OXFORD PLANETARY GROUP

TITLE: Nemesis

'Nemesis' by Alfred Rethel (1816 – 1859)

AUTHOR:

.....
P IRWIN (5/10/09)

DOCUMENT NO.: CIRS/OX/TR/1392

DOCUMENT TITLE
Nemesis
DOCUMENT NUMBER: CIRS/OX/TR/1392

DOCUMENT STATUS SHEET			
ISSUE	REV	DATE	REASON FOR CHANGE
A	0	20/10/03	NEW
A	1	4/11/03	Revised
A	2	5/12/03	Revised to include surface emission and the retrieval of ground temperature.
A	3	10/2/04	Revised to include averaging over several observation geometries for numerical fov-averaging
A	4	20/8/04	Revised to change handling of forward errors and also to read in previous retrieval results where available
A	5	10/9/04	Updated to allow calculation of spectra in either wavelength or wavenumber space. Updated to allow scattering calculations (although with discrete calculation of Jacobian rather than implicit or 'gradient' method. Updated method of calculating gain matrix and averaging kernels
A	6	18/2/05	Major overhaul to allow for analysis if multiple field-of-view-averaged observations, such as a set of limb views.
A	7	31/3/05	Allows error propagation of previous retrievals and also allows previous retrievals to set a priori. Channel filter integration also added.
A	8	21/7/05	Updated to allow use of channel-integrated k-tables. Also modified to allow different number of spectral points at different angles.
A	9	20/11/05	Updates to allow: 1) scattering at limb, 2) retrieval of surface albedo spectra and 3) retrieval of tangent height correction for limb observations.
A	10	30/6/06	Extra LIN=3 option added to use previous retrievals in a more generalised way. Also added new continuous a priori profile definition which allows for the profile to vary with latitude.
A	11	13/10/06	Added new profile definition.
A	12	27/11/06	Updated for retrieving the pressure at a given altitude for Mars/MCS retrievals.
A	13	3/4/07	Added fractional cloud cover retrieval
A	14	5/10/09	Minor updates
A	15	7/6/10	More Apr setups
A	16	26/10/11	Addition of LBL
B	1	25/4/12	Major overhaul and stripping out of vestigial codes
B	2	30/4/15	Moderate revisions and overhaul
B	3	9/2/17	Addition of pre-tabulated LBL

DISTRIBUTION LIST					
	NAME			NAME	
1	P. Irwin	Oxford	1	C. Nixon	GSFC
1	N. Teanby	Bristol	1	G. Orton	JPL
1	J. Barstow	UCL	1	L. Fletcher	Leicester
1	R. Garland	Oxford	1	J.-L. Baudino	Oxford
1	D. Toledo	Oxford	1	S. Bauduin	Oxford
1	A. Braude	Oxford	1	J. Sinclair	JPL
1	S. Perez-Hoyos	Granada			

Table of Contents

0. Overview

This document describes the creation, properties and running of the Oxford CIRS group's new retrieval model Nemesis – **N**on-Linear Optimal **E**stimator for **M**ultivariat**E** Spectral Analy**SIS**, which was developed from the Oxcirsg code.

0.1. Reference Documents

[R1] Irwin, P.G.J, Gradient version of Oxford Radiative Transfer and retrieval Code, Oxford CIRS Technical Report: CIRS/OX/TR/1390.

[R2] Irwin, P.G.J, and S.B. Calcutt, RADTRAN, Oxford Planetary Technical Report: NIMS/OX/PGJI/SW/136.

[R3] Hanel R.A., B.J. Conrath, D.E. Jennings and R.E. Samuelson. Exploration of the Solar System by Infrared Remote Sensing: Second Edition, Cambridge University Press, 2003

[R4] Rodgers, C.D. Inverse methods for atmospheric sounding. Theory and practice. World Scientific. 2000

0.2. Defined Fonts

As in [R1], in an attempt to make this document easier to read the following fonts are used to denote different objects.

- Executable programs are underlined. e.g. CIRSdrvg
- Suites of codes in their own subdirectories are in copperplate font. e.g. RADTRANS
- Subroutine files are in courier font. e.g. `cirsradg.f`
- Variables defined within FORTRAN codes are capitalized. e.g. NCONV, IMOD

0.3. Radiance Units

NEMESIS can operate in either wavelength or wavenumber space. Nemesis (or one of its variants) can also compute 5 different types of spectra, defined by the IFORM integer:

IFORM=0 Radiance
IFORM=1 F_{plan}/F_{star} i.e. secondary transit depth
IFORM=2 A_{plan}/A_{star} i.e. primary transit depth
IFORM=3 Integrated spectral power of planet
IFORM=4 Atmospheric transmission multiplied by solar flux

The actual units used in the .spx files (and .mre files) are:
IFORM=0

- Wavelength Space: W cm^{-2} sr^{-1} μm^{-1}
- Wavenumber Space: W cm^{-2} sr^{-1} $(cm^{-1})^{-1}$
  For the .mre files, these are modified for historical reasons to μW cm^{-2} sr^{-1} μm^{-1} and nW cm^{-2} sr^{-1} $(cm^{-1})^{-1}$, respectively.

IFORM=1

- Wavelength or Wavenumber Space: Fplan/Fstar (dimensionless)

IFORM=2

- Wavelength or Wavenumber Space: 100*Aplan/Astar (dimensionless)

IFORM=3

- Wavelength Space: W μm^{-1}
- Wavenumber Space: W $(cm^{-1})^{-1}$
  (Note that these values are scaled internally by a factor of 10^{-18} to ensure numerical stability, but that the final output spectrum in the .mre file has the same units as the input .spx file.)

IFORM=4

- Wavelength Space: W cm^{-2} μm^{-1}
- Wavenumber Space: W cm^{-2} $(cm^{-1})^{-1}$

0.4. NEMESIS variants

The standard NEMESIS program is called Nemesis, but there are a number of variants, which have had to be defined due to the way they use a different layer scheme, or indeed have different output units. These variants are:

Nemesis — Standard model for modelling individual observations on a planet (IFORM = 0, 1, 3 or 4; Default is IFORM=0). If a .cel file is present the model assumes a Selective Chopper Radiometer simulation.

NemesisL — As Nemesis, but optimised to deal with limb-observing geometries. Model uses different method of combining individual layers to make the calculations faster. (IFORM = 0)

NemesisMCS — Extension of NemesisL to model MCS observations of Mars. Model uses additional FOV data to model observations and also pointing data. (IFORM = 0)

Nemesisdisc — Version of Nemesis for specifically modelling power spectra of planets or secondary transit observations. (i.e. IFORM=1 or 3; Default is IFORM=1). Uses analytical calculation of radiation into a hemisphere and so is only good for non-scattering cases.

NemesisPT — Version of Nemesis for specifically modelling the primary transit spectra of exoplanets. (IFORM = 2). Note that for this model, the functional derivatives may be calculated either implicitly or numerically through the INUMERIC flag (See section 3.2). The numeric differentiation scheme is found to better capture rate of change

of primary transit signal with temperature (through temperature's effect on the scale height), but is much, much slower.

NemesisMC As Nemesis, but uses a Monte Carlo layering and scattering scheme (IFORM=0).

1. Introduction

Multivariate retrievals from the CIRS spectra remain the main objective of the Oxford CIRS data analysis effort. In order to improve the speed of retrievals, the main CIRSRAD forward model was overhauled to generate CIRSRADG, which calculates the partial derivatives of the synthetic spectra with respect to atmospheric properties internally, instead of calculating these afterwards by calculating numerous 'perturbed' spectra and taking the difference [R1]. Hence, the main OXCIRS retrieval code was superseded by OXCIRSG, which used this 'gradient' version of the forward model to calculate the Jacobian, or **K**-matrix and was thus much faster.

Following the Cassini spacecraft Jupiter flyby it was decided to radically overhaul the Oxford CIRS retrieval code to generate a new, general purpose retrieval code which could make maximum use of the advantages of CIRSRADG and also be easily switchable between different planets and different observation geometries. The new code is called Nemesis (**N**on-Linear Optimal **E**stimator for **M**ultivariat**E** Spectral Analy**SIS**) and has the following advantages over previous codes.

1. NEMESIS allows the retrieval of continuous profiles of gas abundance, temperature and cloud. Before, gas and cloud profiles were parameterised and hence the functional derivative capability of Cirsradg was being under-utilised.
2. The layering scheme and state vector elements not hard-wired for a particular planet. Instead, the profile levels used by NEMESIS internally are set to the .ref reference profile. The variable elements are then the NVMR gas profiles in the .ref file, the temperature, any of the NCONT aerosol profiles defined, the para-H_2 fraction and the fractional cloud cover. Up to four variable profiles and one other variable (such as surface temperature, surface albedo spectrum or tangent height correction) may currently be simultaneously retrieved (five variables in all).
3. NEMESIS allows either nadir or limb observations. Previous codes were unable to simulate limb-viewing conditions.
4. NEMESIS models the additional thermal emission from the ground for planets with solid surfaces.
5. NEMESIS allows the simultaneous analysis of measurements over a range of observation angles, including combinations of near-nadir and limb views.
6. NEMESIS may perform field-of-view averaging.
7. NEMESIS may operate in either wavenumber, or wavelength space.
8. NEMESIS may also be used for scattering calculations, although in this case the functional derivatives have to be calculated numerically, and thus slowly.

9. NEMESIS may perform channel integration, either by numerically convolving a spectrum with a channel filter function, or by the use of channel-integrated k-tables calculated with Calc_fnktablec.
10. NEMESIS may now calculate spectra using either the line-by-line (LBL) method in addition to the original correlated-k approximation. Implemented by setting ILBL=1.
11. NEMESIS has been extended to be able to model primary and secondary transit spectra of exoplanets.
12. NEMESIS has been extended to be able to deal with profiles where the sum of vmrs at each level is made to add up to 1.0. This also means that the molecular weight can be calculated at each level rather than assumimg the same value at all levels.
13. NEMESIS has been extended to deal with SCR simulations if it detects the presence of a .cel file.
14. NEMESIS has been extended to incorporate pre-tabulated monochromatic gas absorption lbl-tables. The tables are generated with Calc_lbltable and are used in Nemesis by setting ILBL=2.

1.1 Why is it called Nemesis?

Nemesis is traditionally known as the Goddess of Vengeance and Retribution. Some authors connect the name with "to feel just resentment" or "righteous anger". However, the word Nemesis originally meant the distributor of fortune, whether good or bad, in due proportion to each man according to his deserts.

In Greek mythology Nemesis was the daughter of Nyx the primordial goddess of the night. Nyx was born of Chaos. She gave birth to Aether alone and Hemera, Moros, Charon, Eros and the Keres with her brother, Erebus. With Dionysus, she mothered Phthonus. Apart from Nemesis, Nyx was also mother of Momus, Thanatos, Hypnos, the Hesperides, Apate, Philotes, and Geras - the Fates.

Nemesis is said to have been as beautiful as Aphrodite and was seduced by Zeus in the form of a swan. The Goddess of Punishment, Poena, was an attendant of Nemesis.

As Nemesis/Fortuna, a conflation of the Greek deity of fate with the Roman Fortuna, she was perceived not as bringer of retribution, but as having the power of changing fortune. Hence, she was an ideal deity to make patron goddess of gladiators. It is thought that gladiators made offerings to this "goddess of fortune" before fighting in the Roman arenas. It is the "goddess of fortune" view of Nemesis, which inspired the naming of this retrieval code in her honour. It is hoped that Nemesis will bring good fortune and will considerably improve the retrieval of atmospheric properties from remotely-sensed infrared planetary spectra.

2. Running Nemesis

2.1 Input files and running NEMESIS

To run NEMESIS, you need some or all (depending on which mode of NEMESIS you are using) of the following input files (whose formats are presented in Section 3):

<runname>.inp File containing details of model run – formerly read from the standard input.

<runname>.nam File containing name of run. i.e. <runname>

<runname>.set Contains details of the scattering angles to be used and also how the atmosphere is to be split into layers.

<runname>.ref Reference atmospheric profile, containing T, P, and gas abundance profiles.

<runname>.cia File containing the name of the CIA table to be used together with some characteristics of the table.

<runname>.fla Flag file, containing a list of flag integers that used to be hard-wired into the code.

<runname>.zen Zenith angle. Code looks to see if this file is present and if so reads in the IPZEN integer which defines the level in the atmosphere at which the zenith angle of the observation is defined [0) at bottom of deepest layer; 1) at the 0km altitude level; or 2) at the very top of the atmosphere]. If the file is not present IPZEN defaults to 0.

aerosol.ref Reference aerosol profile file.

parah2.ref Reference para-H_2 fraction profile. **N.B.** This file only needs to be present if the planet under consideration is a giant planet and calculation is in wavenumbers. If calculation is in wavelengths then it is assumed that it's in the near-IR in which case there is not yet a tabulation of how CIA varies with para-H_2 fraction.

fcloud.ref Reference fractional cloud cover profile. **N.B.** This file only needs to be present if a scattering run is being undertaken.

<runname>.xsc Aerosol cross-section file (x-sections and single scattering albedos as a function of wavenumber or wavelength).

<runname>.sur Surface emissivity file (surface emissivity as a function of wavenumber or wavelength). **N.B.** This file needs to be defined only for cases where the observation is not limb, and the planet is not a giant planet. For scattering calculations (ISCAT=1 to 4) the surface albedo is set to 1-emissivity if GALB, defined in the .set file (and subsequently written to the .sca file) is set negative. Albedo is also set to 1-emissivity for Monte Carlo scattering.

<runname>.apr *A priori* set up file. Defines variables to be set and how they are represented.

<runname>.kls List of *k*-distribution table (*.kta) files to be used in calculation.

<runname>.lls List of lbl-table (*.lta) files to be used in calculation.

<runname>.spx Spectrum or spectra to be simulated and FOV averaging geometries and weights to use.

<runname>.abo Abort file, terminates retrievals early if required.

<runname>.fil Contains channel filter functions. **N.B.** This file only needs to be present if channel integrations are required. See section 3.3.

<runname>.lbl Contains the wavenumber range, step, wing, v_rel and v_cutoff of lines to be included for LBL Nemesis runs (ILBL=1). **N.B.** only necessary for LBL calculations. Note: the wavenumber range only needs to cover the output spectral range (in wavenumbers of wavelengths) and **does not** additionally need to account for v_rel.

<runname>.sha Instrument lineshape to be used in final spectral convolution for LBL Nemesis runs (ILBL=1,2). 0=square, 1=triangular, 2=Gaussian, 3=Hamming, 4=Hanning. **N.B.** only necessary for LBL calculations.

<runname>.key Line data .key file to be used for Nemesis LBL calculations. **N.B.** only necessary for LBL calculations.

<runname>.pra If present, this file lists which lineshape should be used for which gas absorption lines. **N.B.** only necessary for LBL calculations.

<runname>.sol The name of the reference solar/stellar spectral that should be used if this is a reflectivity or a transit (either primary or secondary) spectroscopy calculation.

<runname>.rfl If present, this file has details of any reflecting layer calculations to be added to the output of Nemesis or Nemesisdisc. Described in Section 3.9.

<runname>.vpf If present, this file lists the gases whose VMRs are to be limited by saturation and for each gas lists the desired limiting relative humidity and whether the volatile is arriving from the deep interior or from space. Described in Section 3.10.

<runname>.cel If present, this file lists the cell definition parameters that are to be pasted into the .pat file to simulate a Selective Chopper Radiometer (SCR) observation. The format of this section is described in the Radtrans manual. Only SCR cell types are currently supported. If Nemesis detects the presence of this file an SCR simulation is performed.

<runname>.rdw If present, this file lists the 'ranking' of each of the wavelengths in the .spx file, so that the first few iterations of the retrieval model only fits the highest-ranked wavelengths, with the rest of the wavelengths to be fit only in later iterations. This can save a substantial amount of computational time for a retrieval using a large number of wavelengths, while losing almost no information in the final state vector. Note that this scheme cannot yet be used for SCR observations.

<runname>.pre Raw fitted state vectors and covariance matrices from a previous retrieval run. These are output to <runname>.raw, which is simply renamed to <runname>.pre.

hgphase(1-n).dat NCONT files in all containing the Henyey-Greenstein phase functions for each particle type as a function of either wavenumber or wavelength. **N.B.** These files are only required for scattering calculations.

The format of some of these files is described in the next section.

The code is then run either by typing 'Nemesis' and then entering <runname>, or by typing 'Nemesis < runname.nam > test.prc &'.

N.B. There is now an additional version of **NEMESIS,** called **NEMESISL,** which has been optimised for multiple limb retrievals. Its functionality is identical to the main code, but it splits up the atmosphere more efficiently for multiple limb views. If you are modelling a single limb view, the default **NEMESIS** is faster and more accurate.

2.2 Intermediate files

Intermediate files created by NEMESIS to run the Cirsradg (or Cirsrad for a scattering calculation) routines, on which it is based, are:

<runname>.pat	Radtrans path file [R2].
<runname>.sca	Radtrans scattering file.
<runname>.str	meta file for passing previously retrieved parameters to the routine which writes the .prf files.
<runname>.prf	Radtrans atmospheric profile file
aerosol.prf	Radtrans aerosol profile file
parah2.prf	Radtrans para-H_2 profile file (if planet is a Giant Planet and calculation is in wavenumbers).
fcloud.prf	Radtrans fractional cloud cover profile file (if you are doing a scattering calculation).

Intermediate files created by Cirsradg/Cirsrad:

<runname>.drv	Radtrans driver file.

2.3 Output files and inspecting output

The final output file of NEMESIS is:

<runname>.mre	Fitted spectrum, retrieved state vector and errors for one or several measurements (defined in <runname>.spx). This file can be plotted by either the IDL routines `plotmretnewX.pro` (for individual profiles) or `scanmretnewX.pro` (for plotting several retrievals consecutively.

There is also an additional output file for version A7 onwards:

<runname>.raw	Raw fitted state vectors and covariance matrices. These are output in case the results of previous retrievals (including retrieval errors) are required in later retrievals, in which case this file is renamed as <runname>.pre

In addition, if only one profile is retrieved from the input .spx file, then a number of other output files are written for diagnostic purposes:

<runname>.itr	Full ASCII record of the state vector, fitted spectrum and **K**-matrix for each iteration of the retrieval. Can be read and plotted by the IDL routine `plotiternewX.pro`.
<runname>.cov	ASCII 'Covariance' file. Contains the **K**-matrix as well as the gain matrix **G**, averaging kernels **A** and error covariance matrices. Can be read and displayed by the IDL routine `imagecovariance.pro`.
kk.out	Unformatted binary file containing the **K**-matrix. Can be plotted with the IDL routine `plotkkimageX.pro`.

3. Input File Formats

Most of the input files required are standard Radtrans format files described by [R2]. The .ref files are basically .prf files and provide reference profiles that remain static during a NEMESIS run. The format of the temperature/pressure/vmr .prf and .ref files has recently been updated: 1) both now list the volume mixing ratios of all the gases on the **same** lines as the height, pressure and temperature, rather than in separate 6-column blocks; and 2) the .ref file can now include reference profiles for a range of different latitudes – the number of latitudes included is listed after the AMFORM parameter and the file then holds a set of .prf files at these different latitudes. We assume that all the profiles use the same AMFORM and hold the same gas vmrs.

The actual profiles used at each iteration of the forward model are generated from the .ref files and the variable profiles defined in the .apr *a priori* file and are written to intermediate .prf files. The formats of the other major input files will now be described.

3.1 *A priori* .apr file.

The .apr file format is as follows:

```
******** any header info you like. One line only ********
NVAR                ! number of variable profiles (vmr,T, or cont)
VARIDENT(1,1:3)     ! Identity of profile 1
VARPARAM(1,*)       ! Any extra parameters, or filename
VARIDENT(2,1:3)     ! Identity of profile 2
VARPARAM(2,*)       ! Any extra parameters, or filename )
Etc.
```

An example .apr file is:

```
******** any header info you like. One line only ********
2                   ! number of variable profiles (vmr,T, or cont)
11 0 1              ! Ammonia, deep, fsh
0.7                 ! pknee
```

```
2.19e-4 2.19e-5     ! deep vmr and error
0.15 0.05           ! fsh and error
0 0 0               ! Temperature - continuous
tempapr.dat
```

The top line of the file is assumed to contain header information and is skipped. The next line contains NVAR, the number of variable profiles that are to be fitted. For each variable profile, the .apr file contains VARIDENT(IVAR,1:3) which is read in next. In the case above, the VARIDENT(IVAR,1:3) of the first variable is 11, 0, 1. The first two integers describe the identity of the profile, and the third integer describes how the profile is parameterised. The profile may be gas abundance, temperature, aerosol density, para-H_2 fraction, surface temperature, surface albedo spectrum or tangent height correction, etc., depending on VARIDENT(IVAR,1) as follows. **Note that for model types with VARIDENT(IVAR,1) greater than NGAS, then VARIDENT(IVAR,3) should be set equal to VARIDENT(IVAR,1).**

1) If VARIDENT(IVAR,1) is greater than 0 then the profile is a gas volume mixing ratio, and the first two integers then contain IDGAS and ISOGAS respectively, as defined by the Radtrans manual [R2].
2) If VARIDENT(IVAR,1) is equal to zero, then the profile is a temperature profile.
3) If VARIDENT(IVAR,1) is less than zero, then the profile is either aerosol density **OR** para-H_2 fraction **OR** fractional cloud cover. Defining N as –VARIDENT(IVAR,1), if N ≤ NCONT (the number of aerosol types defined in aerosol.ref and runname.xsc) then the profile is aerosol density with ICONT = N.
4) If VARIDENT(IVAR,1) is less than zero and –VARIDENT(IVAR,1) = NCONT+1, then the profile is the para-H_2 fraction.
5) If VARIDENT(IVAR,1) is less than zero and –VARIDENT(IVAR,1) = NCONT+2 then the profile is the fractional cloud cover.
6) If VARIDENT(IVAR,1) is equal to 999 then the parameter described is the surface temperature. **Relevant only for non-giant planets.** The next line of the .apr file then contains the *a priori* surface temperature and error.
7) If VARIDENT(IVAR,1) is equal to 888 then the parameter described is a surface albedo spectrum. **Relevant only for non-giant planets.** The next line contains the number of wavelengths/wavenumbers for which the surface albedo spectrum is tabulated and the correlation length (in wavelengths). Following lines contain the wavelengths/wavenumbers and the *a priori* albedos and errors. The number of spectral points and the wavelengths/wavenumbers should agree with those defined in the accompanying .sur file.
8) If VARIDENT(IVAR,1) is equal to 889 then the parameter described is a surface albedo scaling factor. **Relevant only for non-giant planets.** The next line contains the a priori scaling factor and error.
9) If VARIDENT(IVAR,1) is equal to 887 then the parameter described is the cross-section spectrum of one of the cloud types. The next line contains the number of wavelengths/wavenumbers for which the surface albedo spectrum is tabulated, the cloud ID (1 – NCLOUD) and the correlation length (in wavelengths). Following lines contain the wavelengths/wavenumbers and the *a priori* albedos and errors. The number of spectral points and the wavelengths/wavenumbers should agree with those defined in the accompanying .xsc file.
10) If VARIDENT(IVAR,1) is equal to 777 then the parameter described is a correction to the tangent height altitude for limb observations. The next line contains the assumed tangent height correction (in km) together with the error.

11) If VARIDENT(IVAR,1) is equal to 666 then the parameter described is a retrieval of the pressure at a defined altitude used for Mars MCS limb observations. The next line contains the assumed defined altitude and the following line gives the assumed pressure together with the error.
12) If VARIDENT(IVAR,1) is equal to 555 then the parameter described is a retrieval of the planetary radius. The next line contains the assumed radius (in km) correction together with the error.
13) If VARIDENT(IVAR,1) is equal to 444 then the parameter described is a retrieval of the imaginary part of a cloud's complex refractive index spectrum. The cloud particle identifier is given by VARIDENT(IVAR,2). The next line contains the name of a separate input file, which contains the following information. Line 1 contains the mean radius of the particle size distribution and error (assumes standard size distribution), while line 2 gives the variance of the size distribution and error. Line 3 gives the number of wavelengths/wavenumbers for which imaginary refractive index spectrum is tabulated, together with the correlation length of this a priori spectrum. Line 4 gives a reference wavelength/wavenumber and the value of the real part of the refractive index at that reference. Line 5 gives the wavelength/wavenumber to which the extinction cross-section spectrum should be normalised. Following lines contain the wavelengths/wavenumbers and the *a priori* values of the imaginary refractive index spectrum and errors. In this model, the code the real part of the refractive index spectrum is calculated with a Kramers-Kronig analysis and then the Mie scattering properties of the particles calculated. **Note that the wavelengths/wavenumbers must match those in the accompanying <runname>.xsc file.**
14) If VARIDENT(IVAR,1) is equal to 445 then the parameter retrieves a cloud's imaginary refractive index using a Mie coated sphere model (Toon and Ackerman, 1981). The cloud particle identifier is given by VARIDENT(IVAR,2). The next line contains the name of a separate input file, which contains the following information. Line 1 contains the mean radius of the particle size distribution and error (assumes standard size distribution), while line 2 gives the variance of the size distribution and error. Line 3 gives the ratio of the radius of the outer 'shell' of a single cloud particle with respect to the total radius of the cloud particle, with error. Line 4 gives the number of wavelengths/wavenumbers for which the imaginary refractive index spectrum to be retrieved is tabulated, together with the correlation length of the corresponding a priori spectrum. Line 5 gives a reference wavelength/wavenumber and the real parts of the refractive indices of the core and shell respectively at that reference. Line 6 gives the wavelength/wavenumber to which the extinction cross-section spectrum should be normalised. Following lines contain the wavelengths/wavenumbers and the *a priori* values of the two imaginary refractive index spectra to be retrieved with errors (core in second and third columns, shell in fourth and fifth). In this model, the real parts of the refractive index spectra are calculated with a Kramers-Kronig analysis and then the Mie scattering properties of the particles calculated using dmiess.f. **Note that the wavelengths/wavenumbers must match those in the accompanying <runname>.xsc file.**
15) If VARIDENT(IVAR,1) is equal to 333 then the parameter described is a retrieval of the planetary surface gravity parameter: $\log_{10}(g)$, where g is units of cm s^{-2}. The next line contains the assumed value of $\log_{10}(g)$ together with the error.
16) If VARIDENT(IVAR,1) is equal to 222 then the Uranus cloud layering scheme of Sromovsky et al. (2011) is used, which specifies 5 distinct cloud layers with 5

optical depths and 3 variable base pressures leading to 8 free parameters. The three lowest cloud decks are very thin and so a specialised atmospheric layering scheme is used to line up the layers with the cloud decks. The 8 following lines contain the variable parameters and the last line contains 5 fixed parameters. Please see `readapriori.f` for more detail.

17) If VARIDENT(IVAR,1) is equal to 223 then a revised Uranus cloud layering scheme is used where a depletion of methane in the troposphere is also allowed. The format is the same as for the 222 model, except that there is an extra variable parameter, the methane fraction, and the meaning of two other parameters is slightly changed. Again, please see `readapriori.f` for more detail.
18) If VARIDENT(IVAR,1) is equal to 224 then a revised Uranus cloud layering scheme is used where the methane cloud is vertically extendable. The format is the same as for the 223 model, except that there is an extra variable parameter, the methane clouid fractional scale height. Again, please see `readapriori.f` for more detail.
19) If VARIDENT(IVAR,1) is equal to 225 then a revised Uranus cloud layering scheme is used where the methane cloud is vertically extendable and a depletion of methane in the troposphere is allowed. The format is the same as for the 224 model, except that there is yet another an extra variable parameter! Again, please see `readapriori.f` for more detail.
20) If VARIDENT(IVAR,1) is equal to 227 then the *Crème Brûlée* layering scheme of Sromovsky et al (2017)/Baines et al (2016) is used. Next lines contain the base pressure, opacity and fractional scale height of the main tropospheric cloud layer (TC), then the base pressure and opacity of the chromophore layer, (CB), then the base pressure and opacity of the stratospheric haze layer (SH), all together with errors. Finally, the last line specifies the top pressure of the CB as a fraction of the base pressure (eg. setting this line to 0.2 would make the top pressure of the CB equal to 1/5 of the CB base pressure). To use the default *Crème Brûlée* model, where the top pressure of the CB is equal to 0.9 x the CB base pressure, set this value to 0.
21) If VARIDENT(IVAR,1) is equal to 102 then we assume there are two TP profiles (defined by model 29) and we want to compute the weighted average of the spectra calculated using both profiles. Next line contains a priori fraction (0.0 – 1.0) and error. In the retrieval the fraction is limited to remain in this range.

For non-atmospheric parameters (except type 444), then VARIDENT(IVAR,2) has no meaning and is **ignored**. However, if the parameter considered is atmospheric then the third element of VARIDENT(IVAR), i.e. VARIDENT(IVAR,3), is a parameterisation code for how the profile is to be represented. There are currently 20 methods, but this may be easily extended in future. Currently valid VARIDENT(IVAR,3) codes are:

0 Profile is to be treated as continuous over the pressure range of runname.ref, the next line of the .apr file should then contain a filename, which specifies the *a priori* profile as a function of height and should have the same number of levels as the .ref file. This filename has the following format:

```
N              CLEN
P(1)   X(1)    ERR(1)
P(2)   X(2)    ERR(2)
…
P(N)   X(N)    ERR(N)
```

N must be the same as NPRO defined in runname.ref, and the pressure grid should also be identical. X(1:N) is the *a priori* profile, and ERR(1:N) the associated errors. CLEN contains the assumed correlation length of the profile (in terms of log(P)).

1 Profile is to be represented as a deep value up to a certain 'knee' pressure, and then a defined fractional scale height. The next line of the .apr file then contains the 'knee' pressure, followed by the *a priori* deep and fractional scale height values together with their estimated errors.

2 Profile is to be represented by a simple scaling of the corresponding profile runname.ref (for T, v.m.r.), aerosol.ref (for aerosol density), parah2.ref (for para-H_2 fraction) or fcloud.ref (for fractional cloud cover). The next line of the .apr file then contains the *a priori* factor and error.

3 Profile is again to be represented by a simple scaling of the corresponding profile runname.ref (for T, v.m.r.), aerosol.ref (for aerosol density), parah2.ref (for para-H_2 fraction) or fcloud.ref (for fractional cloud cover). However, in this option the *a priori* factor and error, contained in the next line of the .apr file, are first converted to log value and fractional error. This ensures that the profile can never go negative since no matter how small the log-value gets, its exponent will still be positive. At the end of the retrieval, the exponent of final log value and error are output to the .mre file.

4 Very similar to case when VARIDENT(IVAR,3) = 1 in that the profile is to be represented as a deep value up to a certain 'knee' pressure, and then a defined fractional scale height. However, in this case the knee pressure is also a variable parameter and thus must be supplied with an error estimate.

5 No longer supported.

6 Profile is a cloud profile represented by a base height, optical depth and cloud scale height. The next line of the .apr file then contains the reference altitude, followed by the *a priori* optical depth and scale height values together with their estimated errors. All quantities are taken as logs so negative fractional scale heights are not allowed.

7 Very similar to case when VARIDENT(IVAR,3) = 1 in that the profile is to be represented by value at a certain 'reference' pressure, and then a defined fractional scale height. However, in this case the profile is extended both upwards and below the reference pressure.

8 Profile is a cloud profile represented by a variable base pressure, specific density at the level and fractional scale height. The next line of the .apr file then contains the *a priori* base pressure, followed by the *a priori* abundance and fractional scale height values together with their estimated errors. All quantities are taken as logs so negative fractional scale heights are not allowed.

9 Profile is a cloud profile represented by a variable base height, optical depth and fractional scale height. The next line of the .apr file then contains the *a priori* base altitude, followed by the *a priori* abundance and fractional scale height values together with their estimated errors. All quantities are taken as logs so negative fractional scale heights are not allowed.

10 Profile is a condensing cloud. The parameterisation variables contain the deep gas vmr, the required relative humidity above the condensation level, the required optical depth of the condensed cloud and the fractional scale height of

the condensed cloud. The resulting cloud density will condense in cloud profile defined by VARPARAM(IVAR,1).

11 Condensing gas, but no associated cloud. Model requires the deep gas abundance and the desired relative humidity above the condensation level only (VARPARAM(IVAR,1))=0) or at all levels (VARPARAM(IVAR,1))=1).

12 Profile is a cloud with a specific density profile that has the shape of a Gaussian line. The profile is parameterised with a peak specific density, the pressure level of that peak and the width of distribution in units of log(pressure). The next line of the .apr file then contains the *a priori* peak specific density (i.e. particles/gram) and error, followed by the *a priori* pressure at the peak and the log-pressure width, with their respective errors. All quantities are taken as logs.

13 Profile is a cloud with a specific density profile that has the shape of a Lorentzian line. The profile is parameterised with a peak specific density, the pressure level of that peak and the width of distribution in units of log(pressure). The next line of the .apr file then contains the *a priori* peak specific density (i.e. particles/gram) and error, followed by the *a priori* pressure at the peak and the log-pressure width, with their respective errors. All quantities are taken as logs.

14 Profile is a cloud with a specific density profile that has the shape of a Gaussian line. The profile is parameterised with in integrated optical depth, the altitude where the distribution peaks and the width of distribution in units of km. The next line of the .apr file then contains the *a priori* integrated optical depth and error, followed by the *a priori* altitude where the distribution peaks and the log width in km, with their respective errors. All quantities are taken as logs, except the altitude of the peak.

15 Profile is a cloud with a specific density profile that has the shape of a Lorentzian line. The profile is parameterised with in integrated optical depth, the altitude where the distribution peaks and the width of distribution in units of km. The next line of the .apr file then contains the *a priori* integrated optical depth and error, followed by the *a priori* altitude where the distribution peaks and the log width in km, with their respective errors. All quantities are taken as logs, except the altitude of the peak.

16 Profile is specified by a value at a reference pressure together with a lapse rate (assumed positive and in units of K/km) above and below. This is of most use for temperature. It is assumed that the temperature increases above and below the reference pressure level and so the lapse rate is stored as a positive number in both instances in order that it can be held as a log number. The next line of the .apr file then contains the *a priori* nominal temperature and error at the reference tropopause pressure, followed on the next line by the *a priori* reference tropopause pressure and error. The next line gives the tropospheric lapse rate and error (i.e. the lapse rate at pressures greater than the reference pressure) while the final line gives the stratospheric lapse rate.

17 Very similar to case when VARIDENT(IVAR,3) = 7 in that the profile is represented by value at a certain ‘reference’ pressure, and then a defined fractional scale height, which extends both upwards and below the reference pressure. However, in this model the profile is limited to not increase or decrease at pressures less than a second specified pressure level (assumed to be less than the reference pressure.

18 Very similar to case when VARIDENT(IVAR,3) = 7 in that the profile is represented by value at a certain 'reference' pressure, and a defined fractional scale height. However, in this model the profile is fixed at altitudes above the reference pressure level.

19 Very similar to case when VARIDENT(IVAR,3) = 9, where a cloud profile is represented by a variable base height, optical depth and fractional scale height. However, here there is an extra parameter, which forces the profile to zero for pressures less than 0.1 atm (to simulate a tropopause 'lid'). The next line of the .apr file then contains the *a priori* base altitude, followed by the *a priori* abundance and fractional scale height values together with their estimated errors. The final line provides the cut-off parameter and error. All quantities are taken as logs so negative fractional scale heights are not allowed. The extra parameter, α, is used to provide an additional multiplicative factor to the density: $x(p) = 1.0 - exp\left(\left(log(p) - log(0.1)\right)/\alpha\right)$. NB $x(p) = 0$ for $p < 0.1$ atm.

20 Very similar to case 1 in that profile is to be represented as a deep value up to a certain 'knee' pressure, and then a defined fractional scale height. However, in this parameterisation, the profile is forced to a very small number at pressures less than a 'tropopause' temperature. The next line of the .apr file then contains the 'knee' and 'tropopause' pressures, followed by the *a priori* deep and fractional scale height values together with their estimated errors.

21 Similar to model 9. Profile is represented by a value at a variable height, rather than PRESSURE plus a fractional scale height. Below the reference height the profile is set to zero. In addition, this model scales the profile to give the requested integrated cloud optical depth. The radius is set to that assumed/fitted for the associated 444 model input. The fractional scale height is scaled according to the radius (so small particles have larger FSH).

22 Temperature profile after Robinson and Catling (2012). Input parameters:

a. tau0 optical depth at bottom of atmosphere
b. n tau depends on pressure as tau=tau0*(p/p0)^n
c. Teff Radiative equilibrium temperature (sets stratospheric temperature)
d. alpha Modification to DALR to account for SALR. In range 0.7-0.9
e. T0 Temperature at bottom of atmosphere

Temperature at top of atmosphere set to Eddington gray approximation, as this extends to lower altitudes, the lapse rate is either limited to the DALR modified by alpha or limited to lapse rate extending up for temperature T0 at P0.

23 Profile is defined by two (p,v) points, with a linear gradient (in log p) in between. The low pressure point is at (p1,v1) and the high pressure point is at (p2,v2). Profile is constant above/below this gradient region (i.e. p<p1 v=v1 and p>p2 v=v2.) All variables are retrieved. Input format:

IDGAS IDISO 23
VMR1 VMR1_ERR
P1 P1_ERR
VMR2 VMR2_ERR
P2 P2_ERR

24 Profile is represented by two separate values above and below a knee pressure (Sromovsky et al, 2017). Next lines of the .apr file respectively contain the

profile value below the knee pressure, the profile value above the knee pressure, and the value of the knee pressure itself, together with estimated errors

25 Profile is to be treated as continuous as when VARIDENT(IVAR,3) = 0, but represented by fewer points than in <runname.ref> to achieve implicit smoothing and faster retrieval times. The next line of the .apr file should then contain a filename, which specifies the *a priori* profile as a function of height using the following format:

```
NP  CLEN  NLAYER
P(1)  X(1)  ERR(1)
P(2)  X(2)  ERR(2)
…
P(NP)  X(NP) ERR(NP)
```

Where NLAYER is the number of homogeneous layers to split the profile into. This must be equal to the value of NLAYER specified in <runname.set>. If more than one VARIDENT(IVAR,3) = 25 profile present, the homogeneous layers will be split either according to a separate pressure grid specified in the file 'pressure.lay', or according to the pressure grid found in the first profile listed in <runname.apr>. Unlike in the VARIDENT(IVAR,3) = 0 case, NP does not need to be equal to NPRO (though both the maximum- and minimum pressure values from <runname.ref> do need to be present to prevent infinity errors).

26 Profile is defined by two (p,v) points, with a linear gradient (in log p) in between. The low pressure point is at (p1,v1) and the high pressure point is at (p2,v2). Profile is constant above this gradient region and zero below (i.e. p<p1 v=v1 and p>p2 v=0.) All variables are retrieved. Same input format as 23.

27 Step profile. Three lines

```
KNEE_PRESSURE        ERR
DEEP_VALUE           ERR
SHALLOW_VALUE        ERR
```

Can be used for vmr or temperature. If temperature, then deep/shallow values are handled as logs. Uses a tanh approximation to the step function in a similar way to 24, except pressure is logged to give more consistent step transition (in log(p)).

28 Profile is kept as specified in runname.ref/aerosol.ref, but at one level, where the profile is allowed to vary. The next two lines must contain:

```
LEVEL
X(LEVEL)   ERR(LEVEL)
```

Where LEVEL is then index of the point that is allowed to vary (from 1 to NPRO)

29 Profile is defined at multiple latitudes and longitudes to enable disc integration over an inhomogeneous disc. Very similar to model 0, but at multiple locations. This filename has the following format:

```
NLOCATION        N      CLEN
LAT(1) LON(1)
LAT(2) LON(2)
…
```

LAT(NLOCATION) LON(NLOCATION)
P(1) X(1) ERR(1)
P(2) X(2) ERR(2)
…
P(N) X(N) ERR(N)
…
Same blocks of P,X,ERR for remaining NLOCATION-1 profiles.

N must be the same as NPRO defined in runname.ref, and the pressure grid should also be identical. X(1:N) is the *a priori* profile, and ERR(1:N) the associated errors. CLEN contains the assumed correlation length of the profile (in terms of log(P)).

Further parameterisation schemes may be defined in the future as required. Any additional parameters (e.g. the knee pressure for VARIDENT(IVAR,3)=1,4) are held in the VARPARAM(NVAR,NPARAM) array.
Routines that need modifying are:

logflag.f
npvar.f
readapriori.f
stripvar.f
subprofretg.f

3.2 Input .inp file

This file contains specific run information and used to be read in from the standard input. The format is:

```
ISPACE, ISCAT, ILBL
WOFF
ENAME
NITER
PHILIMIT
NSPEC, IOFF
LIN
IFORM       (optional)
PERCBOOL    (optional)
```

ISPACE is the wavelength space in which to calculate the spectra and in which the k-tables are tabulated. 0 = wavenumber (cm^{-1}) and 1 = wavelength (μm). N.B. all other tabulated spectra files (e.g. '.xsc', '.sur', 'hgphase.dat' etc) should be in the wavespace specified by ISPACE.

ISCAT = 1 indicates whether a multiple scattering calculation is required. If ISCAT = 0, then a thermal emission calculation (with addition of ground radiance for non-giant

planets) is assumed. For scattering calculations the non-gradient forward model is used. (*It was originally hoped to update the 'gradient' method into the scattering code, but this turns out not to give any speed advantage*). If ISCAT = 2, then the internal scattered radiation field is calculated first (required for limb-scattering calculations). If ISCAT = 3, then a single scattering plane-parallel atmosphere calculation is performed. If ISCAT = 4, then a single scattering spherical atmosphere calculation is performed.

ILBL = 0 indicates that a correlated-K calculation is required. ILBL=1,2 indicates a line-by-line calculation. **This is an important change from previous versions**. Note that for NemesisPT, this third integer actually sets INUMERIC, which determines whether the code calculates the functional derivatives using implicit differentiation or numerically. ILBL=1 indicates a line-by-line calculation from scratch, while ILBL=2 indicates a line-by-line calculation using pre-tabulated monochromatic absorption look-up tables, calculated with the program Calc_lbltable.

WOFF is any wavenumber/wavelength calibration error, which needs to be added to the synthetic spectra.

ENAME is the name of the file which contains the forward modelling errors to be added to the measurement covariance matrix. The file starts with the number of wavelengths followed by two columns: wavenumber/wavelength and noise. This file is subsequently interpolated to required output wavelengths.

NITER is the number of iterations of the retrieval model required

PHILIMIT is the percentage convergence limit. If the percentage reduction of the cost function PHI between iterations is less than PHILIMIT then the retrieval is deemed to have converged, and the retrieval terminated early.

NSPEC is the total number of retrievals to perform (for measurements contained in the <runname.spx> file. IOFF is the index of the first spectrum to fit. For example, the <runname.spx> file may contain two sets of observations and you only want to retrieve the second, in which case, IOFF = 2, and NSPEC = 1.

LIN is an integer indicating whether the results of a previous retrieval run are to be used to set any of the model atmospheric profiles, and if so how. For example you might want to retrieve temperature first with one set of wavelengths and subsequently fit gas abundances from another set. Previous retrievals are read in from a '.pre' file (which is direct copy of the '.raw' file of the previous retrieval). The same number (IOFF) of retrievals is skipped as in the '.spx' file.

> If LIN is set to 0, then no previous retrievals are read in.
>
> If LIN is set to 1, then the previous retrievals (by copying <runname>.raw to <runname.pre>) are used both to fix the relevant variables at their last-retrieved value, and also to calculate the effect that their retrieval errors have on the current retrieval by calculating and adding this to the measurement covariance matrix $\mathbf{S}_{\varepsilon}$. Here $\mathbf{S}_x$ is the previously retrieved covariance matrix of

the constituent, and $\mathbf{K}_n$ is the Jacobian for that constituent calculated for the new wavelength array specified by the current measurement .spx file.

If LIN is set to 2, then the previous retrievals are read in, and if any variable is the same as one of the variables to be retrieved (as listed in the .apr file) then the *a priori* elements and covariance matrix are set to these last retrieved values.

If LIN is set to 3, then the previous retrievals are read in, and if any variable is the same as one of the variables to be retrieved (as listed in the .apr file) then the *a priori* elements and covariance matrix are set to these last retrieved values. In addition, all other parameters are fixed to their last-retrieved value, and their retrieval errors used to modify the measurement covariance matrix $\mathbf{S}_\varepsilon$ as per LIN=1.

IFORM defines the unit of the calculated spectrum and is defined in Section 0.3. The default value is equal to 0.

PERCBOOL defines whether the data in ENAME is given as a systematic offset in radiance values regardless on the brightness of the spectrum, or if it is given as a percentage error on the radiance of the observed spectrum. If PERCBOOL is not specified, the former is assumed, otherwise PERCBOOL must be set to *.true.*. Note that if PERCBOOL is explicitly specified in the .inp file, then IFORM cannot be omitted even if it is set to the default value.

3.3 Spectrum .spx file

This file contains the spectrum to be fitted together with FOV averaging details. It has a similar format to its .spc predecessor but includes improved FOV averaging locations and weights (which need to be generated off-line).

```
FWHM, LATITUDE, LONGITUDE, NGEOM
NCONV(1)
NAV(1)
FLAT(1,I), FLON(1,I), SOL_ ANG(1,I), EMISS_ANG(1,I), AZI_ANG(1,I),
      WGEOM(1,I)
…
FLAT(1,NAV(1)), FLON(1,NAV(1)), SOL_ ANG(1,NAV(1)),
      EMISS_ANG(1,NAV(1)), AZI_ANG(1,NAV(1)), WGEOM(1,NAV(1))
VCONV(1,1), Y(1,1), ERR(1,1)
VCONV(1,2), Y(1,2), ERR(1,2)
…
VCONV(1,NCONV), Y(1,NCONV), ERR(1,NCONV)
NCONV(2)
NAV(2)
FLAT(2,I), FLON(2,I), SOL_ ANG(2,I), EMISS_ANG(2,I), AZI_ANG(2,I),
      WGEOM(2,I)
…
FLAT(2,NAV(2)), FLON(2,NAV(2)), SOL_ ANG(2,NAV(2)),
      EMISS_ANG(2,NAV(2)), AZI_ANG(2,NAV(2)), WGEOM(2,NAV(2))
```

VCONV(2,1), Y(2,1), ERR(2,1)
VCONV(2,2), Y(2,2), ERR(2,2)
…
VCONV(2,NCONV), Y(2,NCONV), ERR(2,NCONV)
.
.
… weights, angles and spectra repeated for NGEOM spectra in total.

FWHM is the full-width half-maximum of the square box to be convolved with the CIRSRADG calculated spectrum. **N.B.** If FWHM is negative, then channel-integration is assumed and a <runname>.fil file must also be present which contains the channel filter functions. The format of this file is straightforward and is best explained by looking at the subroutine `wavesetb.f`, which reads in the filter function and then determines the wavenumbers/wavelengths in the k-tables for which the radiances need to be calculated in order to perform the channel integration.
If FWHM is set to zero, then **NEMESIS** assumes that channel-integrated k-tables have been defined and so no further convolution is required.

LATITUDE is the planetocentric latitude at the centre of the FOV

LONGITUDE is the longitude at the centre of the FOV

NGEOM is the number of different observation geometries under which the location is observed.

NCONV is the number of convolution wavenumbers/wavelengths in each spectrum. This must at present be the same for all spectra pertaining to a given location on the planet.

NAV For each of the NGEOM spectra, NAV defines how many individual spectral calculations need to be performed to construct the field-of-view-averaged spectrum.

For each viewing geometry (NGEOM in total), the parameters NCONV and NAV are first read in. NCONV is the number of convolution wavenumbers/wavelengths in each spectrum, which do not now have to be the same for all NGEOM spectra. NAV specifies how many individual spectra need to be calculated and averaged to simulate the measured field-of-view-averaged spectrum. The next NAV lines contain the integration point latitudes (FLAT), longitudes (FLON), viewing angles (SOL_ANG, EMISS_ANG, AZI_ANG) and weights (WGEOM). The angle definitions are outlined in Figure 1. Following this, there then follows the actual measured wavelengths/wavenumbers, spectrum and errors (all of length NCONV) which are read in and put in total measurement vector **y**. The measurement covariance matrix is here assumed to be diagonal, with variances equal to ERR^2. If simulating a Selective Chopper Radiometer (SCR) then for each spectrum, the first NCONV/2 rows list the sideband radiances and the second NCONV/2 rows list the wideband radiances. The wavenumbers/wavelengths of these two blocks should match.

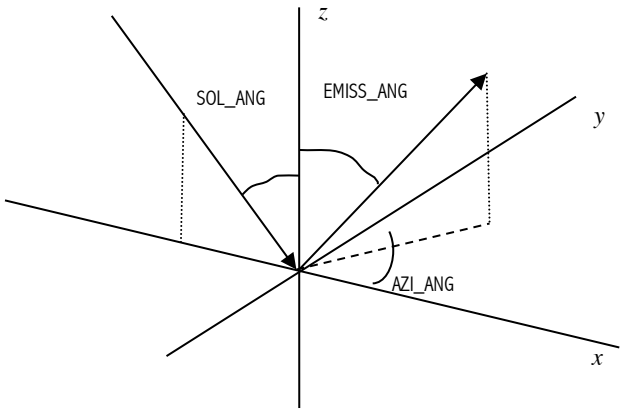

Figure 1. Definition of Viewing Angles used by **NEMESIS** and **RADTRANS**

When reading in the viewing angles, a negative emission zenith angle (EMISS_ANG) indicates that the observed spectrum is actually a limb observation where the solar zenith angle (SOL_ANG) then contains the tangent altitude (km). The definition of these angles is explained in Fig. 1. N.B. Setting AZI_ANG=0 implies FORWARD scattering.

3.4 Setup .set file

This file contains scattering quadrature information (if a scattering run is being performed) and layering information. A typical example is:

```
**********************************************************
Number of zenith angles :  5
 0.165278957666387       0.327539761183898
 0.477924949810444       0.292042683679684
 0.738773865105505       0.224889342063117
 0.919533908166459       0.133305990851069
  1.00000000000000       2.22222222222222D-002
Number of fourier components :  0
Number of azimuth angles for fourier analysis : 100
Sunlight on(1) or off(0) :  1
Distance from Sun (AU) :   5.200
Lower boundary cond. Thermal(0) Lambert(1) :  1
Ground albedo :   0.000
Surface temperature: 150.0
**********************************************************
Alt. at base of bot.layer (not limb) :  -40.000
Number of atm layers : 100
Layer type :  1
Layer integration :  1
**********************************************************
```

The meaning of the fields should be fairly clear. The first half of the file contains setup information for a scattering run, which can now be performed with NEMESIS, although the Jacobian has to be calculated numerically. The second half contains information on how the atmosphere is to be split into layers by `subpathg.f` with the layering codes equal to LAYHT, NLAYER, LAYTYP and LAYINT respectively and defined in the Radtran manual [R2]. LAYHT is used as set in the .set file UNLESS a limb-observing geometry is indicated by the .spx spectral observation file. In this special case LAYHT is set to SOL_ ANG (section 3.3). Note that if GALB < 0, then the albedo is set to 1.0 minus the emissivity defined in the emissivity (.sur) file.

3.5 Fractional cloud cover file fcloud.prf format

If aerosols are defined then RADTRANS (and thus NEMESIS) needs to know if the cloud is in the form of a uniform thin haze or is instead arranged in thicker clouds covering a certain fraction of the mean area. These details are supplied in the file fcloud.prf file.

The first line contains the number of profile levels and the number of cloud particle types (which should match that defined in the .xsc file and dust profile file). The following lines then contain the profile heights, fractional cloud cover and identifiers as to which cloud particle types contribute to the fractional cloud. Hence, the first few lines of the fcloud.prf file appear as:

NPRO, NCONT
HEIGHT(1), FRAC(1), ICLOUD(1,1), ICLOUD(2,1),…,ICLOUD(NCONT,1)
HEIGHT(2), FRAC(2), ICLOUD(1,2), ICLOUD(2,2),…,ICLOUD(NCONT,2)
…
HEIGHT(NPRO), FRAC(NPRO),ICLOUD(1,NPRO),ICLOUD(2,NPRO),…,
ICLOUD(NCONT,NPRO)

If ICLOUD(I,J) is set to 1, then aerosol type I contributes to the broken cloud at level J, which has a fractional cloud cover of FRAC(J). If ICLOUD(I,J) is set to 0, then aerosol I is treated as being part of a uniform haze.

3.6 Reference Solar/Stellar Spectrum .sol File.

This file (<runname>.sol) contains the name of the solar or stellar spectrum file, which is assumed to reside in the raddata/ directory.
The solar/stellar spectral file format is as follows. The file can contain as many header lines as necessary, each line beginning with a '#' character.
The first line after the header contains the wavenumber/wavelength space of the spectrum: 0 = wavenumber, 1 = wavelength.
The next line contains the radius of the Sun/star in units of km.
The rest of the file contains the wavelength/wavenumbers and spectral luminosity in two columns. Wavelengths/wavenumbers are in units of μm, or cm^{-1}. Spectral luminosity is in units of W μm, or W $(cm^{-1})^{-1}$.

3.7 Collision induced absorption .cia file.

This file (<runname.cia>) contains the name of the CIA file to be used (assumed to exist in the raddata directory of radtrancode.
The first line contains the name of the CIA file. The CIA file is always in wavenumber space.
The second line defines the wavenumber step, dnu, of the CIA table.
The third line gives the number of para-H_2 fractions listed, NPARA.
CIA tables are in two formats, one which lists H_2-H_2 (eqm), H_2-He(eqm), H_2-H_2 (normal), H_2-He (normal) and then 5 other pairs: H_2-N_2, H_2-CH_4, N_2-N_2, CH_4-CH_4 and H_2-CH_4. The other type of CIA table lists H_2-H_2 and H_2-He only, but for a number of different ortho/para fractions. For the usual CIA table format, NPARA should be set to zero. For a variable para-H_2 CIA table, NPARA can be set to be between 0 and 24. The exact number depends on the table itself.

3.8 Additional flags .fla file.

This file (<runname.fla>) contains the following integer flags that used to be hardwared in different parts of the code:

INORMAL	0 or 1 depending on whether the ortho/para-H_2 ratio is in equilibrium (0) or normal 3:1 (1).
IRAY	Sets the Rayleigh optical depth calculation: 0 = Rayleigh scattering optical depth is not included. 1 = Rayleigh optical depths suitable for gas giant atmosphere 2 = Rayleigh optical depths suitable for CO_2 dominated atmosphere >2 = Rayleigh optical depths suitable for a N_2-O_2 atmosphere.
IH2O	Turns additional H_2O continuum off (0) or on (1)
ICH4	Turns additional CH_4 continuum off (0) or on (1)
IO3	Turns additional O_3 UV continuum off (0) or on (1)
INH3	Turns additional NH_3 continuum off (0) or on (1)
IPTF	Used in only a few routines to switch between normal partition function calculation (0) or the high-temperature partition function for CH_4 for Hot Jupiters.
IMIE	Only relevant for scattering calculations. If set to 0, the phase function is computed from the associated Henyey-Greenstein hgphase*.dat files. However, if set to 1, the phase function is computed from the Mie-Theory calculated PHASEN.DAT file.

3.9 Additional reflecting atmosphere calculation definition .rfl file.

This file (<runname.rfl>), if present, contains the following lines:

1) Header (any)
2) Incident solar angle for calculation
3) Reflected zenith angle
4) Layer number of reflecting layer
5) Reflecting layer albedo

3.10 Additional vapour saturation definition .vpf file.

This file (<runname.vpf>), if present, lists the gases whose VMRs are to be limited by saturation and for each gas lists the desired limiting relative humidity and whether the

volatile is arriving from the deep interior or from space. If the file is not present, then the vapour pressure of all gases is left untouched. Individual gas SVP curves are listed in the raddata/SVP.dat reference data file..

The first line of the file gives the number of gases, NVP, whose abundances are to be limited by condensation. There then flow NVP lines, each listing:

ID, ISO, VP, SVPFLAG

where ID and ISO are the identifiers of the gas concerned, VP is the limiting relative humidity required (normally between 0.0 and 1.0), and SVPFLAG is a control flag integer to govern the modification behaviour. SVPFLAG may take one of four values:

0 = ignore gas on this line (same as not having a line for this gas).
1 = apply the SVP-limited value at all levels.
2 = assume an interior source and disallow local minima allowed, i.e. the gas can only decrease with altitude
3 = assume an external source, but allow local minima for pressures less than 0.05 atm. The gas VMR can only decrease with decreasing altitude deeper than 0.05 atm. The value of 0.05 atm is what Nick Teanby using for Titan, but it should be OK for the giant planets too as any photochemical weirdness is usually higher up.

At the moment, these flags do not cause any additional factors to be applied to XMAP, but this is something that we might want to think about depending on how well it works for different applications.

N.B. Technically XMAP should be set to 0.0 for gases whose VMRS are being limited by such condensation. However, we have found that this hard limit leads to undesirable retrieval behaviour such as: 1) if a gas VMR drops just below condensation in one iteration it can never return; and 2) it can lead to a sharp edge and erratic retrieval behavior. Therefore, to solve this XMAP is instead scaled by svp/pp. This also has a steep drop-off, but gives a more gentle response and more desirable retrieval behaviour.

3.11 Reduced wavelength scheme (.rdw) file

This file allows a Nemesis retrieval to be speeded up by giving each wavelength in a spectrum a 'ranking' (IRANK), and thereby allowing Nemesis to only retrieve a select sample of wavelengths in the first few iterations before moving on to the full wavelength grid. The procedure is as follows:

- In ITER=0, select only the wavelengths of IRANK=1 to perform the forward model (forwardnogX – B)
- Perform iterations with these IRANK=1 wavelengths until Nemesis converges to a solution as per the standard Nemesis convergence criteria
- Once convergence is achieved, add on the IRANK=2 wavelengths to the IRANK=1 wavelengths (forwardnogX – D), and perform further iterations in the same fashion.
- Repeat the same procedure until all wavelengths of IRANK<=MAXIRANK are included. Once convergence is achieved with wavelengths of IRANK<=MAXIRANK, the retrieval ends.
- If convergence is still not achieved after NITER-2 iterations, wavelengths of

IRANK<=MAXIRANK are added on automatically for the final 2 iterations.

The format of a typical .rdw file is as follows:

```
#Lines of comment preceded by a hash (or 'pound' if you're American) sign.
#Leave a blank line below the last comment

NCONV
ISPEC
MAXIRANK
ICONV(1)      IRANK(1)
ICONV(2)      IRANK(2)
…
ICONV(NCONV)      IRANK(NCONV)
```

Note that the largest value of IRANK does not necessarily have to be smaller than MAXIRANK. It's just that any wavelengths with IRANK>MAXIRANK will be left out of the retrieval. This allows the user to easily remove any wavelengths out of the retrieval temporarily, without having to delete rows from the .spx file.

4. Differences between NEMESIS and previous CIRS retrieval code

The main differences in running Nemesis compared to previous retrieval codes Oxcirs and Oxcirsg are:

1. The .apr *a priori* file has completely changed. See section 3.1.
2. The hard limits previously used by Oxcirs and Oxcirsg have been removed. This is because they can cause instabilities in the retrieval model if the trial state vector runs into a hard stop.
3. NEMESIS converts variable gas abundances and aerosol profiles to logs prior to retrieval to prevent negative answers and also to make the code more stable.
4. NEMESIS can retrieve the para-H_2 profile.
5. NEMESIS can retrieve the surface temperature.
6. NEMESIS can simulate spectra averaged over a range of observation geometries.
7. The contents of the .inp file are now read through a Fortran input file rather than from the standard input.
8. Under OXCIRSG, there were three executables in all: Oxcirsoneg, Oxcirsretg and Oxmultiretg. Under NEMESIS, it was decided to have one executable only. Single-shot, single and multiple retrievals are now all handled by Nemesis.
9. The use of the correlation length in defining the *a priori* covariance matrix has been corrected in the light of numerical instabilities. See section 8.
10. The code now works in either wavenumber or wavelength space, and incorporates scattering (albeit in a non-gradient form).
11. The code can now deal with multiple field-of-view-averaged observations.
12. The code now allows calculated of filter-averaged radiances, either by numerical convolution of a calculated spectrum, or by using channel-integrated *k*-tables generated by Calc_ktablec.

13. The code can now also retrieve a surface albedo spectrum.
14. The code can now also retrieve a correction to the tangent height altitudes form limb observations.
15. The code can now deal with scattering under limb observations using the plane-parallel parameterisation of Barney Conrath and Mike Smith at GSFC.
16. NEMESIS can retrieve the fractional cloud cover profile.
17. NEMESIS can now do LBL calculations (from scratch with ILBL=1, and from pretabulated tables with ILBL=2).
18. NEMESIS can now model Selective Chopper Radiometers.

4.1 Running Nemesis in LBL mode

NEMESIS now offers a LBL mode if ILBL is set to 1 in the .inp file. The set ups are identical to a normal correlated-k run, but additional input files are required to run NEMESIS in this mode:

1. A <runname.lbl> file is required. The first line of this file contains the wavenumber range (VMIN,VMAX) required to calculate the radiance over the wavelength/wavenumber range specified in the <runname>.spx file, together with the required wavenumber step for the LBL calculation. NEMESIS will calculate the LBL spectrum over this range with this step and then convolve with the instrument function specified by the <runname>.spx and <runname.sha> files. The code does not use an adaptive integrator and so the user needs to ensure that the calculation is performed at sufficient precision to model accurately the absorption features. The second line contains the LBL parameters: WING, VREL and VCUTOFF. These are described more fully in the accompanying Radtrans manual, but in summary the explicit line shape is used to calculate the contribution of a particular line to wavenumbers within WING. For larger distances from a line, it is assumed that we are into the Lorentz tail and so the contribution can be calculated at a much coarser resolution. VREL allows lines outside of VMIN,VMAX to contribute such that all the lines in the range VMIN-VREL to VMAX+VREL are included. Finally, VCUTOFF limits the contribution of any line past the specified distance from the line centres. This simulates the sub-Lorentzain behaviour of real lines. It is usual to set VREL=VCUTOFF. NB VREL will automatically be used to extend the wavenumber range defined by VMIN, VMAX. There is no need to pre-add/subtract VREL to the requested wavenumber range.
2. A <runname.sha> file is required to tell NEMESIS what instrument lineshape is required. The file contains a single line with a single integer: 0=square, 1=triangular, 2=Gaussian, 3=Hamming, 4=Hanning.
3. A <runname.key> file is required which is a line data .key file as used for Radtrans LBL calculations, which specifies which line database is to be used and which gas information files. The contents of this file are described in the Radtrans manual.
4. Optionally a <runname.pra> can be provided. If present, this file lists which line shape should be used for particular gases. If the file is absent then the Voigt lineshape is used for all gases. If present, the file contains one row for each gas to be modified containing 'ID ISO IPROC', where ID, ISO are Radtrans ID/Isotope numbers of the gas to be modified and IPROC is required lineshape. Allowed values of IPROC are listed in the Radtrans manual.

4.2 Running Nemesis in pretabulated LBL mode.

NEMESIS now offers a pre-tabulated LBL mode if ILBL is set to 2 in the .inp file. The set ups are identical to a normal correlated-k run, but additional input files are required to run NEMESIS in this mode:

1. A <runname.sha> file is required to tell NEMESIS what instrument lineshape is required. The file contains a single line with a single integer: 0=square, 1=triangular, 2=Gaussian, 3=Hamming, 4=Hanning.
2. A <runname.lls> file is required, which fulfills the same function as the <runname.kls> file for a correlated-k calculation. The <runname.kls> lists the pre-tabulated monochromatic gas absorption lbl-tables (pre-tabulated with Calc_lbltable, rather than Calc_fnktablec as for k-tables), which like .kta files are in direct-access binary format and have a .lta file extension.

5. Location of code and example input files

NEMESIS source code is now under github management to maintain all versions of the software that might be released. The github central repository of RADTRANS (and NEMESIS) is https://gitlab.physics.ox.ac.uk/planetary/radtrancode (please contact P. Irwin for details on how to access the files). Please see the Radtrans manual, section 1.1, for further details.

6. Recent Developments

Since the last version of NEMESIS, the following notable changes have been made:

1. The code can now do LBL calculations.
2. The code can deal with profiles where the sum of vmrs adds up to 1 and so the molecular weight can be calculated at each level.

7. Future developments

NEMESIS is intended to be the main retrieval tool of the Oxford Planetary Data Analysis group for future missions and is designed to be general purpose and extendable. Future upgrades that are under consideration and may/may not be implemented in the near future are:

1. Update to allow 'gradient' or implicit differentiation to determine Jacobian for scattering retrievals (hard). This has already been attempted, but the resulting code was actually slower than the numerical differentiation scheme. To implement this in a way that gains any advantage may require some clever and elegant reprogramming.

8. Significant offshoots

The overarching intention of NEMESIS is to provide a single retrieval code that can be applied to any planet and in which improvements and debugs made in one research application are then available to researchers analysing different data. While every attempt has been made to adhere to this goal, some cases have arisen where it has proven necessary to form an offshoot. These versions are different from the central NEMESIS version on the way they set up the spectral calculation and the way the

underlying radiative transfer calculations are combined to give the final result. Specifically the different offshoots generate different .pat files, which in turn generates different .drv files and the outputting path calculations are recombined differently. The current offshoots are as follows.

8.1 NemesisL

NemesisL is specifically designed for limb calculations. The atmosphere is split into the same number of layers, NLAYER, as before, but only once and from the specified lowest altitude. Limb paths are then calculated through these layers with 2 × NLAYER layers included in the lowest path and 2 layers in the top path.

There are now multiple options for running this code and a modified set of input parameters for the .inp file. The .inp file is otherwise identical to that needed for NEMESIS, but the initial line is now:

ISPACE, ISCAT, OCCULT, ILBL, INUM

where ISPACE, ISCAT and ILBL are as defined elsewhere, and OCCULT and INUM are as defined below.

The transmission and/or thermal emission for these tangent paths are calculated once and then interpolated to the actual tangent altitude required. The OCCULT parameter controls what sort of limb calculation is required. The options are:

OCCULT = 0. Thermal emission calculation only
OCCULT = 1. Solar irradiance*transmission + thermal emission
OCCULT = 2. Transmission calculation only
OCCULT = 3. (Solar irradiance*transmission + thermal emission)/
(Solar irradiance)

For most cases the Jacobeans can be calculated analytically, for which INUM=0. However, when trying to retrieve temperature and pressure together, the effects of varying each changes the atmospheric profile in a complicated and non-linear way. Hence, these Jacobeans have to be calculated by numerical differencing, set by INUM=1.

For a single limb calculation, you should use Nemesis, but for multiple tangent heights through the same atmosphere, NemesisL is much faster and not significantly less accurate.

8.2 NemesisMCS

NemesisMCS is based on NemesisL, but is specifically tailored to model MCS radiances, which require a complicated FOV combination of the individual path calculations. Other modifications are made in the way the spx files are read in, and implicitly use small differences in the wavenumbers listed to identify different detectors, which have slightly different spectral and FOV responses.

8.3 Nemesisdisc

Nemesisdisc sets up the .pat file for a disc-averaged calculation and outputs the results in a form compatible with modelling the secondary transits of exoplanets.

8.4 NemesisPT

NemesisPT is again based on NemesisL, but instead uses the limb path calculations to estimate the effective planetary radius at different wavelengths. This code is thus used for modelling the primary transits of exoplanets.

8.5 CIRSdrv_wave

CIRSdrv_wave is a very useful program for running the nemesis/cirsrad subroutines on cases that are not covered by the Nemesis programs, which write their own .pat files to model specific geometries. CIRSdrv_wave can be run in command line, but you're bombarded with quite a few questions and so I often run it as:

```
CIRSdrv_wave < cirsdrv_wave.inp > test.prc &
```

A typical cirsdrv_wave.inp file would look like this:

```
zero                (root name of the .pat, .fla, .kls and .cia files)
1                   (iwave: 0=wavenumber, 1=wavelength)
y                   (Do you want the wavelengths extracted from the .pat file?
3                   (IPLANET, 3=Earth, 4=Mars, etc.)
1                   (1=Fast, 2=slow. Historical – always choose '1')
0.35 0.005          (minimum wavelength/wavenumber and step of k-tables used)
11                  (This defines which scloudNNwave.f model to use, if
                    scattering)
200.                (Surface temperature)
```

The same inputs are always required even when they're not actually necessary! Hence, you need to provide a number for scloudNNwave and surface temperature even for calculations where these numbers are irrelevant. The basic advantage of CIRSdrv_wave is that it can run on ANY .pat file and so can test situations outside of the usual Nemesis-like calculations. For example, you can use it to calculate transmission weighting functions. The output of CIRSdrv_wave is an ASCII file containing the computed spectra for each path defined in the .pat file. Two spectra are output per path, one which is the raw spectrum calculated at the k-table wavelengths/wavenumbers spanning the requested range and the second is this spectrum convolved with a box of width FWHM (defined in the .pat file) and output at the wavelengths/wavenumbers specified in the .pat file.

8.6 Lbldrv_wave

Lbldrv_wave is exactly the same as CIRSdrv_wave except that it uses the new nemesis/cirsrad LBL subroutines. The input script is essentially identical, except that there is now no need to define the k-table minimum wavelength/wavenumber and step. However, the user does need to provide the additional .key, .lbl and .sha files necessary for a nemesis-LBL run. The output is also an ASCII file, but here there is one spectrum per defined path, smoothed with the FWHM defined in the .pat file and with the instrument function defined in the .sha file.

9. Notes

9.1 Matrix inversion instability

Debugging **NEMESIS** it was noticed that the cost function, calculated by `calc_phiret.f` was sometimes returning negative numbers. Closer investigation revealed that the problem arose from inverting the *a priori* covariance matrix, which because of the way the correlation length was used contained loads of off-diagonal elements which were almost zero but not quite. This led to instabilities in calculating the inverse of the matrix which could not be rectified even by going to double precision. **NEMESIS** now sets off-diagonal elements using the usual correlation-length formula used by Oxcirs and Oxcirsg except that off-diagonal elements smaller than a certain prescribed factor are now set to zero. This modification makes the matrix inversion stable and leads to sensible cost function values. The retrievals of **NEMESIS** now converge smoothly. The *a priori* covariance matrix is then inverted in double precision with a Cholesky decomposition routine that checks to ensure the resultant inverse actually works (i.e. $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$). The inverse is then stored to speed up resultant routines.

N.B. The same 'feature' is probably also present in Oxcirs and Oxcirsg and probably explains why during the retrievals, the Marquardt-Levenburg parameter ALAMBDA sometimes reduces nicely until the `calc_phiret.f` deduces that the solution starts getting further from the optimal fit. ALAMBDA then increases at every iteration until the maximum iteration number is reached. This behaviour can be explained if `calc_phiret.f` is calculating slightly the wrong value of the cost function and thus while the retrieval routine is trying to get closer to the solution, the cost function routine thinks it is getting further away! This means that in some previous retrievals, the presented solutions were probably not quite converged if this behaviour was present.

9.2 Constraints and 'exact' solutions

During retrieval tests of PH_3 and NH_3 profiles in Saturn's atmosphere it became clear that the retrieved solution was weighted too strongly by the measurements and tended to the 'exact' solution. In this case the correlation length set in the *a priori* profiles has very little effect and the solution becomes 'wiggly' and unattractive. I found it initially rather difficult to judge how to better constrain the solution and how to test if it was constrained!

To investigate this I went back to look at the Barney Conrath approach (see chapter 8 in [R3]) and found many similarities with optimal estimation, but perhaps a more realistic way of considering the constraints. In the optimal estimation approach used, the solution (in the non-linear case) is:

$$\mathbf{x}_{n+1} = \mathbf{x}_0 + \mathbf{S}_x\mathbf{K}_n^{\mathrm{T}}\left(\mathbf{K}_n\mathbf{S}_x\mathbf{K}_n^{\mathrm{T}} + \mathbf{S}_\varepsilon\right)^{-1}\left(\mathbf{y}_m - \mathbf{y}_n - \mathbf{K}_n\left(\mathbf{x}_0 - \mathbf{x}_n\right)\right) \qquad (8.1)$$

where $\mathbf{S}_x$ and $\mathbf{S}_\varepsilon$ are the *a priori* and measurement covariance matrices respectively and $\mathbf{K}_n$ is the matrix of functional derivatives, or Jacobian.

In Conrath's retrieval method (as I understand it!) the formalism is slightly different:

$$\mathbf{x}_{n+1} = \mathbf{x}_0 + \hat{\mathbf{S}}_x \mathbf{K}_n^{\mathrm{T}} \left(\mathbf{K}_n \hat{\mathbf{S}}_x \mathbf{K}_n^{\mathrm{T}} + \gamma \mathbf{S}_\varepsilon \right)^{-1} \left(\mathbf{y}_m - \mathbf{y}_n - \mathbf{K}_n \left(\mathbf{x}_0 - \mathbf{x}_n \right) \right) \tag{8.2}$$

where $\hat{\mathbf{S}}_x$ is now the *a priori* correlation matrix, $\mathbf{S}_\varepsilon$ and $\mathbf{K}_n$ are as before, and γ is an adjustable parameter to fine-tune the balance between measurement and *a priori* constraint.

Clearly the two formalisms are extremely similar with, I think, the optimal estimation approach more flexible in that different elements of the measurement vector can have different *a priori* constraints whereas they are all the same in Conrath's approach (if their correlation matrix is what I think it is, i.e. $\hat{S}_{ij} = S_{ij} / \left(S_{ii} \times S_{jj} \right)^{1/2}$.

I find it very difficult to decode some of the work of Clive Rodgers [R4], excellent though it is. In Rodgers (2000) there are pages and pages discussing constraints, contribution functions, averaging kernels and error propagations without ever (as far as I can tell) arriving at a simple way of judging whether there is enough constraint in the retrieval or whether it tends to the 'exact' and thus unsmoothed and unreliable one.

I have now thought of a simple test! In Eq. 8.1 if the measurement errors are tiny, the contribution function, or gain matrix, $\mathbf{G}_n$ tends to:

$$\mathbf{G}_n = \mathbf{S}_x \mathbf{K}_n^{\mathrm{T}} \left(\mathbf{K}_n \mathbf{S}_x \mathbf{K}_n^{\mathrm{T}} + \mathbf{S}_\varepsilon \right)^{-1} \rightarrow \mathbf{S}_x \mathbf{K}_n^{\mathrm{T}} \left(\mathbf{K}_n \mathbf{S}_x \mathbf{K}_n^{\mathrm{T}} \right)^{-1}$$

and the solution is exact. If however the measurement errors are huge then the contribution function $\mathbf{G}_n$ tends to:

$$\mathbf{G}_n = \mathbf{S}_x \mathbf{K}_n^{\mathrm{T}} \left(\mathbf{K}_n \mathbf{S}_x \mathbf{K}_n^{\mathrm{T}} + \mathbf{S}_\varepsilon \right)^{-1} \rightarrow \mathbf{S}_x \mathbf{K}_n^{\mathrm{T}} \mathbf{S}_\varepsilon^{-1}$$

and too much constraint is applied.

The happy medium seems to me to be when the diagonal elements of $\mathbf{K}_n \mathbf{S}_x \mathbf{K}_n^{\mathrm{T}}$ are of a similar magnitude to the diagonal elements of $\mathbf{S}_\varepsilon$ (which is usually diagonal anyway). This is now tested for in the code and reported. Errors in either the *a priori,* or the measurement covariance matrices can then be adjusted to ensure sufficient constraint and this sufficient vertical smoothing. This may be achieved by modifying the extra forward modelling error file (section 3.2) and is very much akin to modifying the γ trade-off parameter in Conrath's approach. Alternatively, by reducing the *a priori* errors or increasing the correlation length, the magnitude of $\mathbf{K}_n \mathbf{S}_x \mathbf{K}_n^{\mathrm{T}}$ can be reduced leading to a similar result.

There are of course other ways of assessing the retrieval stability and the IDL code `imagecovariance.pro` which reads the <runname>.cov file (section 2.3) displays every diagnostic plot imaginable which should help to investigate any retrieval problems.

Optimal estimation was developed for terrestrial retrievals where there is good statistical knowledge of the expected state vectors. This is NOT the case for planetary

work where what our method (and that of Barney Conrath's) actually does is to extract a smoothed representation of a real continuous profile from a limited set of data. For a reliable retrieval we require that the level of smoothing is sufficient to damp the ripples that appear in exact retrievals. If the *a priori* errors are large, then the retrieval is exact and although the retrieved errors are small, the profile is 'wiggly'. As the *a priori* errors are reduced, more and more smoothing is applied and so the retrieval errors reduce further but the retrieved profile is smoother. Another test for sufficient constraint may be to compare the measument error $\mathbf{S}_m$ and the smoothing error $\mathbf{S}_n$. These would appear to have to be of similar size at the peak of the weighting functions when the retrieval is 'balanced'. This is still being investigated.

9.3 Converting NEWCPHASE retrievals to NEMESIS

Previous NIMS retrievals used Ncretrieve, the executable program of the NEWCPHASE suite. With all the developments and advantages of NEMESIS it was decided to modify NEMESIS to do scattering retrievals in wavelength space. To do NIMS retrievals with NEMESIS, a number of steps have to be completed:

1. The measurement vector file (.syn) has to be converted to a NEMESIS .spx file. This may be done with the IDL code `convertsynspec.pro`.
2. The cloud *a priori* state vector (previously optical depth per layer) has to be converted to an equivalent aerosol.ref file. This may be done with the IDL code `convertcloud.pro`.
3. The NIMS .kta files need to be concatenated together to provide single .kta files for each gas where each file covers identical wavelength ranges. This may be done with the program Concat_nimsk, which is under CVS control in radtran/nimsrad.
4. All .xsc and hgphase*.dat files need to be converted to the same wavelength space as the k-tables and .spx file.

9.4 Optimisation of retrieval code.

The equation we solve in these retrievals is:

$$\mathbf{x}_{n+1} = \mathbf{x}_0 + \mathbf{S}_x\mathbf{K}_n^{\mathrm{T}}\left(\mathbf{K}_n\mathbf{S}_x\mathbf{K}_n^{\mathrm{T}} + \mathbf{S}_\varepsilon\right)^{-1}\left(\mathbf{y}_m - \mathbf{y}_n - \mathbf{K}_n\left(\mathbf{x}_0 - \mathbf{x}_n\right)\right) \tag{8.3}$$

or equivalently

$$\mathbf{x}_{n+1} = \mathbf{x}_0 + \mathbf{G}_n\left(\mathbf{y}_m - \mathbf{y}_n\right) - \mathbf{A}_n\left(\mathbf{x}_0 - \mathbf{x}_n\right) \tag{8.4}$$

where $\mathbf{G}_n$ is the gain matrix and $\mathbf{A}_n$ (= $\mathbf{G}_n\mathbf{K}_n$) is the averaging kernel matrix. Previous versions of the code calculated the gain matrix as:

$$\mathbf{G}_n = \mathbf{S}_x\mathbf{K}_n^{\mathrm{T}}\left(\mathbf{K}_n\mathbf{S}_x\mathbf{K}_n^{\mathrm{T}} + \mathbf{S}_\varepsilon\right)^{-1} \tag{8.5}$$

which for cases where the length m of the measurement vector was large, tended to be extremely slow. However an equivalent formulation of the gain matrix is:

$$\mathbf{G}_n = \left(\mathbf{K}_n^{\mathrm{T}}\mathbf{S}_\varepsilon^{-1}\mathbf{K}_n + \mathbf{S}_x^{-1}\right)^{-1}\mathbf{K}_n^{\mathrm{T}}\mathbf{S}_\varepsilon^{-1} \tag{8.6}$$

Since $\mathbf{S}_\varepsilon$ is assumed to be diagonal, and since $\mathbf{S}_x^{-1}$ is pre-computed in order to work out the cost function, this formulation is much faster to calculate for $n < m$.

The gain matrix (and the averaging kernel matrix) are now calculated by the subroutine `calc_gain_matrix.f`, and the calculation of Eq. 8.4 done by `calcnextxn.f`. These two routines replace the old `dretrieve.f` subroutine. Having the gain matrix and averaging kernel matrix available also makes the calculation of the final errors in `calc_serr.f` much faster too! One final improvement is that Cholesky, rather than LU decomposition, is now used for the matrix inversion, which is an optimised code for inverting positive-definite matrices, such as covariance matrices.

In version A7 onwards, the results of previous retrievals may be included (together with error) by incorporating them, appropriately, into $\mathbf{S}_{\varepsilon}$.

9.5 Adding New Parameterisations

If you want to add a new profile parameterisation to NEMESIS, then there are numerous steps to complete:

1. You need to add another parameterisation definition in `readapriori.f`. In `readapriori.f` you need to decide what new parameterisation scheme you want and which of the state vector elements need to be held internally as log values and which are held as linear values. For example, we always hold volume mixing ratios as log values so that the actual vmrs that get written to the .prf file can never become negative. For each variable type, there are three variables initially read in: VARIDENT(IVAR,1-3), which defines what sort of profile this is. This is all defined in section 3.1, but in summary:
    a. If VARIDENT(IVAR,1)=0, then this is a temperature profile
    b. If VARIDENT(IVAR,1) > 0 and < 100, then this defines a gas volume mixing ratio profile, in which case IDGAS = VARIDENT(IVAR,1) and ISOGAS = VARIDENT(IVAR,2).
    c. If VARIDENT(IVAR,1) is negative then the profile defines either an aerosol abundance profile, a para-H2 fraction profile, or fractional cloud cover. See section 3.1 for details.
    d. If VARIDENT(IVAR,1) < 100, then VARIDENT(IVAR,3) defines the profile type. If you have a new profile parameterisation in mind, then please pick the next unused number.
    e. If VARIDENT(IVAR,1) > 100, then this defines a 'special' parameterisation that might use a special layering technique, or define the cloud imaginary refractive index spectrum, etc. In such cases, please set VARIDENT(IVAR,3) = VARIDENT(IVAR,1), and set VARIDENT(IVAR,2) = 0.
2. You also need to update `logflag.f` to correctly identify which of the new state vector elements are held as log values.
3. If you are defining a new temperature, gas volume mixing ratio, cloud, para-H_2 or fractional cloud cover profile parameterisation you also need to update `npvar.f`, which defines how many elements in the state vector, NP, are necessary to define the new parameterisation.
4. If you are defining a 'special' parameterisation scheme then you will have a considerably more painful time as you need to define NP (the number of

variables associated with each parameterisation in the state vector) individually in the following files: `check_iteration.f`, `coreret.f`, `coreretdisc.f`, `coreretMC.f`, `extractsrom.f`, `extractsromch4.f`, `generateMCSspx.f`, `generatespx.f`, `gsetrad.f`, `modxvecMCMC.f`, `modxcevMCMCA.f`, `readraw.f`, `rmodapriori.f`, `scankkx.f`, `setifix.f`, `writeout.f`. This is not very elegant and should probably be improved. However, you will hopefully just be making a new temperature, gas volume mixing ratio, cloud, para-H_2 or fractional cloud cover profile parameterisation and so will not need to worry about this step!

5. You need to edit `subprofretg.f` to use the new x-vector elements to modify either the <runname>.prf, aerosol.prf, parah2.prf, or fcloud.prf files. You also need to define the internal gradients, which is a little trickier. For each profile parameterisation, `subprofretg.f` takes the NP state vector elements and constructs the profile as you require, but you also need to define the relative elements of the XMAP(MAXV,MAXGAS+2+MAXCON,MAXPRO) array. XMAP is a functional derivatives matrix and elements of XMAP are the rate of change of the profile vectors (i.e. temperature, vmr, etc.) with respect to the rate of change in the state vector elements. So, for example, if X1(J) is the modified temperature, vmr, clouds at level J to be written out to runname.prf or aerosol.prf then XMAP(K,L,J) is d(X1(J))/d(XN(K)), where L is the internal variable identifier which identifies what kind of profile this is and lies in the range (1 to NGAS+1+2*NCONT). The different values of L are:
    a. If L is in the range (1,NGAS), then we're defining a gas volume mixing ratio.
    b. If L = NGAS+1, then this is a temperature profile.
    c. If L is in the range (NGAS+2,NGAS+2+2*NCONT), then this is an aerosol profile, para-H2 fraction or fractional cloud cover profile.

   You need to think carefully about how the gradients are calculated. You also need to think about whether the individual elements in the state vector are held as linear or log values. Obviously if X1(J) = XN(K), then d(X1(J))/d(XN(K)) = 1, but if X1(J) = EXP(XN(K)), then d(X1(J))/d(XN(K)) = EXP(XN(K)). The best thing to do is find an existing profile parameterization that most closely matches what you want to do, study how it and the XMAP elements are defined and then define a new profile type based on this and modify what you need to change. That's what I always do!

10. K-table location

To facilitate future research, an attempt has been made to co-locate k-tables generated for various projects by the Nemesis modelling community. This k-tables are not currently part of the Nemesis/Radtrans distribution, but for Oxford users, the k-tables may be found in: /network/group/aopp/planetary/PGJI001_IRWIN_LBLKTAB.

11. Note on k-table formats

By default it is assumed that k-tables are compiled with square averaging bins of width FWHM, with bin centres spaced by DELV, starting at VMIN. Normally,

DELV=FWHM/2 to achieve Nyquist-sampling. If DELV > 0, then **NEMESIS** works which wavelengths/wavenumbers in the table cover the range of interest defined in the <runname>.spx file and then convolves the output with a second box of full-width-half-maximum, FWHM1, defined in the <runname>.spx file. Usually, FWHM1 = FWHM and so the resulting convolution of two square boxes gives a triangular instrument lineshape (ILS) with the same full-width-half-maximum. If FWHM1 is set to zero, then the output spectrum is simply interpolated to the required wavelengths/wavenumbers and has a square ILS of full-width-half-maximum equal to that of the k-table, i.e. FWHM.

If DELV > 0 and FWHM=0 in the k-table, then it is assumed that a non-square lineshape has been used to compile the k-table over a regular grid of wavelengths/wavenumbers. This will not normally require further convolution and so FWHM1 should be set to zero in the <runname>.spx file and the output is interpolated to the required wavelengths/wavenumbers.

IF DELV < 0 then the k-table has been calculated for a non-regular set of central wavelengths/wavenumbers, with, usually, non-square lineshapes that have a FWHM that is different for every central wavelength or wavenumber in the table. In other words, the k-table describes a set of filter-averaged channels. In such tables, FWHM is set by convention to be equal to zero. **NEMESIS** will compare the requested wavelengths in the <runname>.spx file, with those available in the k-table and 'snap' to the nearest available entry to run the calculation. No further interpolation or smoothing is applied. This method is the fastest calculation as only one entry in the k-table is used for each output wavelength/wavenumber in the <runname>.spx file and no further convolution is applied. It is possible for such tables to be regularly gridded and have the same ILS at all wavelengths.