COMPOSITE INFRARED SPECTROMETER

TITLE: Nemesis

AUTHOR:

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

DOCUMENT NO.:   CIRS/OX/TR/1392

COMPOSITE INFRARED SPECTROMETER
Nemesis

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

COMPOSITE INFRARED SPECTROMETER
Nemesis

DISTRIBUTION LIST					
	NAME			NAME	
1	P Irwin	Oxford	1	C Nixon	GSFC
1	N Teanby	Bristol	1	R de Kok	SRON
1	J. Hurley	Oxford	1	C Merlet	Oxford
1	D. Tice	Oxford	1	L. Fletcher	Oxford

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. Radiance units of the .spx files, and the units used within NEMESIS, are (for the default Nemesis program):

- Wavelength Space: $W\ cm^{-2}\ sr^{-1}\ \mu m^{-1}$
- Wavenumber Space: $W\ cm^{-2}\ sr^{-1}\ (cm^{-1})^{-1}$

For the .mre files, these are modified to $\mu W\ cm^{-2}\ sr^{-1}\ \mu m^{-1}$ and $nW\ cm^{-2}\ sr^{-1}\ (cm^{-1})^{-1}$, respectively.

0.4. NEMESIS variants

The standard Nemesis program is called <u>Nemesis</u>, 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:

<u>Nemesis</u>   Standard model for modelling individual observations on a planet.

<u>NemesisL</u>   As <u>Nemesis</u>, but optimised to deal with limb-observing geometries. Model uses different method of combining individual layers to make the calculations faster.

<u>NemesisMCS</u> Extension of <u>NemesisL</u> to model MCS observations of Mars. Model uses additional FOV data to model observations and also pointing data.

<u>Nemesisdisc</u>   Version of NEMESIS for specifically modelling disc-averaged spectra of planets or secondary transit observations.
If a solar reference file is detected, then the code computes the planet/star flux ratio for a secondary transit.
If no solar reference file is detected, then the code computes the surface spectral irradiance of the planet in units of $W\ cm^{-2}\ \mu m^{-1}$ or $W\ cm^{-2}\ (cm^{-1})^{-1}$.

<u>NemesisPT</u>   Version of NEMESIS for specifically modelling the primary transit spectra of exoplanets.
Output units are 100*planet_area/stellar_area.

<u>NemesisMC</u>   As <u>Nemesis</u>, but uses a Monte Carlo layering and scattering scheme.

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 <u>continuous</u> 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 <u>Calc_fnktablec</u>.
10. NEMESIS may now calculate spectra using either the line-by-line (LBL) method in addition to the original correlated-k approximation.

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.

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.

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 multiple scattering calculations (ISCAT=1) 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 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. **N.B.** only necessary for LBL calculations.

<runname.sha>   Instrument lineshape to be used in final spectral convolution for LBL Nemesis runs. 0=square, 1=triangular, 2=Gaussian. **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.

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 <u>by</u> 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 `plotmretnew.pro` (for individual profiles) or `scanmretnew.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 `plotiter_new.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 `plotkkimage.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 which remain static during a NEMESIS run. 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 depending on VARIDENT(IVAR,1) as follows:

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. 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 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.

10) 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.

11) If VARIDENT(IVAR,1) is equal to 555 then the parameter described is a retrieval of the planetary radius.

For non-atmospheric parameters, 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 ten 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 <u>pressure</u>, 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 <u>height</u>, 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.

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.

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

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 old non-gradient forward model is used. Eventually the 'gradient' method will be built into the scattering code. If ISCAT = 2, then the internal scattered radiation field is calculated first (required for limb-scattering calculations). If ISCAT is anything other than 0, 1 or 2 then a single scattering calculation is performed.

ILBL = 0 indicates that a correlated-K calculation is required. ILBL=1 indicates a line-by-line calculation. T**his is an important change from previous versions**.

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 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 $\mathbf{S}_\epsilon$. 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 $\mathbf{S}_\epsilon$ as per LIN=1.

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 <u>negative</u>, 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.

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
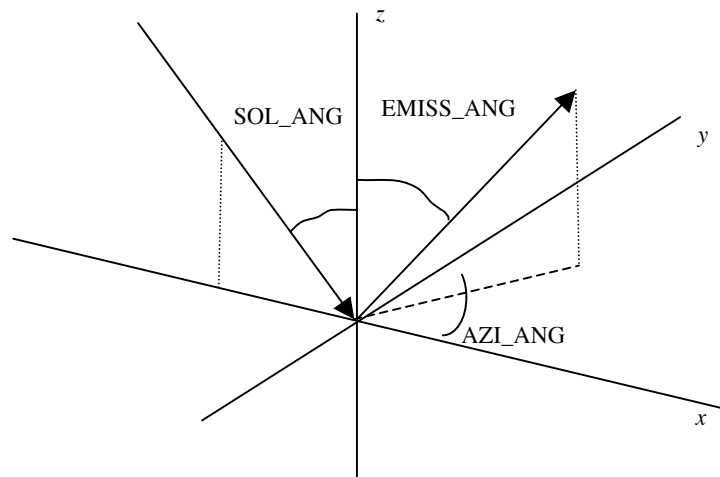is explained in Fig. 1. N.B. Setting AZI_ANG=0 implies FORWARD scattering.

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

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)
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.

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 huimidity 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.

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.

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.

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. 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 'process 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.

5. Location of code and example input files

NEMESIS source code is now under SVN management to maintain all versions of the software that might be released. The SVN central repository of RADTRANS (and

NEMESIS) is http://scm.physics.ox.ac.uk/svn/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. The transmissions through these layers are then calculated once and then interpolated to the actual tangent altitude required. For a single limb calculation, you should used 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.

8.3 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.

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 <u>almost</u> zero but <u>not quite</u>. 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 <u>now set to zero</u>. 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 which 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:

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

where $\hat{\mathbf{S}}_x$ is now the *a priori* <u>correlation</u> 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 <u>NOT</u> 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 <u>Ncretrieve</u>, 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 <u>Concat_nimsk</u>, 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:
$$\mathbf{x}_{n+1} = \mathbf{x}_0 + \mathbf{S}_x \mathbf{K}_n^{\mathrm{T}} \left(\mathbf{K}_n \mathbf{S}_x \mathbf{K}_n^{\mathrm{T}} + \mathbf{S}_\varepsilon \right)^{-1} \left(\mathbf{y}_m - \mathbf{y}_n - \mathbf{K}_n \left(\mathbf{x}_0 - \mathbf{x}_n \right) \right) \qquad (8.3)$$
or equivalently
$$\mathbf{x}_{n+1} = \mathbf{x}_0 + \mathbf{G}_n \left(\mathbf{y}_m - \mathbf{y}_n \right) - \mathbf{A}_n \left(\mathbf{x}_0 - \mathbf{x}_n \right) \qquad (8.4)$$
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:
$$\mathbf{G}_n = \mathbf{S}_x \mathbf{K}_n^{\mathrm{T}} \left(\mathbf{K}_n \mathbf{S}_x \mathbf{K}_n^{\mathrm{T}} + \mathbf{S}_\varepsilon \right)^{-1} \qquad (8.5)$$
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:

$$G_n = \left(K_n^T S_\varepsilon^{-1} K_n + S_x^{-1} \right)^{-1} K_n^T S_\varepsilon^{-1} \qquad (8.6)$$

Since S_ε is assumed to be diagonal, and since 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 S_x.

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: /home/jupiter/plan2/specdata.