dp()%
          dspec_type : derivative of coarse grid
                       spectrum with respect to pressure
           Dimension : para%n_p_para
outdat%mw()%geo()%smw()%dspec_dp()%spec()
                real : derivative spectra
           Dimension : outdat%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m

outdat%mw()%geo()%smw()%dspec_dT()%
          dspec_type : derivative of coarse grid
                       spectrum with respect to temperature
           Dimension : para%n_T_para
outdat%mw()%geo()%smw()%dspec_dT()%spec()
                real : derivative spectra
           Dimension : outdat%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m

outdat%mw()%geo()%smw()%dspec_dTgrad()%
          dspec_type : derivative of coarse grid
                       spectrum with respect to temperature-gradient
           Dimension : para%n_Tgrad_para
outdat%mw()%geo()%smw()%dspec_dTgrad()%spec()
                real : derivative spectra
           Dimension : outdat%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m

outdat%mw()%geo()%smw()%dspec_dTvib()%
          dspec_type : derivative of coarse grid
                       spectrum with respect to vibrational temperature
           Dimension : (para%n_Tvib_para_max,para%n_nlte_state)
outdat%mw()%geo()%smw()%dspec_dTvib()%spec()
                real : derivative spectra
           Dimension : outdat%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m

outdat%mw()%geo()%smw()%dspec_dvmr()%
          dspec_type : derivative of coarse grid
                       spectrum with respect to vmr
           Dimension : (para%n_vmr_para_max,para%n_vmr_speci)
outdat%mw()%geo()%smw()%dspec_dvmr()%spec()
                real : derivative spectra
           Dimension : outdat%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m

outdat%mw()%geo()%smw()%dspec_dvmrgrad()%
          dspec_type : derivative of coarse grid
```

```
                        spectrum with respect to vmr-gradient
           Dimension : (para%n_vmrgrad_para_max,para%n_vmrgrad_speci)
outdat%mw()%geo()%smw()%dspec_dvmrgrad()%spec()
                real : derivative spectra
           Dimension : outdat%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m
outdat%dangle_dztang()
                real : derivative of nadir angle wrt tangent altitude [rad/km]
           Dimension : n%obsgeo
              Origin : ini_output@iniout_m
outdat%ztang()
                real : tangent altitudes
           Dimension : n%obsgeo
              Origin : ini_output@iniout_m
===============================================================================

outheader
       character(80) : header for output files
              Origin : input_main@input_m
===============================================================================

para%
        retpara_type : retrieval parameters
para%alt_aerabs()
                real : altitudes of continuum paramete5rs
           Dimension : para%n_aerabs_para
              Origin : ini_para_aerabs@inipar_m
para%brdaeronum()
                real : parameters for broadband aerosol number density
           Dimension : para%n_brdaeronum_para
              Origin : ini_para_num@inipar_m
para%i_nlte_iso
             integer : global nlte_isotope number
                       for which Tvib derivatives are calculated
              Origin : ini_para_Tvib@inipar_m
para%i_nlte_speci
             integer : global species number for which Tvib
                       derivatives are calculated
              Origin : ini_para_Tvib@inipar_m
para%i_nlte_state()
             integer : pointer to the global nlte state number
           Dimension : para%n_nlte_state
              Origin : ini_para_Tvib@inipar_m
para%i_vmrgrad_speci()
             integer : pointer to global species numbering
           Dimension : para%n_vmrgrad_speci
              Origin : ini_para_vmrgrad@inipar_m
para%i_vmr_speci()
             integer : pointer to global species numbering
           Dimension : para%n_vmr_speci
              Origin : ini_para_vmr@inipar_m
para%Mie(:,:)
                real : Mie-parameter parameters
           Dimension : (brdaero%nMie,para%n_Mie_para)
```

```
             Origin : ini_para_Mie@inipar_m
para%n_aerabs_para
            integer : number of aerosol abs.coef. parameters
             Origin : ini_para_aerabs@inipar_m
para%n_brdaeronum_para
            integer : number of broadband aerosol number density parameters
             Origin : ini_para_num@inipar_m
para%nlte_speci
            integer : pointer to the nlte-species of nlte()%..input variable
             Origin : ini_para_Tvib@inipar_m
para%n_Mie_para
            integer : number of Mie parameters
             Origin : ini_para_Mie@inipar_m
para%n_nlte_state
            integer : number of T-vib states for which derivatives
                      should be calculated
             Origin : ini_para_Tvib@inipar_m
para%n_p_para
            integer : number of p parameters
             Origin : ini_para_p@inipar_m
para%n_Tgrad_para
            integer : number of T gradient parameters
             Origin : ini_para_Tgrad@inipar_m
para%n_T_para
            integer : number of T parameters
             Origin : ini_para_T@inipar_m
para%n_Tvib_para_max
            integer : maximum number of Tvib parameters
             Origin : ini_para_Tvib@inipar_m
para%n_vmrgrad_speci
            integer : number of species for which
                      vmr-gradient is retrieved
             Origin : ini_para_vmrgrad@inipar_m
para%n_vmrgrad_para_max
            integer : maximum number of vmr-gradient parameters
             Origin : ini_para_vmrgrad@inipar_m
para%n_vmr_para_max
            integer : maximum number of vmr parameters
             Origin : ini_para_vmr@inipar_m
para%n_vmr_speci
            integer : number of species for which vmr is retrieved
             Origin : ini_para_vmr@inipar_m
para%p()
               real : parameters for p profile
          Dimension : para%n_p_para
             Origin : ini_para_p@inipar_m
para%T()
               real : parameters for T profile
          Dimension : para%n_T_para
             Origin : ini_para_T@inipar_m
para%Tgrad()
               real : parameters for T gradient profiles
          Dimension : para%n_Tgrad_para
             Origin : ini_para_Tgrad@inipar_m
```

```
para%mw()%
     paraaerabs_type : microwindows for each of which an
                       aerosol abs.coef. profile is parametrized
           Dimension : n%mw
para%mw()%aerabs()
                real : parameters for aerosol absorption coef. profile
           Dimension : para%n_aerabs_para
              Origin : ini_para_aerabs@inipar_m

para%speci()%
        paravmr_type : species of vmr parameters (index is on
                       global species numbering !!)
           Dimension : n%speci
para%speci()%n_vmrgrad_para
             integer : number of vmr-gradient parameters
              Origin : ini_para_vmrgrad@inipar_m
para%speci()%n_vmr_para
             integer : number of vmr parameters
              Origin : ini_para_vmr@inipar_m
para%speci()%vmr()
                real : parameters for vmr profile
           Dimension : para%speci()%n_vmr_para
              Origin : ini_para_vmr@inipar_m
para%speci()%vmrgrad()
                real : parameters for vmr-gradient profile
           Dimension : para%speci()%n_vmrgrad_para
              Origin : ini_para_vmrgrad@inipar_m

para%state()%
       paranlte_type : nlte-states (index is on global state numbering
           Dimension : para%n_nlte_state
para%state()%n_Tvib_para
             integer : number of Tvib parameters
              Origin : ini_para_Tvib@inipar_m
para%state()%Tvib()
                real : parameters for Tvib profile
           Dimension : para%state()%n_Tvib_para
              Origin : ini_para_Tvib@inipar_m
===============================================================================

Sails%
        outdata_type : spectrum data vector with ails (and fov)
                       convolved spectrum and derivatives
                       with internal sub-microwindow indexing
Sails%n_mw
             integer : number of microwindows
              Origin : alloc_Sails@radtra_m
Sails%wdel
                real : wavenumber grid distance
              Origin : alloc_Sails@radtra_m

Sails%mw()%
          outmw_type : data for each microwindow
```

```
          Dimension : n%mw
Sails%mw()%n_geo
            integer : number geometries
             Origin : alloc_Sails@radtra_m

Sails%mw()%geo()%
        outgeo_type : data for each geometry
          Dimension : Sails%mw()%n_geo
Sails%mw()%geo()%n_smw
            integer : number of external sub-microwindows for each geometry
             Origin : alloc_Sails@radtra_m
Sails%mw()%geo()%l_smw()
            logical : occupation vector, which internal sub-mw
                      belongs to each geometry
          Dimension : Sails%mw()%geo()%n_smw
             Origin : alloc_Sails@radtra_m

Sails%mw()%geo()%smw()%
           out_type : output data for each sub-microwindow
          Dimension : Sails%mw()%geo()%n_smw
Sails%mw()%geo()%smw()%dspec_dapo()
               real : derivative of coarse grid
                      spectrum with respect to linear apodisation
          Dimension : Sails%mw()%smw()%n_pts
             Origin : ilsapofov_calc@radtra_m
             Modify : sca@offsca_m
Sails%mw()%geo()%smw()%dspec_dlos()
               real : derivative of coarse grid
                      spectrum with respect to line of sight
                      elevation angle
          Dimension : Sails%mw()%smw()%n_pts
             Origin : ilsapofov_calc@radtra_m
             Modify : sca@offsca_m
Sails%mw()%geo()%smw()%dspec_doff()
               real : derivative of coarse grid
                      spectrum with respect to offset
          Dimension : Sails%mw()%smw()%n_pts
             Origin : off_deri@offsca_m
Sails%mw()%geo()%smw()%dspec_dpha()
               real : derivative of coarse grid
                      spectrum with respect to phase
          Dimension : Sails%mw()%smw()%n_pts
             Origin : ilsapofov_calc@radtra_m
             Modify : sca@offsca_m
Sails%mw()%geo()%smw()%dspec_dsca()
               real : derivative of coarse grid
                      spectrum with respect to scale
          Dimension : Sails%mw()%smw()%n_pts
             Origin : sca_deri@offsca_m
Sails%mw()%geo()%smw()%dspec_dshift()
               real : derivative of coarse grid
                      spectrum with respect to shift
          Dimension : Sails%mw()%smw()%n_pts
             Origin : ilsapofov_calc@radtra_m
```

```
              Modify : sca@offsca_m
Sails%mw()%geo()%smw()%n_pts
             integer : number of spectral points
              Origin : alloc_Sails@radtra_m
Sails%mw()%geo()%smw()%spec()
                real : spectrum on coarse grid
           Dimension : Sails%mw()%smw()%n_pts
              Origin : ilsapofov_calc@radtra_m
              Modify : sca@offsca_m, off@offsca_m
Sails%mw()%geo()%smw()%w1
                real : first wavenumber of output spectrum
              Origin : alloc_Sails@radtra_m

Sails%mw()%geo()%smw()%dspec_daerabs()%
          dspec_type : derivative of coarse grid
                       spectrum with respect to continuum
                       absorption coefficient
           Dimension : para%n_aerabs_para
Sails%mw()%geo()%smw()%dspec_daerabs()%spec()
                real : derivative spectra
           Dimension : Sails%mw()%smw()%n_pts
              Origin : ilsapofov_calc@radtra_m
              Modify : sca@offsca_m

Sails%mw()%geo()%smw()%dspec_dbrdaeronum()%
          dspec_type : derivative of coarse grid
                       spectrum with respect to broadband aerosol
                       number density coefficients
           Dimension : para%n_brdaeronum_para
              Origin : ini_output@iniout_m
Sails%mw()%geo()%smw()%dspec_dbrdaeronum()%spec()
                real : derivative spectra
           Dimension : Sails%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m

Sails%mw()%geo()%smw()%dspec_dMie(:,:)%
          dspec_type : derivative of coarse grid
                       spectrum with respect to Mie parameters
           Dimension : brdaero%nMie,para%n_Mie_para
              Origin : ini_output@iniout_m
Sails%mw()%geo()%smw()%dspec_dbrdaeronum()%spec()
                real : derivative spectra
           Dimension : Sails%mw()%geo()%smw()%n_pts
              Origin : ini_output@iniout_m

Sails%mw()%geo()%smw()%dspec_dp()%
          dspec_type : derivative of coarse grid
                       spectrum with respect to pressure
           Dimension : para%n_p_para
Sails%mw()%geo()%smw()%dspec_dp()%spec()
                real : derivative spectra
           Dimension : Sails%mw()%smw()%n_pts
              Origin : ilsapofov_calc@radtra_m
              Modify : sca@offsca_m
```

```
Sails%mw()%geo()%smw()%dspec_dT()%
          dspec_type : derivative of coarse grid
                       spectrum with respect to temperature
           Dimension : para%n_T_para
Sails%mw()%geo()%smw()%dspec_dT()%spec()
                real : derivative spectra
           Dimension : Sails%mw()%smw()%n_pts
              Origin : ilsapofov_calc@radtra_m
              Modify : sca@offsca_m

Sails%mw()%geo()%smw()%dspec_dTgrad()%
          dspec_type : derivative of coarse grid
                       spectrum with respect to temperature-gradient
           Dimension : para%n_Tgrad_para
Sails%mw()%geo()%smw()%dspec_dTgrad()%spec()
                real : derivative spectra
           Dimension : Sails%mw()%smw()%n_pts
              Origin : ilsapofov_calc@radtra_m
              Modify : sca@offsca_m

Sails%mw()%geo()%smw()%dspec_dTvib()%
          dspec_type : derivative of coarse grid
                       spectrum with respect to vibrational temperature
           Dimension : (para%n_Tvib_para_max,para%n_nlte_state)
Sails%mw()%geo()%smw()%dspec_dTvib()%spec()
                real : derivative spectra
           Dimension : Sails%mw()%smw()%n_pts
              Origin : ilsapofov_calc@radtra_m
              Modify : sca@offsca_m

Sails%mw()%geo()%smw()%dspec_dvmr()%
          dspec_type : derivative of coarse grid
                       spectrum with respect to vmr
           Dimension : (para%n_vmr_para_max,para%n_vmr_speci)
Sails%mw()%geo()%smw()%dspec_dvmr()%spec()
                real : derivative spectra
           Dimension : Sails%mw()%smw()%n_pts
              Origin : ilsapofov_calc@radtra_m
              Modify : sca@offsca_m

Sails%mw()%geo()%smw()%dspec_dvmrgrad()%
          dspec_type : derivative of coarse grid
                       spectrum with respect to vmr-gradient
           Dimension : (para%n_vmrgrad_para_max,para%n_vmrgrad_speci)
Sails%mw()%geo()%smw()%dspec_dvmrgrad()%spec()
                real : derivative spectra
           Dimension : Sails%mw()%smw()%n_pts
              Origin : ilsapofov_calc@radtra_m
              Modify : sca@offsca_m
==============================================================================

sim%
   sim_geometry_type : simulated geometries
```

```
sim%alt()
                real : observer altitude
           Dimension : n%simgeo
              Origin : make_modelgeo@modgeo_m
sim%angle()
                real : nadir angles [rad]
           Dimension : n%simgeo
              Origin : make_modelgeo@modgeo_m, raytrace_ctrl@rayctl_m
sim%aziview()
                real : viewing azimuth angle [rad]
           Dimension : n%simgeo
              Origin : make_modelgeo@modgeo_m
sim%dangle_dztang()
                real : derivative of nadir angle with respect to tangent
                       altitude [rad/km]
           Dimension : n%simgeo
              Origin : dangle_dztang@rayctl_m
sim%lat()
                real : latitudes of obverver/tangent points [rad]
           Dimension : n%simgeo
              Origin : make_modelgeo@modgeo_m
sim%length
                real : path length for homogeneous path calculation [km]
              Origin : make_modelgeo@modgeo_m
sim%lobs()
             logical : =T if simulation is also an observation
           Dimension : n%simgeo
              Origin : make_modelgeo@modgeo_m
sim%lon()
                real : latitudes of obverver/tangent points [rad]
                       (increasing in easterly direction)
           Dimension : n%simgeo
              Origin : make_modelgeo@modgeo_m
sim%ztang()
                real : tangent altitudes [km]
           Dimension : n%simgeo
              Origin : make_modelgeo@modgeo_m, calc_ztang@modlev_m,
                       raytrace_ctrl@rayctl_m
===============================================================================

speci()%
          speci_type : species definition
           Dimension : n%speci
speci()%hwhm_f_max
                real : maximum air-broadened half width [cm-1]
              Origin : readlines@inspec_m
speci()%hwhm_f_min
                real : minimum air-broadened half width [cm-1]
              Origin : readlines@inspec_m
speci()%isoprof
             integer : profile identifier (hitran iso number) or
                       (0) for all isotopes not included
                       in another species
              Origin : speciorder@input_m
```

```
speci()%itotiso()
             integer : isotope numbers the species consists of
           Dimension : mxiso
              Origin : speciorder@input_m
speci()%mo
             integer : hitran molecule number
              Origin : speciorder@input_m
speci()%n_nlte_iso
             integer : number of nlte isotopes
              Origin : allocno_nlte@inspec_m, alloc_nlte@inspec_m
speci()%ntotiso
             integer : total number of isotopes the species
                       consists of
              Origin : speciorder@input_m

speci()%cross%
      x_section_type : cross-section data
speci()%cross%n
             integer : original gasnumber belonging to each of
                       the n%xspeci xsection gases
              Origin : make_x@xinput_m
speci()%cross%nxmw
             integer : number of (internal x-section) mws
              Origin : make_x@xinput_m

speci()%cross%xmw()%
            xmw_type : internal x-section microwindows
           Dimension : speci()%cross%nxmw
speci()%cross%xmw()%dw
                real : wavenumber grid point distance
              Origin : make_x@xinput_m
speci()%cross%xmw()%mw
             integer : original mw belonging to each
                       internal x-section mw
              Origin : make_x@xinput_m
speci()%cross%xmw()%mw_orbit
             integer : original mw belonging to each
                       internal x-section mw in orbit (DFD)
              Origin :
speci()%cross%xmw()%ngrid
             integer : number of wavenumber grid points
              Origin : make_x@xinput_m
speci()%cross%xmw()%npT
             integer : number of x-section p-T measurements
              Origin : make_x@xinput_m
speci()%cross%xmw()%p
                real : pressure of each x-section p-T measurement [hPa]
           Dimension : speci()%cross%xmw()%npt
              Origin : make_x@xinput_m
speci()%cross%xmw()%T
                real : temperature of each x-section p-T measurement [K]
           Dimension : speci()%cross%xmw()%npT
              Origin : make_x@xinput_m
speci()%cross%xmw()%w1
```

```
                real : begin wavenumb. of each intern. xsct. mw
              Origin : make_x@xinput_m
speci()%cross%xmw()%w2
                real : end wavenumb. of each intern. xsct. mw
              Origin : make_x@xinput_m
speci()%cross%xmw()%x(:,:)
                real : measured cross sections for
                       each wavenuber grid point
                       and each pT measurement
           Dimension : speci()%cross%xmw()%ngrid,
                       speci()%cross%xmw()%npT
              Origin : readx@xinput_m

speci()%iso()%
           iso_type2 : isotope (spectroscopic data tree)
           Dimension : 0:speci()%n_nlte_iso
speci()%iso()%i
             integer : hitran isotope number
              Origin : allocno_nlte@inspec_m, alloc_nlte@inspec_m
speci()%iso()%nlte_speci
             integer : pointer to the nlte-species of nlte()%..
                       input variable
              Origin : alloc_nlte@inspec_m
speci()%iso()%n_nlte_bands
             integer : number of nlte bands
              Origin : allocno_nlte@inspec_m, alloc_nlte@inspec_m

speci()%iso()%band()%
           band_type : band (spectroscopic data tree)
           Dimension : 0:speci()%iso()%n_nlte_bands
speci()%iso()%band()%ivl
             integer : lower state global quanta index for nlte band
              Origin : allocno_nlte@inspec_m, alloc_nlte@inspec_m
speci()%iso()%band()%ivl_state
             integer : pointer to the nlte_state for lower state
                       (=0 for ground state)
              Origin : allocno_nlte@inspec_m, alloc_nlte@inspec_m
speci()%iso()%band()%ivu
             integer : upper state global quanta index for nlte band
              Origin : allocno_nlte@inspec_m, alloc_nlte@inspec_m
speci()%iso()%band()%ivu_state
             integer : pointer to the nlte_state for upper state
              Origin : allocno_nlte@inspec_m, alloc_nlte@inspec_m
speci()%iso()%band()%n_lm_branches
             integer : number of line-mixing branches
              Origin : allocno_lm@inspec_m, alloc_lm@inspec_m

speci()%iso()%band()%branch()%
         branch_type : branch (spectroscopic data tree)
           Dimension : 0:speci()%iso()%band()%n_lm_branches
speci()%iso()%band()%branch()%b0_rot
                real : rotational moment of molecule
              Origin : (read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
```

```
speci()%iso()%band()%branch()%coefrm_f()
                real : coef. of a-T parametrisation (air)
           Dimension : (3,4)
              Origin : (read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%coefrm_s()
                real : coef. of a-T parametrisation (self)
           Dimension : (3,4)
              Origin : (read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%mw_l1()
             integer : first line for each mw
           Dimension : n%mw
              Origin : pointmw@inspec_m
speci()%iso()%band()%branch()%mw_l2()
             integer : last line for each mw
           Dimension : n%mw
              Origin : pointmw@inspec_m
speci()%iso()%band()%branch()%mw_orbit_l1()
             integer : first line for each mw in orbit
           Dimension : n%mw (for orbit DFD)
              Origin :
speci()%iso()%band()%branch()%mw_orbit_l2()
             integer : last line for each mw in orbit
           Dimension : n%mw (for orbit DFD)
              Origin :
speci()%iso()%band()%branch()%nlines
             integer : number of lines
              Origin : allocopy1@inspec_m, allocopy2@inspec_m,
                       alloc_lm@inspec_m
              Modify : delete_lmlines@inspec_m
speci()%iso()%band()%branch()%symprop_f()
                real : beta factors
           Dimension : 2
              Origin : (read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%symprop_s()
                real : beta factors
           Dimension : 2
              Origin : (read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m

speci()%iso()%band()%branch()%line()%
           line_type : line specific data
           Dimension : speci()%iso()%band()%branch()%nlines
speci()%iso()%band()%branch()%line()%elow
                real : lower state energy [cm-1]
              Origin : (readlines@inspec_m, read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%line()%expt
                real : coef. of T-dependence of air-broadened halfwidth
              Origin : (readlines@inspec_m, read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%line()%hwhm_f
```

```
                 real : air-broadened half width [cm-1]
               Origin : (readlines@inspec_m, read_lmdata@inspec_m),
                        input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%line()%hwhm_s
                 real : self-broadened half width [cm-1]
               Origin : (readlines@inspec_m, read_lmdata@inspec_m),
                        input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%line()%iso
              integer : hitran isotope number
               Origin : (readlines@inspec_m, read_lmdata@inspec_m),
                        input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%line()%ivl
              integer : lower state global quanta index
               Origin : (readlines@inspec_m, read_lmdata@inspec_m),
                        input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%line()%ivu
              integer : upper state global quanta index
               Origin : (readlines@inspec_m, read_lmdata@inspec_m),
                        input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%line()%lnsint
                 real : log(line intensity)
               Origin : (readlines@inspec_m, isoabun@inspec_m),
                        input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%line()%mo
              integer : hitran molecule number
               Origin : (readlines@inspec_m, read_lmdata@inspec_m),
                        input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%line()%pshift
                 real : pressure shift
               Origin : (readlines@inspec_m, read_lmdata@inspec_m),
                        input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%line()%ql
         character(9) : lower state local quanta
               Origin : (readlines@inspec_m, read_lmdata@inspec_m),
                        input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%line()%qu
         character(9) : upper state local quanta
               Origin : (readlines@inspec_m, read_lmdata@inspec_m),
                        input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%line()%sint
                 real : line intensity  [cm-1/(molec*cm-2)] multiplied
                        by T-idependent part for optimized calculation
                        (sint * exp( hck * elow / T0hit) /
                        (1 - exp( -hck * w / T0hit)) * q296
               Origin : (readlines@inspec_m, read_lmdata@inspec_m,
                        isoabun@inspec_m), input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%line()%w
                 real : line wavenumber [cm-1]
               Origin : (readlines@inspec_m, read_lmdata@inspec_m),
                        input_spectroscopy@inspec_m

speci()%iso()%band()%branch()%lmline()%
          lmline_type : line-mixing spect.data
            Dimension : speci()%iso()%band()%branch()%nlines
```

```
              Origin : (read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%lmline()%coefy_f()
                real : coef. of y T-parametrisation (air-broad.)
           Dimension : 4
              Origin : (read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%lmline()%coefy_s()
                real : coef. of y T-parametrisation (self-broad.)
           Dimension : 4
              Origin : (read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%lmline()%csym
        character(2) : symmetry of i and f state
              Origin : (read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%lmline()%jrot
             integer : rotational quantum number
              Origin : (read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
speci()%iso()%band()%branch()%lmline()%trans
                real : transition probability =sqrt(hitran_R)
              Origin : (read_lmdata@inspec_m),
                       input_spectroscopy@inspec_m
================================================================================

sw%
         switch_type : switches
sw%addnoise
             logical : =.true. if noise should be added to the spectrum
              Origin : input_main@input_m
sw%alloc_deri
             integer : switch for allocation grade of deri% variable
                        =0  partly allocated
                        =1  totally allocated
              Origin : kopra, kopra_forwrd@kopfwd_m
sw%alloc_geo
             integer : switch for allocation grade of geo% variable
                        =0  not allocated
                        =1  totally allocated
              Origin : kopra, kopra_forwrd@kopfwd_m
sw%alloc_modprof
             integer : switch for allocation grade of modprof% variable
                        =0  not allocated
                        =1  totally allocated
              Origin : kopra, kopra_forwrd@kopfwd_m
sw%alloc_mw
             integer : switch for allocation grade of mw% variable
                        =0  partly allocated
                        =1  totally allocated
              Origin : kopra, kopra_forwrd@kopfwd_m
sw%alloc_outdat
             integer : switch for allocation grade of outdat% variable
                        =0  not allocated
```

```
                     =1  totally allocated
              Origin : kopra, kopra_forwrd@kopfwd_m
sw%alloc_Sails
             integer : switch for allocation grade of Sails% variable
                     =0  not allocated
                     =1  totally allocated
sw%alloc_sim
             integer : switch for allocation grade of sim% variable
                     =0  not allocated
                     =1  totally allocated
sw%baselev
             integer : switch for base-levels defining the
                       layering for the forward calculation
                       0= the input-profile levels are used exclusively
                       1= the input-profile levels are used and additional
                           levels with respect to criteria 7.3-7.6 are added
                       2= the levels under $7.32 are used exclusively
                       3= the levels under $7.32 are used and additional
                           levels with respect to criteria 7.3-7.6 are added
                       4= the levels are set up automatically with respect
                           to criteria 7.3-7.6
              Origin : input_main@input_m
sw%brdaero
             logical : consideration of broadband-aerosol
              Origin : input_main@input_m
sw%deriaer
             logical : continuum derivative
              Origin : input_main@input_m
sw%deriapo
             logical : linear apodisation derivative
              Origin : input_main@input_m
sw%derilos
             logical : line of sight derivative
              Origin : input_main@input_m
sw%deriMie
             logical : Mie parameter derivative
              Origin : input_main@input_m
sw%derinum
             logical : number density of broadband-aerosol derivative
              Origin : input_main@input_m
sw%derioff
             logical : offset derivative
              Origin : input_main@input_m
sw%derip
             logical : pressure derivative
              Origin : input_main@input_m
sw%deripha
             logical : phase derivative
              Origin : input_main@input_m
sw%derisca
             logical : scale derivative
              Origin : input_main@input_m
sw%derishift
             logical : wavenumber shift derivative
```

```
             Origin : input_main@input_m
sw%deriT
            logical : temperature derivative
             Origin : input_main@input_m
sw%deriTgrad
            logical : Temperature gradient derivative
             Origin : input_main@input_m
sw%deriTvib
            logical : Tvib derivative
             Origin : input_main@input_m
sw%derivmr
            logical : vmr derivative
             Origin : input_main@input_m
sw%derivmrgrad
            logical : vmr gradient derivative
             Origin : input_main@input_m
sw%derix
            logical : nlte-model parameter derivative
             Origin : input_main@input_m
sw%firstrun
            logical : =.true. for first time run of forward-model
             Origin : kopra
sw%fov
            logical : field of view calculation
             Origin : input_main@input_m
sw%fovils
            logical : field of view effect on instrumetal profile
             Origin : input_main@input_m
sw%fq_equ1
            logical : =.true.: fq set to 1.0
             Origin : input_main@input_m
sw%horigrad
            logical : horizontal gradient calculation
             Origin : input_main@input_m
sw%horigrad_p
            logical : horizontal gradient calculation for p
             Origin : input_main@input_m
sw%horigrad_T
            logical : horizontal gradient calculation for T
             Origin : input_main@input_m
sw%horigrad_Tvib
            logical : horizontal gradient calculation for Tvib
             Origin : input_main@input_m
sw%horigrad_vmr
            logical : horizontal gradient calculation for vmr
             Origin : input_main@input_m
sw%hydrostat
            integer : hydrostatic equilibrium
                      (0=no,1=level-pressures,2=level-altitudes)
             Origin : input_main@input_m
sw%ilscalc
            integer : mode of ils calculation
                      1= circular aperture with phase and lin. apo. errror
                      2= ESA parametrization
```

```
                  3= read in for each microwindow
         Origin : input_main@input_m
sw%mainspeci
        integer : species-number of main gas
                  (the Curtis-Godson T of which will be
                  used for the Planck function)
                  >1 = C-G T of this species will be used
                  0  = no main gas: mix C_G T's of all gases
                  -1 = C-G T of air will be used
         Origin : make_mainspeci@input_m
sw%Miemod
        logical : should the Mie-model be used for broadband-aerosol
         Origin : input_main@input_m
sw%mix
        logical : line mixing
         Origin : input_main@input_m
sw%mix_ddros
        integer : direct diagonalisation (1), Rosenkranz (2)
         Origin : input_main@input_m
sw%mix_qpr
        integer : only Q- (1), Q- and PR- coupling (2)
         Origin : input_main@input_m
sw%mode_obs
        integer : mode of observation
                   1= satellite / limb / tangent altitude
                   2= satellite / limb / nadir angle and observer altitude
                   3= balloon / limb / tangent altitude and observer altitude
                   4= balloon / limb / nadir angle and observer altitude
                   5= upward / nadir angle and observer altitude
                   6= upward+limb / nadir angle and observer altitude
                   7= homogeneous path (cuvette)
         Origin : input_main@input_m
sw%new_absco
        logical : absorption coefficients should be determined new
         Origin : kopra
sw%new_modelgeo
        logical : model geometries should be determined new
         Origin : kopra
sw%new_modelgrid
        logical : model levels should be determined new
         Origin : kopra
sw%nlte
        logical : nlte calculation
         Origin : input_main@input_m
sw%nlte_model
        logical : nlte-model calculation
         Origin : input_main@input_m
sw%outcoarse
        logical : switch for output on coarse grid
         Origin : input_main@input_m
sw%outfine
        integer : switch for output on fine grid
                  0 = no output on fine grid
                  1 = output on non-equidistant fine grid
```

```
                         2 = output on equidistant fine grid
             Origin : input_main@input_m
sw%paraaer
            logical : use continuum parameters
             Origin : input_main@input_m
sw%paraapo
            logical : use linear apodisation parameters
             Origin : input_main@input_m
sw%paralos
            logical : use line of sight parameters
             Origin : input_main@input_m
sw%paraMie
            logical : use Mie model parameters
             Origin : input_main@input_m
sw%paranum  logical : use broadband-aerosol number density parameters
             Origin : input_main@input_m
sw%paraoff
            logical : use offset parameters
             Origin : input_main@input_m
sw%parap
            logical : use pressure parameters
             Origin : input_main@input_m
sw%parapha
            logical : use phase parameters
             Origin : input_main@input_m
sw%parasca
            logical : use scale parameters
             Origin : input_main@input_m
sw%parashift
            logical : use wavenumber shift parameters
             Origin : input_main@input_m
sw%paraT
            logical : use temperature parameters
             Origin : input_main@input_m
sw%paraTgrad
            logical : use Temperature gradient parameters
             Origin : input_main@input_m
sw%paraTvib
            logical : use Tvib parameters
             Origin : input_main@input_m
sw%paravmr
            logical : use vmr parameters
             Origin : input_main@input_m
sw%paravmrgrad
            logical : use vmr gradient parameters
             Origin : input_main@input_m
sw%parax
            logical : use nlte-model parameter derivatives
             Origin : input_main@input_m
sw%rs_nlte
            logical : calculate rot/spin nlte
             Origin : input_main@input_m
sw%testout
            integer : test output grade
```

```
             Origin : input_main@input_m
sw%weighting_fct
            logical : calculation of weighting functions
             Origin : input_main@input_m
===============================================================================

wgrid%
      wavegrid_type : wavenumber grid
wgrid%coarse
               real : coarse grid [cm-1]
             Origin : input_main@input_m
wgrid%fine
               real : fine grid [cm-1]
             Origin : input_main@input_m
wgrid%iratio
               real : ratio of fine and coarse grid
             Origin : input_main@input_m
===============================================================================
```