Technical report; TR-255

KNMI

Documentation of the software package GomeCal

Jos van Geffen

De Bilt, 2003

Technical Report = Technisch Rapport; TR-255

De Bilt, 2003

PO Box 201, 3730 AE De Bilt
The Netherlands
Wilhelminalaan 10
http://www.knmi.nl
Telephone +31 30 22 06 911
Telefax +31 30 22 10 407

Author: Geffen, J.van

UDC: 551.501.71
551.510.534
681.3.06

ISSN: 0169-1708

ISBN: 90-369-2239-9

KNMI

De Bilt, The Netherlands
Technical report TR-255

Documentation of the software package GomeCal

http://www.knmi.nl/gome_fd/gomecal/

Version: 1.0 -- 9 September 2003

Author: Jos van Geffen

Table of contents

GomeCal documentation -- version 1.0

GomeCal

a spectral recalibration tool
for GOME / GDP level-1 spectra

Introduction and contents

The documentation of the GomeCal software package starts with a brief introduction, followed by the contents of the documentation and some note on the notation used on these pages.

The software package is distributed via the World Wide Web and can be accessed through the GomeCal home page (http://www.knmi.nl/gome_fd/gomecal/). There you can also find the latest version of this documentation on-line as well as in PostScript and PDF format. Version 1.0 of the documentation has been published as a Technical Report by KNMI, De Bilt, Netherlands.

Introduction

The Global Ozone Monitoring Experiment (GOME) is a nadir-viewing ultraviolet and visible spectrometer aboard the ERS-2 (European Remote Sensing) satellite launched by the European Space Agency (ESA) in April 1995. GOME measures the spectra of sunlight scattered in the atmosphere and reflected by the surface of the Earth in the wavelength range 240 to 790 nm, and once a day a direct solar spectrum is measured.

Solar and earthshine spectra, the so-called level-1 data, are the official product release of the GDP (GOME Data Processor) of the ESA Processing and Archiving Facility at DLR (Deutsches Zentrum für Luft- und Raumfahrt) in Oberpfaffenhofen, Germany. These data are released on CD-ROMs and via the Internet.

GomeCal is a spectral recalibration tool for GOME/GDP level-1 spectra. The main program of the GomeCal software package is called **gomecal**. That program can perform several calibration corrections on GOME/GDP level-1 spectra extracted from the GDP-made CD-ROMs, where the user can choose which correction steps are to be performed.

The recalibration steps **gomecal** can perform are:

> a *wavelength calibration*, which improves upon the calibration of the GDP

> a *polarisation correction*, which fully replaces the polarisation correction from the GDP
> a *radiometric correction*, which consists of three parts:
- a radiometric calibration
- a correction for the degradation of the instrument
- a correction to remove residual effects of the interference of the Peltier cooler signals

in addition to the corrections of the GDP.

In principle **gomecal** is a stand-alone program which is able to recalibrate the level-1 spectra of an entire GOME orbit in one go. Processing selected sets of ground pixels is possible by using gomecal on each set separately.

The input of **gomecal** is a set of extracted ASCII level-1 files, one for each spectrum. The output is corresponding set of ASCII level-1 files with adapted wavelength and/or (ir)radiance values that can be used by other programs, *e.g.* for the retrieval of level-2 products.

The package furthermore contains some additional programs for specific purposes and a few useful UNIX scripts; the programs are written in Fortran-77 and require no other external libraries.

Contents

Notation used on these pages

On these pages, program names are written in **bold font**. Things to be typed, either at the command line or in an input file, filenames and text appearing on the screen or in data files are written in `fixed-width font`.

Hyperlinks in this documentation lead to other HTML pages of the documentation. Apart from that, some links lead to external Web site, *e.g.* for program or project information. In the printed version of this documentation, hyperlinks are underlined and the URL of external web sites is added between brackets.

Files of the GomeCal package mentioned in this documentation are given relative to the main GomeCal directory, *i.e.* the directory where the executables of the programs end up. This file, for example, is `docs/introduction.html`. In this documentation the main directory is named `GomeCal/`.

The programs of the GomeCal package are assumed to be run in a working directory named `gc-example/`, to distinguish it from the main GomeCal directory.

Citation for the published GomeCal documentation

Van Geffen, J.H.G.M.: 2003.
Documentation of the software package GomeCal (Version 1.0),
Technical report TR-255, KNMI, De Bilt, The Netherlands.

The calibration steps

The GOME Data Processor (GDP) is the operational off-line processor that treats the data measured by the GOME instrument (aboard the ERS-2 satellite, launched in 1995). The GDP incorporates a processing of the raw level-0 data into calibrated spectra, the level-1 data. See the chapter Level-1 spectra and the GDP_01 extractor for some information on accessing the GDP-generated level-1 spectra.

This chapter briefly introduces the calibration steps of **gomecal** a user can choose to apply to extracted GDP level-1 spectra in order to improve the calibration of these spectra:

- Wavelength calibration
- Polarisation correction
- Radiometric correction

These calibration steps are symbolised in the GomeCal logo.
Also given in this chapter are:

- Restrictions on usage in gomecal
- Wavelength ranges of GOME's detector channels
- Default wavelength windows

For more details on the correction steps, the user is referred to the given references.

Wavelength calibration

One of the calibration steps to be performed in the level 0-to-1 processing is a wavelength calibration, linking each detector pixel to a specific wavelength. The standard wavelength calibration of the GDP uses spectral lines of an onboard PrCr/Ne hollow cathode lamp to correct for shifts in the wavelengths associated with the detector pixels with respect to a pre-flight calibration. The calibration lamp provides a large set of emission lines throughout GOME's wavelength range, though the coverage is sparse in some parts of the spectral range.

This calibration method, however, proved to be not accurate enough for the retrieval of several level-2 data products, such as ozone columns and ozone profiles. Furthermore, there were signs that the calibration lamp was having problems and might have to be turned off. For these reasons, a new calibration method has been developed. The method uses as reference spectrum a high-resolution solar spectrum, with irradiance values given at 0.01 nm intervals. This spectrum, released by Chance and Spurr [1997], has been obtained from ground-based and balloon-based measurements.

To be able to fit the observations with this reference spectrum, *i.e.* with the Fraunhofer lines in the solar spectrum, the reference spectrum is convolved with GOME's slit function and subsequently integrated over the spectral bins of the detector. The location and width of these bins are allowed to vary along the detector: both a shift and a squeeze are applied to the measurements (the GDP's calibration method with the lamp lines applies only a shift). For more details on the wavelength calibration method and its accuracy, see Van Geffen & Van Oss [2003].

In **gomecal** the wavelength calibration starts from the wavelength grid provided by the GDP in the extracted level-1 spectrum files and updates this grid by performing a recalibration in user-selected wavelength windows. The high-resolution reference spectrum is provided in the form of data block files, one for each of the four GOME channels. The width of the windows used for the calibration as well as the resolution of the GOME slit function can be chosen by the user; default values for the wavelength windows and the resolution of the slit funtion are defined for **gomecal**.

The wavelength calibration gives an improved wavelength grid only in the selected wavelength windows. This means that the wavelength grid is no longer continuous along a channel, but shows steps, which is not desirable for most applications. The user can therefore chose to expand the calibration results of the individual windows to the entire channel, after which the wavelength grid of the channel is continuous again (by default **gomecal** performs this expansion).

The GOME detector consists of four channels (see below) and for earthshine spectra channel 1 is subdivided in band 1a and band 1b. The integration time of the earthshine radiance is 1.5 seconds for all bands/channels, except for band 1a: that band has an integration time of usually 12 (sometimes 60) seconds. In order to calibrate the wavelengths of channel 1, both bands 1a and 1b need to be available. Hence, for each band 1a the appropriate eight (12 sec / 1.5 sec) or fourty (60/1.5) spectra are gathered together and averaged, making all data in channel 1 have the same integration time.

The user can specify the number of earthshine spectra to average before calibrating the wavelength grid. The resulting new wavelength grid is then assigned to all the spectra used in the averaging. The number of spectra to be averaged can be different for the four channels. In view of the difference in integration time of bands 1a and 1b mentioned above, the number of spectra to average for channel 1 refers to the number of band 1a spectra, *i.e.* to 12 (or 60) second integrated spectra. For the other channels/bands the number refers to individual (1.5 second integrated) spectra. This means that if a user wants to calibrated spectra over a total of 240 seconds (with 12 second band 1a data), he/she should set for:

- channel 1: average over 20 (=240 sec / 12 sec) spectra
- channels 2-4: average over 160 (=240 sec / 1.5 sec) spectra

Since the calibration results (the above mentioned shift and squeeze) vary along an orbit [see Van Geffen, 2003], the number of spectra averaged should not be too large -- the numbers given in this example seem a fair choice.

Polarisation correction

Although sunlight itself is unpolarised (when integrated over the entire solar disk), its reflection off the Earth's atmosphere is not: air molecules, cloud droplets, aerosols and ice crystals all contribute to a polarisation of scattered light. This polarisation of the earthshine spectra influences the response of

space-borne instruments such as GOME, and it has to be corrected for in the level 0-to-1 processing to improve the accuracy of products derived from the level-1 spectra, such as the vertical ozone profile.

When extracting GDP level-1 spectra one can give an option to the extractor for a polarisation correction (the `P` parameter in the `c_filter` of the `-c` option), which is based on a theoretical value at 300 nm (UV) and polarisation measurements in 100 nm wide windows around 350, 500 and 700 nm. This correction, however, appears to be insufficient, in particular in the UV range, causing rather large errors in the reflectance of up to 10%.

An improved polarisation correction has been developed by Schutgens & Stammes [2002] and has been implemented into **gomecal**. This correction *fully replaces* the polarisation correction of the GDP extractor. Which means that when extracting level-1 spectra the `P` parameter in the `c_filter` of the `-c` option should *not* be used. By the way, the improved polarisation correction method has also been implemented in the level 0-to-1 processor of the GOME Fast Delivery Service (http://www.knmi.nl/gome_fd/) at KNMI.

The new polarisation correction is based on a parametrisation of the UV earthshine polarisation between 290 and 330 nm, as function of solar zenith angle, viewing geometry, scene albedo and total ozone colum. (For wavelengths above 330 nm the polarisation correction is the same as the correction of the GDP.) The parametrisation is derived from a large number of top-of-atmosphere spectra computed with the DAK radiative transfer model [De Haan *et al.*, 1987]. Only molecular Rayleigh scattering and ozone absorption were considered in a 192-layer plane-parallel atmosphere. For more details on the polarisation correction method and its accuracy, see Schutgens & Stammes [2002].

The correction is computed on the bases of the viewing and zenith angles of the ground pixel being processed, the scene albedo and the total ozone column associated with that pixel, as well as data taken from the polarisation parameters in the PCD file, generated with the `-w` option of the GDP extractor and read by **gomecal**. Also necessary is data from the KeyData Data file; the relevant parts of this file are stored in a data block file.

The *total ozone column* is derived from the climatology of Fortuin & Kelder [1998], which specifies monthly ozone values in 10-degree latitudinal bands. How sensitive the polarisation correction is to the exact value of the ozone column needs to be investigated. Perhaps it is necessary to compute the actual ozone from the level-1 spectra instead of using the climatology, but that will be a rather complex matter as this will require among others ozone cross-section data, which depends on the atmospheric temperature at the geolocation of the level-1 spectra.

The *scene albedo* is computed from the reflectance at 380 nm with an 11-point triangular slit function and a look-up table with entries for the solar zenith angle, the relative viewing angle and the ground pixel type. The look-up table has been generated with the LIDORT model [Spurr, 2001] in the quasi-spherical mode, and is incorporated in **gomecal** as a data block.

Radiometric correction

The calibration steps performed by the GDP includes a calibration of the absolute (ir)radiances. For the retrieval of *e.g.* total ozone columns with a DOAS-technique (which uses reflectivities rather then radiances), this radiomatric calibration seems good enough. For the retrieval of vertical profiles of ozone, however, the GDP's radiometric calibration appears to be insufficiently accurate.

Additionally, it has been observed that the GOME instrument shows a degradation, notably since 1998. This degradation is time and wavelength dependent, and the measured earthshine radiances degrade differently from the solar irradiance. The degradation is only partly corrected for when using the `-e` and `-f` options of the GDP extractor and this correction assumes that solar and earthshine spectra degrade in a similar way. The latter assumption does not affect the retrievals that use reflectivities, but is incorrect when absolute radiances are needed.

In the GOME instrument, Peltier elements are used for cooling the detector. There is some interference of the Peltier cooler conduct signals on the detector signals, for which a correction is included in the standard calibration of the GDP, but this correction does not remove all interference effects.

For these reasons, radiometric correction software was made by Van de A [2001; see also Van der A *et al*, 2002] consisting of three parts which can be applied independently by **gomecal** in addation to the corrections of the GDP:

- a radiometric calibration
- a correction for the degradation of the instrument
- a correction to remove residual effects of the interference of the Peltier cooler signals

The data for the radiometric calibration and the degradation correction are incorporated in the **gomecal** program as a data block and a polynomial functions. The Peltier cooler signal correction is done on the basis of the data from the KeyData Data file; the relevant parts of this file are stored in a data block file.

Restrictions on usage in gomecal

There are a few restrictions on the usage of the calibration steps in **gomecal**, some of which cause the program to skip the level-1 spectrum in question, while others make the program continue with fingers crossed. In any case, a message about the encountered restrictions is written to the logfile.

Wavelength calibration

- The calibration of the wavelengths of channel 1 can only be performed if both band 1a and band 1b are available in at least one spectrum file. If none of the earthshine spectrum files contains band 1a data, then it is not possible to calibrate the wavelengths in channel 1 for any of the earthshine spectra (even if only wavelength windows in band 1b are given).

- The wavelength windows selected for the wavelength calibration have to be at least 1 nm wide for channels 1 and 2, and at least 2 nm wide for channels 3 and 4 (to assure that there are at least a few data points available), but should preferrably be somewhat larger. The windows are not allowed to be wider than 95 nm (which would be a rediculously large window). Note that each window should fall entirely within one of GOME's channels. There can be no more than 10 windows per channel. Default wavelength windows have been defined for **gomecal**; see below.

- The wavelength calibration results are only valid/meaningful for the selected wavelength windows. When using the expansion of the results to the entire channel, the results for a channel have a meaning from somewhat below the first to somewhat above the last window in that channel. For example for channel 1 the first default window is 271.0 to 277.0 nm and the last is 307.5 to 312.0 nm, which implies that the calibration results after expansion are valid for wavelengths above about 260 nm. The reason for this choice of wavelength windows is that the radiometric corrections are applied only to wavelength above 260 nm (see below). If a user wants a good wavelength calibration also below 260 nm, the user will have to define wavelength windows there.

Polarisation correction

- The look-up table for the scene albedo has been made for solar zenith angles (SZAs) between 5 and 85 degrees. Level-1 spectra that have a SZA outside this range are skipped entirely (not even a wavelength calibration is performed). Note that SZAs below 5 degrees never happen for GOME measurements, but SZAs above 85 degrees may occur in polar regions, depending on the season.

- The polarisation correction has been derived for SZAs below 75 degrees (or more precise: $\mu 0 ==$ cosine(SZA) <= 0.25), but for SZAs up to 85 degrees the polarisation correction should still give acceptable results. The value of $\mu 0$ is given in the logfile for each spectrum.

- The polarisation correction is derived for ozone column values between 222 and 439 Dobson Units (DU). The value of the ozone column (retrieved from the Fortuin & Kelder [1998] climatology) is given in the logfile for each spectrum.

- The albedo look-up table for the polarisation correction, the extracts from the KeyData Data file, as well as the radiometric calibration data files, are made only for the normal and small swath ground pixels: east, centre (nadir), west and backscan. There are no entries for other ground pixel types, such as nadir-static, and hence spectra of these ground pixel types are skipped.

- Albedo values make sense only if they are between zero and one, which is also the range for which the polarisation correction is derived. Depending on viewing geometry and SZA it may happen that an albedo value larger than one or smaller than zero results from the look-up tables, which is then set to one or zero, respectively, and marked in the logfile.

- The polarisation correction is also restricted to viewing zenith angles (VZAs) below about 36 degrees (arccos(0.8) to be precise). As GOME's scanning angle is always between about +31 and -31 degrees with respect to nadir, this restriction will in general be no problem; the VZA can be larger in polar regions, depending on the time of year. The value of cos(VZA) is given in the logfile for each spectrum.

Radiometric correction

- The data blocks and polynomial functions for the radiometric calibration and degradation correction are determined up to 1 July 2001. It is not possible to extend this period, due to lack of reliable solar spectra. To at least have some form of correction after 1 July 2001, the correction of 1 July 2001 is applied.

- The begin-date of the radiometric calibration is the date of GOME's launch (21 April 1995). The degradation of GOME became noticable in 1998 and so the begin-date of the degradation correction correction is set at 25 April 1998 (1100 days after the launch).

- The radiometric calibration and degradation correction are applied only to detector pixels with wavelengths between about 265 and 390 nm. Below 265 nm the quality and signal-to-noise of the data is too poor to determine the correction data. Above 390 nm there is no more need for a correction. These limits mean that there will be jumps in the radiance values in the new level-1 data files at about 265 and 390 nm.

- For the degradation correction, it is assumed that the user applies the `-e` and `-f` options of the GDP extractor with the latest available (by DLR supplied) degradation correction data file. There is no check on whether the user has actually applied the standard degration corrections with the `-e` and `-f` options.

- The Peltier cooler correction is applied to all dates since the launch in channel 1 up to 310 nm. Since this correction is determined from the earthshine radiances at three wavelengths (277, 280 and 283 nm), it can only be computed for the band 1a spectra.

Negative (ir)radiance and wavelength values

A line with measurement data in a level-1 spectrum file looks like this:

```
307.2517 1.80462e+11 3.36329e+09 3.21433e-02 0
```

where the first entry is the wavelength, the second the (ir)radiance and the third the absolute error on the (ir)radiance. The last entry, an integer, is a flag indicating the quality of the measurement (this flag is not used by **gomecal**). The wavelength calibration adapts the first entry, the polarisation and radiometric corrections adapt the second entry. The third entry, the error, remains in principle unchanged; only if **gomecal** is instructed to output reflectivities instead of radiances, the error is adapted.

- If an *earthshine spectrum* has for one or more detector pixels a negative wavelength or radiance value, this spectrum is skipped from processing by **gomecal** in the preprocessing of the earthshine spectra, reporting the reason in the logfile. Negative radiance values are indications that there is (probably) something wrong with the measurement of this spectrum. A negative wavelength value is clearly unacceptable.

  The reason to skip the spectrum if a negative radiance value is encountered, is that a recalibration of negative radiances (polarisation and radiometric correction) cannot be performed. And a negative radiance value within a window of the wavelength calibration makes this calibration

impossible

A negative radiance for a given detector pixel is in itself unphysical. In combination with a large absolute error, however, the result can still be positive. For example: reported radiance=-6.22032e+08 and error=1.51819e+09 give a possible radiance of +8.96158e+08. In this case the measurement could perhaps still be used in the calibrations after setting the reported radiance value to zero. This is a matter of further investigation and might be incorporated in a future release of GomeCal. Notably the effect on the interpolations and fitting in the wavelength calibration and radiometric corrections needs attention, as well as the question what radiance to report in the new level-1 spectrum files: zero or the original negative value.

If both the wavelength and the radiance value are positive but the error is negative and/or the flag is non-zero, then the error is set to zero for use in the wavelength calibration. This has no effect on the error reported in the new level-1 data file: the GDP's error is copied unchanged.

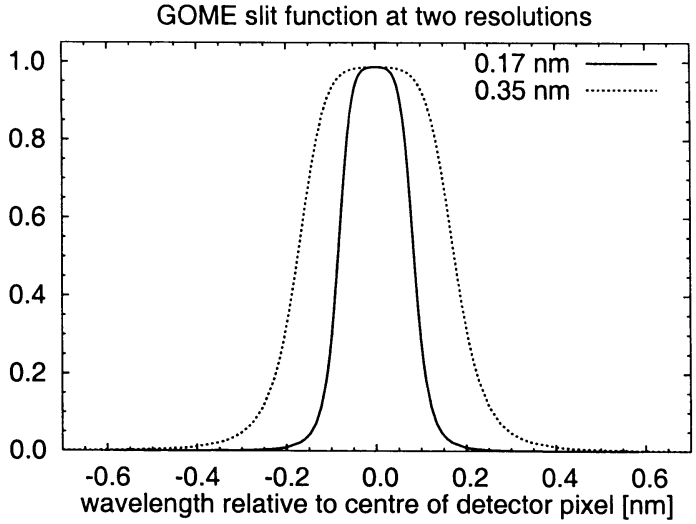
- A *solar spectrum* that has for one or more detector pixels a negative wavelength, irradiance or error value is unphysical and therefore it is unusable in any respect, by **gomecal** and by other programs. When **gomecal** encounters a negative value in the solar spectrum, this is reported in the logfile (with the number of the detector pixel) and **gomecal** stops running altogether.

Wavelength ranges of GOME's detector channels

The following table shows the wavelength ranges of the four detector channels of the GOME instrument, the approximate detector pixel size and the approximate spectral resolution of GOME for each channel. The last column shows the number of data points (detector pixels) for each channel in GDP level-1 spectra.

channel	*wavelength range*	*pixel size*	*resolution*	*#points*
1	237 - 315 nm	0.11 nm	0.17 nm	695
2	313 - 405 nm	0.12 nm	0.17 nm	841
3	407 - 608 nm	0.22 nm	0.30 nm	1024
4	599 - 794 nm	0.22 nm	0.35 nm	1024

The values listed under "resolution" are the default FWHM (full-width at half-maximum) values of the GOME slit function, a normalised flat-topped Gaussian-like function that is assumed to have the same shape for all the channels -- see the graph below. The FWHM values given in the table are approximations, as their value has not been determined well [see also Van Geffen & Van Oss, 2003]. For this reason, it is possible in **gomecal** to manually give the resolution for each of the wavelength windows selected for the wavelength calibration. The resolution of the slit function is not fitted by the wavelength calibration method, because that is probably only useful if the slit function itself is known well, which is not the case for GOME.



Detector channels 1 and 2 are subdivided into two bands ('a' and 'b') for earthshine spectra (not for solar spectra). The band 1a - 1b division point has been changed once since the launch of GOME: after the launch the division was around 307.2 nm (leaving 70 detector pixels in band 1b), between 3 and 8 June 1998 the division was moved to 282.9 nm (with 295 data points in band 1b).
The division point of bands 2a - 2b has not changed and is at 312.1 nm. There are only 9 detector pixels in band 2a and the (ir)radiance values of the first 13 detector pixels of channel 2 seem to be zero always, so that band 2a never has any relevant data points.
Detector channels 3 and 4 are not subdivided and hence band 3 (4) is the same as channel 3 (4).

Default wavelength windows

The following tables show the default window definitions for the wavelength calibration in the four channels and the default FWHM (full-width at half-maximum) values of the GOME slit function in the **gomecal** program.

channel	*wavelength range*	*resolution*
1	271.00 - 277.00 nm	0.17 nm
	277.00 - 287.00 nm	0.17 nm
	292.51 - 302.96 nm	0.17 nm
	307.50 - 312.00 nm	0.17 nm

channel	*wavelength range*	*resolution*
2	315.30 - 325.00 nm	0.17 nm
	325.00 - 345.00 nm	0.17 nm
	345.00 - 355.00 nm	0.17 nm
	360.32 - 370.00 nm	0.17 nm

channel	*wavelength range*	*resolution*
3	422.50 - 437.50 nm	0.30 nm
	475.00 - 490.00 nm	0.30 nm
	525.00 - 540.00 nm	0.30 nm
	575.00 - 595.00 nm	0.30 nm

channel	*wavelength range*	*resolution*
4	645.00 - 665.00 nm	0.35 nm
	697.50 - 712.50 nm	0.35 nm
	742.50 - 757.50 nm	0.35 nm
	770.00 - 785.00 nm	0.35 nm

The windows in channels 1 and 2 are defined for recalibration of level-1 spectra for use in the retrieval of vertical ozone profiles. Such a retrieval used wavelengths above 260 nm, and for that reason there are no windows below 260 nm defined. The last two windows in channel 4 are on either side of the oxygen-A absorption band in earthshine spectra, used for the retrieval of cloud information (cloud top height and cloud fraction). The other two windows in channel 4 and the four windows in channel 3 are spread evenly and cover clear Fraunhofer lines in the solar spectrum.

The default wavelength windows are perhaps not the most suitable or best chosen windows for the purpose the **gomecal** user has in mind for the recalibrated spectra. For this reason it is possible to supply other wavelength windows, overriding the default window definitions.

Installing the package and the documentation

This chapter deals with installing the GomeCal package and its documentation -- a more detailed desription on how to use the individual programs follows in the next chapters.

In this chapter:

Some more specific program matters are discussed later on in this documentation: system requirements, memory usage, system calls, temporary files, etc.

Installing the package

When downloading the GomeCal package from the GomeCal home page (http://www.knmi.nl/gome_fd/gomecal/), it comes as a gzip-ed tar-file `GomeCal-v##.tar.gz`, where `##` gives the version number of the package (without the dot). Unzipping and unpacking this file creates a directory called `GomeCal/` which contains a `README` file with some basic information on getting started, the main `Makefile`, the disclaimer, etc. and which will contain after compilation the executables of the programs.

The unpacking also creates a few subdirectories (see the Appendix for some details):

docs/	contains the most recent version of this documentation in HTML, PostScript and PDF form
examples/	contains some example scripts and such mentioned in this documentation; see also the Appendix
GCsrc/	contains the source files of the programs
GCobj/	will contain the object files after building the programs
Data/	contains some additional data files and programs, used for making data blocks for **gomecal**; see also the Appendix

The user is advised to put the GomeCal/ directory in his/her $PATH environment variable, so that the programs can be run from any working directory without running the chance of accidentally changing something in the GomeCal/ directories. Alternatively, the user can copy the executables to an existing directory in his/her $PATH. In this documentation the working directory is named gc-example/, to distinguish it from the GomeCal/ directory.

make-ing the programs

The programs of the GomeCal package are written in Fortran-77 and should therefore be compiled and linked with a Fortran-77 compiler. The main directory GomeCal/ contains the Makefile to do the make-ing, *i.e.* to build the programs:

```
[gomecal]~/GomeCal> make [all]
```

will make all the programs (the "all" is optional), leaving all the object files in the GCobj/ directory, while the source files are in the GCsrc/ directory. Near the top of the Makefile the options for the compiler are to be specified and the user may have to adapt these in the FFLAGS variable. The programs thus compiled are: **gomecal**, **complv1**, **refspec**, and **wlcomp**. (The programs in the Data/ directory are not built, as they will hardly ever be necessary.)

In this make-ing it is assumed that make understands the VPATH macro to separate between directories for source and object files. Most versions of make understand this macro, but not all do. If your make does not, then download and install for example the free-of-charge GNU Make Utility (http://www.gnu.org/software/make/make.html), which understands the VPATH macro. Alternatively you can use:

```
[gomecal]~/GomeCal> make in.source
```

to compile all programs, where the object files end up in the GCsrc/ (source) directory.

The four individual programs mentioned above can also be compiled separately (only with the VPATH macro):

```
[gomecal]~/GomeCal> make <program_name>
```

The program **gomecal** uses some system calls to execute shell commands. These are programmed as general as possible but may not work correctly on all systems. Notably the progress metre of **gomecal** can be a little problematic. See for some details the section on system calls later on in this

documentation.

In the GomeCal package there are some UNIX shell script ('sh', 'csh' or 'awk') in the main directory, in the docs/ and the examples/ directories. These scripts need to be executable. If all is well, they are executable when downloading and installing the package. If not, then:

```
[gomecal]~/GomeCal> make exec
```

makes all necessary script executable.

Cleaning up

To 'clean' the package directories, that is: to remove the executables and object files of the programs of the package, type:

```
[gomecal]~/GomeCal> make clean
```

Note that this does *not* remove the printable versions of the documentation.
The authors editor produces backup files ending with a tilde (~), other editors add .bak to the filename. Typing:

```
[gomecal]~/GomeCal> make rmbak
```

removes these backup files.

Version information

The current version of the GomeCal package as can be viewed by typing:

```
[gomecal]~/gc-example> gomecal version     or: gomecal -V

 GomeCal -- version 1.0 -- (c) KNMI, 2003

[gomecal]~/gc-example>
```

The version number applies to the full package and its documentation. The number is reported in the gomecal logfile and in the header of the new level-1 spectrum files.

The calibration steps of the **gomecal** program were developed independently and the versions incorporated here are:

wavelength calibration: version 1.8
polarisation correction: version 1.2
radiometric correction : version 4.1

If either of these steps changes, its version number and the version number of the package are adapted.

Version history

version	*released*	*history*
1.0	9 Sept. 2003	Official complete release of the package *New features with respect to version 0.9:* > radiometric calibration data programmed as polynomial functions and data blocks rather then having to use external data files > possibility added to write only a part of the channel data to the new level-1 spectrum files > default windows in channels 3 and 4 for the wavelength calibration defined > layout of the logfile improved > update of the documentation > repair of a few small bugs
0.9	18 Nov. 2002	First public release with full functionality

Last modification dates of the files

To ease the development, each file in the GomeCal package has its own date of last modification, given on a line near the top of the file and this date changes when a file is changed. A list of these dates is given in the `version.txt` file, which is updated with:

```
[gomecal]~/GomeCal> make version
```

which runs the `version.awk` script to extract the dates from the respective files (the script works with `nawk` or `gawk`, not with the old `awk`). For most users the `version.txt` list will not be of much use. The "last modification date" may be removed altogether in a later version of the package.

The documentation

This documentation is a set of interlinked HTML files and can thus be viewed with a Web browser. The reason to chose this format is that it makes consulting the documentation easier than a text file that one has to scroll through. Furthermore, the documentation can be put on-line and can be kept up-to-date easily.

Printing the documentation

In principle one can print the HTML pages using the print option of a Web browser. To make this feasible, all pieces of text at fixed-width fonts and inlined pictures are made to fit at A4 sheets of paper when printing the pages with Netscape.

This way of printing, however, is somewhat cumbersome to do, links to other pages are not marked, page numbering is missing and there is no proper table of contents. Life has been made considerably easier by Jan Karrman in this respect, for he developed the **html2ps** program package (http://www.tdb.uu.se/~jan/html2ps.html): an easy-to-use `perl`-script with configuration files that converts a set of linked HTML files into a PostScript file, with for example links showing up underlined, page numbering, running headers, that the URL of external links is shown, etc. And all of that in one go. (The current version of **html2ps** is "1.0 beta3"; it has unfortunately not been updated since the year 2000.)

For the specific purpose of converting this GomeCal documentation with **html2ps** into PostScript, the script `docs/makedoc.sh` is provided. The script requires that the **html2ps** package is properly installed and that the programs needed by **html2ps** (such as **perl** and **ghostscript**) are installed too, and that these are in the user's `$PATH` environment variable. The conversion to PostScript uses the configuration file `docs/makedoc.cfg` to set **html2ps**' options to get a nice and printable version of this documentation. The resulting PostScript file is named `docs/gcdoc.ps`

With the script `docs/makepdf.sh` the PostScript file is converted into a PDF 1.3 (Acrobat 4-and-later compatible) file `docs/gcdoc.pdf`. As the documentation was written specifically for use with a browser or printing after conversion to PostScript with **html2ps**, the bookmarks in the PDF file do not work fully, but the links in the document seem to work satisfactorily.

The documentation in both PostScript and PDF format are available in the `docs/` directory after downloading the GomeCal package. They are also available via the GomeCal home page (http://www.knmi.nl/gome_fd/gomecal/), both on-line and for download.

To make the printable files, simply type:

```
[gomecal]~/GomeCal> make doc
```

The PostScript and PDF files end up in the `docs/` directory, as mentioned above. To remove the two files, type:

```
[gomecal]~/GomeCal> make rmdoc
```

The PostScript and PDF files are formatted for A4-paper; if another paper format is required, this must be schanged in `docs/makedoc.cfg` and in `docs/makepdf.sh`. Note that `docs/makedoc.cfg` is set to add an empty page to ensure that title page, table of contents and the document itself start at odd pages, thus enabling double-sided printing.

HTML standard used for the documentation

The HTML files of this documentation are made to satisfy the HTML 3.2 (125 kB!) (http://www.w3.org/TR/REC-html32.html) standard of the World Wide Web Consortium (http://www.w3.org/), except for one little thing: to avoid the navigation links at the top and bottom of each HTML file from being printed by **html2ps** too, the **html2ps** specific HTML tag construction

```
<DIV class="NOPRINT">
    Text to be omitted from printing
</DIV>
```

is used, which does not satisfy the HTML 3.2 standard, as there is no attribute `class` in the `<DIV>` element [it is part of HTML 4 (55 kB) (http://www.w3.org/TR/html4/)]. But since the navigation links look very odd and are in fact useless in a printed version, the construction is used anyway.

An on-line HTML validator (http://validator.w3.org/) is available at the World Wide Web Consortium; an off-line validator can, for example, be found at this page (ftp://ftp.math.utah.edu/pub/sgml/index.html) (there seems to be something wrong with the SP-1.3.4 validator when checking HTML 4 documents; checking HTML 3.2 works fine).

Level-1 spectra and the GDP_01 extractor

The GOME Data Processor (GDP) incorporates a processing of the raw level-0 data into calibrated spectra, the level-1 data. The level-1 products are generated and released in a certain compressed (binary) format and extraction software, the GDP_01 extractor, is provided to the user of the data.

The GDP_01 extractor program is called **gdp01_ex**. This program is not part of the GomeCal package: it can be found on the CDs from the GDP. The user is adviced to always use the latest version of the extractor. For details on usage, see the **gdp01_ex** help by typing `gdp01_ex` without any command line argument and the `howto.use` file provided on the CDs, and the GDP manuals (see the References).

In this chapter:

Program concept of the extraction software

Extract from Slijkhuis and Loyola (1999), page 14:

There are four basic calibration steps needed to convert the instrument binary data into calibrated physical quantities.

1. Signal processing: correction for dark signal, FPA crosstalk, pixel-to-pixel gain, and straylight.
2. Wavelength calibration: assigning to each detector pixel its associated wavelength.
3. Radiance calibration: conversion of the corrected detector signals of the earth-shine spectra to radiance units; this includes polarisation correction.
4. Irradiance calibration: conversion of the corrected detector signals of the solar spectra to irradiance units; this includes correction for the BSDF of the diffuser plate.

Furthermore the measurements have to be geolocated, i.e. the geographical position of the footprint on the Earth's surface has to be determined from the instrument's scan mirror angle and from the spacecraft data.

Performing these calibration steps is the task of level 0 to 1 processing. In order to keep the data product as small as possible, the GOME processing is done in two steps. All necessary calibration constants are processed from the calibration measurements in the operational processing from level 0

to 1. The Level 1 Product contains the raw detector signals (binary ADC units) of the science measurements plus calibration constants.

The end-user has to run an Extraction program which applies the calibration constants to the data; this inflates the Level 1 Product to a much larger Level 1 data set containing fully calibrated data (the difference is that the Product contains calibration data plus raw signals as 2-byte integers; the extracted data contain calibrated signals as 4-byte floats plus three 4-byte floats for wavelength, accuracy error and precision error).

Apart from product size considerations, an additional advantage of this procedure is that the user can optionally omit certain calibrations to investigate their influence (or perform the calibration himself), and that by optionally filtering out only a subset of the data (i.e. geographical coverage or wavelength range) the final level 1 data may be kept as small as possible.

Use of level-1 spectra in GomeCal

The level-1 files with the spectra to be used by of the main GomeCal program **gomecal** in a single run are expected to be in a single input directory, where each spectrum has to be in a separate ASCII file. These ASCII level-1 specta are extracted from the binary level-1 files in a specific file name format by using the `-n` or `-d` option (see the options section below).

The program expects earthshine spectra in files named `<filemask>_####.el1`, with `####` the number of the ground pixel of the orbit, where each of these level-1 files contains a single earthshine spectrum (what the `<filemask>` is, is explained below). For each orbit there is one solar spectrum, needed by most of the calibration steps **gomecal** can make. When using the `-n` option, this solar spectrum is in a file named `<filemask>.sun`. With the `-d` option, each `<filemask>_####.el1` file also contains the solar spectrum. The output of **gomecal** is files in the same format, with some comment lines added to the file header; see the section on the new level-1 spectrum files.

If the user wants to process sets of not consecutive ground pixels from one or more orbits, the level-1 spectra either must be in different input directories or must have a different `<filemask>`, and the program has to be run as many times as there are different sets; see the section on running gomecal on multiple filemasks.

Options for the extractor

The main usage of the GDP_01 extractor is:

```
gdp01_ex [options] input_file output_file
```

where the `input_file` is a binary level-1 file from one of the CDs made by the GDP, and `output_file` determines what the extracted ASCII level-1 spectra file name(s) look(s) like.

The correct level-1 filename structure for **gomecal** is achieved by using the right options when extracting level-1 spectra. For example:

```
gdp01_ex -n [-w] -p 306 320 -x yyyy -b yyyynnnnnn \
         -e scdegrad.108 -f scdegrad.108 -c c_filter \
         INDIR/80502203.lv1 Lv1/80502203 > gdp01_ex.out
```

Here the `-n` option makes all spectra go to individual files, the names of which are specified by the last command line argument, in this case `Lv1/80502203`. The solar spectrum file will then be `Lv1/80502203.sun` and the earthshine files will be `Lv1/80502203_####.el1`. In other words, the `<filemask>` is in this case the string `80502203`, and the directory of the input level-1 spectra is `Lv1/`.

If instead of `-n` the option `-d` is given, then the solar spectrum is incorporated in the earthshine `Lv1/80502203_####.el1` files and there is no `Lv1/80502203.sun` file.

The `-w` option tells the extractor to write the calibration data records (PCD and FCD) to files named `<filemask>.pcd` and `<filemask>.fcd`, respectively.
The first of these two is needed by the polarisation correction of **gomecal** as it contains polarisation information for each of the extracted ground pixels. For a full orbit, the PCD file is about 1.5 MB. The FCD file contains calibration data records for the orbit from which ground pixels are extracted, independent of the number ground pixels extracted; its length is about 1.4 MB. The FCD file is not used by **gomecal** but its creation by the extrator cannot be turned off. Hence, writing the FCD data can be redirected immediately to `/dev/null` when extracting by making a link like this:

```
ln -sf /dev/null Lv1/80502203.fcd
```

before running the extractor. Else, the FCD can be deleted after the extraction process.

The `-c` option tells the extractor to apply the calibration parameters, with `c_filter` one or more of:

```
L=Leakage, A=FPA, F=Fixed, S=Straylight, N=Normalize,
B=BSDF, P=Polarization, I=Intensity, U=Unit_Conversion
```

so use *e.g.* `LAFSPNIU`; see the GDP manuals for details on these parameters. Note that if no `c_filter` is given, the extractor uses all calibrations (*i.e.* `LAFSPNIU`) by default.

Make sure, however, that the `P` is *not* specified when **gomecal** has to perform the polarisation correction! The polarisation correction of **gomecal** namely *fully replaces* the extractor's (insufficient) polarisation correction. If `P` is used when extracting a level-1 spectrum, the polarisation correction of **gomecal** cannot (and will not) be applied.

The other parameters of `c_filter` should be used in all cases; **gomecal** performs no checks on this.

Other options used in the above example:
`-p` which ground pixels to extract along the orbit
in the example: pixels 306 to 320 inclusive

. `-x` which ground pixel types to extract: east, centre (=nadir), west, backscan
in the example: all four types
`-b` which bands to extract
in the example: bands 1a, 1b, 2a and 2b, but not bands 3, 4, blind 1a, straylight 1a, 1b and 2a
notes:
- since band 1a has an integration time which is longer than the integration time of the other bands, band 1a does not appear in every data file
- the solar spectrum always has data for all four channels

`-e` instrument degradation and other corrections, which are read from the specified file
`-f` idem; both `-e` and `-f` should be used (see below)

The radiometric correction of **gomecal** is *additional to* the latter two extractor options; see below for a few notes on the GDP degradation correction data file.

Apart from these option, the extractor knows several other useful options, *e.g.* to extract ground pixels in a given geographical region (`-r` option) or between two times (`-t` option); see the **gdp01_ex** help by typing `gdp01_ex` without any command line argument and the `howto.use` file provided on the CDs.

The last command line argument of the extractor is the output file mask, which specifies the names of the extracted level-1 spectra, as mention above. The penultimate command line argument specifies the (binary!) input file for the extractor, in this case `INDIR/80502203.lv1`. The input file's name is generally of the format "YMMDDHHM.lv1", where Y gives the year (if Y>4 then the year is 199Y, else the year is 200Y), followed by the date and start-time of the orbit.

The extractor produces a few lines of output on the screen (similar to the header of the level-1 spectra being extracted) and in the above example this output is redirected to a file. Errors encountered by the extractor, *e.g.* on missing data, are written (added) to a file called `gdp01_ex.err` automatically.

NOTE

The `-a` option should compute albedos instead of radiances for the earthshine level-1 files. This option does not work well and should not be used! Besides, even if it did work, **gomecal** expects radiances in the input level-1 files, not albedos. If the users wishes to have albedos in the earthshine level-1 files, let **gomecal** compute reflectivities.

GDP degradation correction data file

The GDP provides a data file to correct for the degradation of the GOME instrument. With the `-e` and `-f` options of the extractor (see above) the degradation correction can be applied. For this the latest available (by DLR supplied) correction file should be used. At the moment of writing this is `scdegrad.108` (being version 01.08; it is available in the `Data/` directory), which is valid from 15 June 1995 up to 10 April 2001 (there is also a version 01.09 file around, but that file contains a serious error and should therefore not be used). For spectra of dates after 10 April 2001, the `-e` and `-f` options cannot be used (if used, the extractor will issue an error message and exit).

The `-e` option corrects for the degradation in the radiance response (the instrument throughput) and for a seasonal variation in the solar spectra due to an asymmetry in the BSDF of the sun diffuser. The argument of the option is the name of the correction file. According to the manual of the extractor

[Slijkhuis and Loyola, 1999], if only the BSDF-correction is needed, the `-f` option should be used. However, the correction file itself says:

> The option '-e <file_name>' in the version v2.0 of the gdp01_ex does not switch on the asymmetric BSDF-correction for solar measurements as specified in the user manual and help output. The workaround is to use both '-e <file_name>' and '-f <file_name>' flag to correct for degradation and for asymmetric BSDF. This bug will be fixed in the next version.

Hence, a **gomecal** user should apply both options to the extractor for the time being.

Example of usage

An example of a `csh`-script calling the extractor is supplied in the GomeCal package: `examples/getlvl1.csh`.

This script was made for and used on the computer where the GomeCal package is developed: it will need to be adapted for use elsewhere, notably the location of the extractor and of the binary input file. For safety the extraction itself is commented out.

Example of a level-1 directory

For the example in this documentation and in the `examples/` directory, the extractor has been used with the `-n` option and the `-w` option, omitting the GDP's polarisation correction, and extracting only channels 1 and 2:

```
gdp01_ex -n -w -p 306 320 -x yyyy -b yyyynnnnnn \
         -e scdegrad.108 -f scdegrad.108 -c LAFSNIU \
         INDIR/80502203.lv1 Lv1/80502203 > gdp01_ex.out
```

The directory with extracted level-1 spectra then looks like this:

```
[gomecal]~/gc-example> ls -l Lv1/
-rw-r--r--    1 gomecal  user     1346236 Aug  1 09:01 80502203.fcd
-rw-r--r--    1 gomecal  user       15164 Aug  1 09:01 80502203.pcd
-rw-r--r--    1 gomecal  user      169308 Aug  1 09:01 80502203.sun
-rw-r--r--    1 gomecal  user       44710 Aug  1 09:01 80502203_0306.el1
-rw-r--r--    1 gomecal  user       44706 Aug  1 09:01 80502203_0307.el1
-rw-r--r--    1 gomecal  user       76240 Aug  1 09:01 80502203_0308.el1
-rw-r--r--    1 gomecal  user       44712 Aug  1 09:01 80502203_0309.el1
-rw-r--r--    1 gomecal  user       44710 Aug  1 09:01 80502203_0310.el1
-rw-r--r--    1 gomecal  user       44706 Aug  1 09:01 80502203_0311.el1
-rw-r--r--    1 gomecal  user       44710 Aug  1 09:01 80502203_0312.el1
-rw-r--r--    1 gomecal  user       44712 Aug  1 09:01 80502203_0313.el1
-rw-r--r--    1 gomecal  user       44710 Aug  1 09:01 80502203_0314.el1
-rw-r--r--    1 gomecal  user       44706 Aug  1 09:01 80502203_0315.el1
-rw-r--r--    1 gomecal  user       76240 Aug  1 09:01 80502203_0316.el1
-rw-r--r--    1 gomecal  user       44712 Aug  1 09:01 80502203_0317.el1
-rw-r--r--    1 gomecal  user       44710 Aug  1 09:01 80502203_0318.el1
-rw-r--r--    1 gomecal  user       44706 Aug  1 09:01 80502203_0319.el1
-rw-r--r--    1 gomecal  user       44710 Aug  1 09:01 80502203_0320.el1
-rw-r--r--    1 gomecal  user       44712 Aug  1 09:01 80502203_0321.el1
```

```
-rw-r--r--    1 gomecal  user       44710 Aug  1 09:01 80502203_0322.el1
-rw-r--r--    1 gomecal  user       44706 Aug  1 09:01 80502203_0323.el1
-rw-r--r--    1 gomecal  user       76240 Aug  1 09:01 80502203_0324.el1
-rw-r--r--    1 gomecal  user       44712 Aug  1 09:01 80502203_0325.el1
-rw-r--r--    1 gomecal  user       44710 Aug  1 09:01 80502203_0326.el1
[gomecal]~/gc-example>
```

In this case, one can see from the file size that ground spectra `0308`, `0316` and `0324` contain band 1a information and the other 18 earthshine files do not.
The file `80502203.fcd` has about the same size no matter how many ground pixels are extracted; as mentioned above, it is not used and can thus be deleted.

In this example only Channels 1 and 2 have been extracted. When extracting all four Channels, the earthshine spectra files are about as big as the solar spectrum file: 180 kB with and 150 kB without Band 1a. This means that extracting a full orbit, which has some 2200 spectra, consumes a lot of disk space: around 300 MB!!

After the extracter has been run as in the above example, but with the `-d` option, the solar spectrum file is not there and the earthshine files are larger as each of them contains the solar spectrum (with `-p 306 310`):

```
[gomecal]~/gc-example> ls -l Lv1d/
-rw-r--r--    1 gomecal  user       15164 Aug  1 09:07 80502203.pcd
-rw-r--r--    1 gomecal  user      213613 Aug  1 09:07 80502203_0306.el1
-rw-r--r--    1 gomecal  user      213609 Aug  1 09:07 80502203_0307.el1
-rw-r--r--    1 gomecal  user      245143 Aug  1 09:07 80502203_0308.el1
-rw-r--r--    1 gomecal  user      213615 Aug  1 09:07 80502203_0309.el1
-rw-r--r--    1 gomecal  user      213613 Aug  1 09:07 80502203_0310.el1
[gomecal]~/gc-example>
```

after the FCD-file was deleted.

The program gomecal

The main program of the GomeCal package is **gomecal**. It reads extracted GDP level-1 spectrum files, performs one or more user-selected calibration steps and writes the updated spectra to new level-1 files. These files have the same structure as the original files, save for the addition of some comment lines in the file header on the calibrations performed.

In this chapter:

In the discussion below it is assumed that the user is running **gomecal** in a directory called `gc-example/`; the input and output files mentioned are given relative to this working directory.

Main usage

The program **gomecal** can be used in three ways:

1. with a configuration file : `gomecal [config_file]`
2. with command line options : `gomecal -option argument ...`
3. in interactive mode : `gomecal screen` *or:* `gomecal -S`

Command line options all start with a '-' (minus sign) and so **gomecal** checks whether the first character of the first command line argument is a '-' or not (which means that the name of the configuration file cannot start with a minus sign). If no command line argument is given, a configuration file called `gomecal.cfg` is assumed in the working directory.

The author of the package prefers to use a configuration file as setting the options is done with well readable keyword-value pairs, and a previously used configuration file can be updated easilty to work with (somewhat) different options. Other users may prefer the use of command line options. When **gomecal** is called from within a script that performs several tasks, the use of command line options may be more practical.

The three ways of usage of **gomecal**:

1. with a configuration file
2. with command line options
3. in interactive mode

are discussed in the following chapters. The three ways of usage have, of course, the same functionality (except for the COMMENT keyword in the configuration file), hence there is some degree of duplication in the discussion. The remainder of this chapter deals with some more general issues.

Output to screen and message file

Whether using a configuration file or command line options, **gomecal** in principle does not write anything to the screen until it has determined the verbose level the user has chosen. There are three names for a configuration file that are not allowed, as **gomecal** uses these names itself. If either of these is encountered, **gomecal** does write an error message to the screen and exits. Also, there are several command line options that have no function on there own and cannot be used as such. For example:

```
[gomecal]~/gc-example> gomecal gomecal.xmpl

 *** Error: 'gomecal.xmpl' is an invalid name for a configuration file,
            as it is gomecal's example configuration file

[gomecal]~/gc-example> gomecal -Q

 *** Error: invalid single-use command line option: -Q

[gomecal]~/gc-example>
```

Several arguments of either of these two usages are meant to get some output to the screen, as listed in the respective chapters; for example asking for version information or some help. After this output to the screen, **gomecal** exits.

If none of the above applies and **gomecal** can actually begin to read the configuration file or the command line options, the program first of all opens a message file called `gomecal.msg` in the working directory. Any warning or error message encountered during the configuration is written to this message file, not to the screen. If, for example, an unknown configuration file or an unknown command line option is used, this is reported in the file and **gomecal** exits without saying anything:

```
[gomecal]~/gc-example> gomecal x
[gomecal]~/gc-example> cat gomecal.msg
# Message file on the configuration of gomecal
# Version : 1.0 -- (c) KNMI, 2003
# Run date:  4 August 2003
# Filename: x
```

```
 *** Error: configuration file does not exist.

 --- Program gomecal cannot run; stopping.

# end of message file.
[gomecal]~/gc-example>
```

Once all settings are performed and accepted by **gomecal**, the message file is closed and any further messages are written to the logfile in the output directory. (The message file appears in the working directory because when it is made, **gomecal** does not know yet what the output directory will be called.) Depending on the verbose level the user has selected, **gomecal** can run completely quiet or show the progress it is making on the screen. Whether **gomecal** ends normally or prematurely due to an error can be checked via the exit code of the program.

Structure of the program

The processing of the input level-1 spectra by **gomecal** has the following structure, assuming that all calibration steps are to be performed:

1. Configuration of the program
2. Processing of the solar spectrum:
   - spectrum is read and checked
   - wavelength calibration is performed
   - updated spectrum is written to file
3. Earthshine spectra are preprocessed -- *see below*
4. Wavelength calibration of all earthshine spectra
5. Processing of the earthshine spectra:
   - wavelength calibration is applied
   - polarisation correction
   - Peltier cooler signal correction
   - radiometric calibration
   - degradation correction
   - updated spectrum is written to file

A somewhat more detailed overview of the structure of **gomecal** is given in the Appendix.

The reason to first preprocess the earthshine spectra is that then the remainder of the processing is easier, notably the wavelength calibration. The latter is done separately because of the averaging over a given number of spectra before the wavelength calibration is performed.

Preprocessing the earthshine spectra

In the preprocessing step **gomecal** checks whether the input earthshine spectra are in good order and if so which spectral data bands are available and some other details. If for some reason a spectrum cannot be processed, the reason is given in the logfile and the spectrum is skipped from further processing. The following tables list the possible problems with a spectrum and what can still be done with the spectrum in case of a problem.

	result of the preprocessing							
possible processing steps	all is ok	sza	ipx	neg	bnd	GC	inc	err
perform all calibration steps	+	-	-	-	-	-	-	-
copy unchanged files to output	+	+	+	+	+	+	-	-
output of part of the channel data	+	+	+	-	-	-	-	-
output reflectivities instead of radiances	+	+	+	-	-	-	-	-

all is ok	no errors or problems encountered in the preprocessing
sza	the solar zenith angle is not between 5 and 85 degrees *) **)
ipx	ground pixel type is not a normal or small swath ground pixels, but *e.g.* a nadir-static pixel *)
neg	negative wavelength or radiance values were encountered *) **)
bnd	there are no spectral bands found in the level-1 file **)
GC	the spectrum has been calibrated with GomeCal before
inc	the data of a spectral band seems to be incomplete
err	an error occured while reading the level-1 file

*) See the restrictions on usage of the calibration steps.

**) This may occur at the beginning and/or end of GOME orbits over the polar regions, depending on the time of year; negative radiance values may also occur elsewhere along an orbit if there was a problem with the measurement.

Input and output

What has to be set for input and output of **gomecal** via either of the above ways of usage is at least the directory with the input level-1 spectrum files, *i.e.* the directory where the GDP_01 extractor wrote its output to, and the "filemask" used in the arguments to the extractor. Also to be given is the directory where the new level-1 spectrum files have to be written to. This output directory must be different from the input directory, as the updated level-1 files have exactly the same name as the input level-1 files.

As mentioned earlier, **gomecal** requires that each earthshine spectrum is supplied in a separate data file, *i.e.* that the `-n` or `-d` option of GDP_01 extactor is used. If neither of these two options is used, all earthshine spectra are in a single file, which cannot be processed by **gomecal**. In fact, **gomecal** will simply say that no input files are found and exit.

The following subsections discuss some aspects of the input and output files relevant to the three ways of usage of **gomecal**. De discussion below uses the example input data in `examples/Lv1/` and the example configuration file `examples/gomecal.cfg`; the example shell script `examples/gomecal.sh` gives the same results. These make as output directory `examples/NewLv1/`, but to avoid overwriting the results in there, the directory has been renamed to `examples/NewLv1.cfg/`. In the example, the filemask of the files is `80502203`.

Note that **gomecal** is designed to process several level-1 spectra from one orbit, or even the entire orbit, in one go. This means using the program with a single filemask. If a user wants to process selected spectra from several orbits at a time, there should be a loop over several filemasks. Within **gomecal** this is impossible, but the user can make a script calling **gomecal** once for each filemask. This can be done easiest when using command line options; see the section on running gomecal on multiple filemasks.

New level-1 spectrum files

The new level-1 spectrum files have exactly the same name as the input level-1 files and their structure is also the same, except for the addition of a few comment lines in the file header saying what **gomecal** has done to the file. These added lines all start with a #-mark in the first column, and they are added right after the lines listing the calibrations applied by the GDP_01 extractor when producing the input level-1 files. For example (cutting the line "`Leakage FPA ...`" in two parts to keep its contents visible):

```
[gomecal]~/gc-example> more NewLv1.cfg/80502203_0306.el1
/*-----------------------------------------------------------------------------*\
**  GDP Level 0-to-1 Extracting * Version 2.00  * Copyright © DLR 1996-1999  **
\*-----------------------------------------------------------------------------*/
Calibrations Applied
Leakage FPA Fixed Straylight Normalize Intensity Unit_Conversion
                                  Degradation_1.08  Asym_BSDF_Correction_1.08
Asym_BSDF_Correction_1.08
# Corrections applied by GomeCal -- version 1.0 -- (c) KNMI, 2003
#   wavelength calibration  : WLCAL=1 (channels: 1 2), WLEXPAND=1
#   polarisation correction : POLCOR=1
#   radiometric correction  : RADCAL=1, DEGCOR=1, PELTIER=1

Units
```

```
Wavelength [-], Irradiance [-], Radiance [-]
E2GOM158570001KSLVL10 DP19991021094637
ERS Information
...
[gomecal]~/gc-example>
```

The wording used in the comments (WLCAL and such) is the same as the keywords in the configuration file, and in case a wavelength calibration is performed, the channels that are processed are reported. For reference, also the GomeCal version number is reported.

The number of comment lines added by **gomecal** depends on what has happend to the spectrum. If the data file is copied unchanged, only one comment line is added. If either of the calibration steps is performed, then the earthshine spectrum files get four comment lines, as in the example above. The solar spectrum file gets in this case just the two first comment lines, as the polarisation and radiometric corrections are not applied to the solar spectrum. One additional line is added if only a part of the full channel data is written to the file and one line is added if reflectivities instead of radiances are written.

After the comment lines added by **gomecal** there is always an open line (whether the GDP_01 extractor adds an open line at this point depends on the calibrations applied, which is rather unlogical). The line following the open line has the word `Units` and on the next line the units are specified. In the above example the units are marked `[-]`, which is actually an error in the GDP_01 extractor, as the quantities most certainly have a physical unit!! There are four possibilities depending on the options for the GDP_01 extractor used:

- The `c_filter` has both `P` (Polarization) and `U` (Unit_Conversion):
  `Wavelength [nm], Irradiance [photons/s.cm^2.nm], Radiance [photons/s.sr.cm^2.nm]`
- The `c_filter` has `P` but not `U`:
  `Wavelength [nm], Irradiance [mW/m^2.nm], Radiance [mW/sr.m^2.nm]`
- The `c_filter` does not have `P`, regardless whether `U` is present:
  `Wavelength [-], Irradiance [-], Radiance [-]`
  which clearly is an error: either of the above two should be shown in this case.
- In case the `-a` option would be used, which is not possible in combination with **gomecal**:
  `Wavelength [nm], Irradiance [mW/m^2.nm], Albedo [-]`

The line specifying the units is not adapted by **gomecal**, except when reflectivities instead of radiances are written to the output file.

At the beginning of the channel or band data in the spectrum file, there is a line specifying the begin and end wavelength of the channel or band and the number of data points in it. For example for channel 1 of a solar spectrum and bands 1a and 1b of an earthshine spectrum:

```
Channel 1 237.0702 314.7146 0695 0.0176 0 0 0
Band 1a   12.00000 237.0704 307.1438  625 0.0177 0 0 0 0 0
Band 1b    1.50000 307.2517 314.7143   70 0.0177 0 0 0 0 0
```

These lines are copied to the new level-1 files without change, although after a wavelength calibration the actual begin and end wavelength values may be different. When a part of the full channel data is written to the new file, however, the wavelength values and the number of data points are adapted.

The logfile

Once **gomecal** has checked and accepted the configuration settings, it closes the message file `gomecal.msg` in the working directory, as mentioned above, creates the output directory and opens a logfile in that directory. By default **gomecal** writes to the file `FILEMASK.log`, but the user can change that name or tell **gomecal** not to write a logfile.

The logfile starts with a header specifying the settings, which will in general give enough information to do the same processing again at a later stage. Then follows the processing of the input level-1 spectrum files, as outlined in the section on the structure of the program. First, the solar spectrum is read (and recalibrated). Then follows the preprocessing of the earthshine spectra; whether a spectrum has to be skipped and for what reason is given in the logfile (there may be more than one reason for skipping the file, but only one is given). After preprocessing the earthshine spectra, the various calibration steps are performed and the updated earthshine spectra are written to files in the output directory.

Here is an extract from the example logfile, to show the structure of the file:

```
[gomecal]~/gc-example> more NewLv1.cfg/80502203.log
# Logfile from gomecal -- version 1.0 -- (c) KNMI, 2003
#
#                    Run date :  4 August 2003
#
# Level-1 files in            : Lv1/
#    with filemask            : 80502203
# Output directory            : NewLv1/
#    copy unchanged files     : yes
#    radiance or reflectivity : radiance
#    output channel parts     : full channels
#
# Wavelength calibration      : yes
#    channel number           :   1   2   3   4
#    earth spectra averaging  :  20 160 160 160
#    calibration windows      : see 80502203.win
#    expand to full channel   : yes
#
# Polarisation correction     : yes
#    using ozone column       : from climatology
#
# Radiometric correction      : yes
#    Radiometric calibration  : yes
#    Degradation correction   : yes
#    Peltier cooler corr.     : yes
#
# Note that, depending on the spectra to be processed,
# some initial settings may be adapted below.
#
# --- user supplied COMMENT lines:
# Configuration used as example in the GomeCal documentation.
# The output directory is renamed to NewLv1.cfg so that re-running
# gomecal does not overwrite the output.
# --- end of COMMENT lines
```

```
#

Solar spectrum
==============
  spectrum: 80502203.sun == 01-MAY-1998 10:57:29.547

wavelength calibration
----------------------
....
  writing new solar spectrum file to NewLv1/

Earth spectra
=============

Pre-processing spectra
----------------------
....

Wavelength calibration
----------------------
....

Polarisation correction
  given are the computed reflectance and albedo at 380 nm, with:
    *) computed albedo was > 1 or < 0, hence set to 1 or 0, respectively.
  also given are quantities for which the parametrisation is derived:
    222.0 <= O3col <= 439.0  [ozone column from Fortuin & Kelder climatology]
    0.250 <= mu0   <= 1.000  [mu0=cos(sza); sza=solar zenith angle]
    0.800 <= mu    <= 1.000  [mu =cos(vza); vza=viewing zenith angle]
Radiometric correction: wavelength [nm]  begin date  end date    note
  radiometric calib.  : 265.1 - 389.9   21 Apr 1995  01 Jul 2001  a
  degradation corr.   : 265.1 - 389.9   25 Apr 1998  01 Jul 2001  a
  peltier cooler corr.: 235.0 - 310.0   21 Apr 1995  present      b
    a) corrections assumed constant after this date
    b) applies only to level-1 spectra with band 1a information
Output is radiance
Write new level-1 files
----------------------
....

#
# --- end of logfile.
[gomecal]~/gc-example>
```

Messages regarding errors or warnings of **gomecal** may appear in the logfile. When an error occurs **gomecal** will end the program. In case of warnings, **gomecal** will continue with fingers crossed. These messages appear in the logfile preceeded by -->. To quickly locate the lines thus marked, the script `logerr.awk` is provided. For this example:

```
[gomecal]~/gc-example> logerr.awk NewLv1.cfg/80502203.log
line 275:     --> no data for this channel in any of the files
line 277:     --> no data for this channel in any of the files
[gomecal]~/gc-example>
```

In this case the messages simply say that there is no spectral data in channels 3 and 4 of the earthshine spectra, which is locigal as these channels were not extracted for the example.

Unchanged level-1 spectrum files

There are several reasons why a certain level-1 spectrum cannot be processed, as listed in the section on preprocessing the earthshine spectra above and reported in the logfile. Skipping the file would normally mean that an unchanged file does not appear in the output directory, leaving "holes" in the file list. As this may be inconvenient for the user, **gomecal** by default copies unchanged files to the output directory. To distinguish such a file from files that have been updated, the file header gets a message simply saying that the file has been copied unchanged:

```
[gomecal]~/gc-example> more NewLv1.cfg/80502203_0306.el1
/*-------------------------------------------------------------------------*\
**  GDP Level 0-to-1 Extracting * Version 2.00  * Copyright © DLR 1996-1999  **
\*-------------------------------------------------------------------------*/
Calibrations Applied
Leakage FPA Fixed Stray_ight Normalize Polarization Intensity Unit_Conversion
                                    Degradation_1.08  Asym_BSDF_Correction_1.08
Asym_BSDF_Correction_1.08
# GomeCal: file copied unchanged

Units
...
[gomecal]~/gc-example>
```

The user can select *not* to have unchanged files copied, which would save disk space, by setting `COPYFILES: 0` in the configuration file or by using the `-U` command line option.

Output radiances or reflectivities

The input earthshine level-1 spectrum files are always expected to give radiances, not reflectivities: **gomecal** cannot handle reflectivities as input (hence do not use the `-a` option of the GDP_01 extractor). This means that in principle the output of **gomecal** is also radiances. If a user wishes to have reflectivities (albedos) in the new level-1 file, this can be done by **gomecal** (by default it writes radiances). Doing this adds another line to the file header:

```
....
# Corrections applied by GomeCal -- version 1.0 -- (c) KNMI, 2003
#   wavelength calibration  : WLCAL=1 (channels: 1 2), WLEXPAND=1
#   polarisation correction : POLCOR=1
#   radiometric correction  : RADCAL=1, DEGCOR=1, PELTIER=1
#   radiances converted to reflectivities (albedos)
....
```

And the line specifying the units is adapted, saying "`Reflectivity [-]`" rather then "`Radiance [....]`".

If the user specifies that reflectivities have to be written, a line in the header of the logfile says:

```
....
#    radiance or reflectivity : reflectivity
....
```

```
#

Solar spectrum
==============
  spectrum: 80502203.sun == 01-MAY-1998 10:57:29.547

wavelength calibration
----------------------
....
  writing new solar spectrum file to NewLv1/

Earth spectra
=============

Pre-processing spectra
----------------------
....

Wavelength calibration
----------------------
....

Polarisation correction
  given are the computed reflectance and albedo at 380 nm, with:
    *) computed albedo was > 1 or < 0, hence set to 1 or 0, respectively.
  also given are quantities for which the parametrisation is derived:
    222.0 <= O3col <= 439.0  [ozone column from Fortuin & Kelder climatology]
    0.250 <= mu0   <= 1.000  [mu0=cos(sza); sza=solar zenith angle]
    0.800 <= mu    <= 1.000  [mu =cos(vza); vza=viewing zenith angle]
Radiometric correction: wavelength [nm]  begin date  end date    note
  radiometric calib.  : 265.1 - 389.9   21 Apr 1995  01 Jul 2001  a
  degradation corr.   : 265.1 - 389.9   25 Apr 1998  01 Jul 2001  a
  peltier cooler corr.: 235.0 - 310.0   21 Apr 1995  present      b
    a) corrections assumed constant after this date
    b) applies only to level-1 spectra with band 1a information
Output is radiance
Write new level-1 files
----------------------
....

#
# --- end of logfile.
[gomecal]~/gc-example>
```

Messages regarding errors or warnings of **gomecal** may appear in the logfile. When an error occurs **gomecal** will end the program. In case of warnings, **gomecal** will continue with fingers crossed. These messages appear in the logfile preceeded by `-->`. To quickly locate the lines thus marked, the script `logerr.awk` is provided. For this example:

```
[gomecal]~/gc-example> logerr.awk NewLv1.cfg/80502203.log
line 275:     --> no data for this channel in any of the files
line 277:     --> no data for this channel in any of the files
[gomecal]~/gc-example>
```

In this case the messages simply say that there is no spectral data in channels 3 and 4 of the earthshine spectra, which is locigal as these channels were not extracted for the example.

Unchanged level-1 spectrum files

There are several reasons why a certain level-1 spectrum cannot be processed, as listed in the section on preprocessing the earthshine spectra above and reported in the logfile. Skipping the file would normally mean that an unchanged file does not appear in the output directory, leaving "holes" in the file list. As this may be inconvenient for the user, **gomecal** by default copies unchanged files to the output directory. To distinguish such a file from files that have been updated, the file header gets a message simply saying that the file has been copied unchanged:

```
[gomecal]~/gc-example> more NewLv1.cfg/80502203_0306.el1
/*-------------------------------------------------------------------------*\
**  GDP Level 0-to-1 Extracting * Version 2.00  * Copyright © DLR 1996-1999  **
\*-------------------------------------------------------------------------*/
Calibrations Applied
Leakage FPA Fixed Straylight Normalize Polarization Intensity Unit_Conversion
                                Degradation_1.08  Asym_BSDF_Correction_1.08
Asym_BSDF_Correction_1.08
# GomeCal: file copied unchanged

Units
...
[gomecal]~/gc-example>
```

The user can select *not* to have unchanged files copied, which would save disk space, by setting `COPYFILES: 0` in the configuration file or by using the `-U` command line option.

Output radiances or reflectivities

The input earthshine level-1 spectrum files are always expected to give radiances, not reflectivities: **gomecal** cannot handle reflectivities as input (hence do not use the `-a` option of the GDP_01 extractor). This means that in principle the output of **gomecal** is also radiances. If a user wishes to have reflectivities (albedos) in the new level-1 file, this can be done by **gomecal** (by default it writes radiances). Doing this adds another line to the file header:

```
....
# Corrections applied by GomeCal -- version 1.0 -- (c) KNMI, 2003
#   wavelength calibration  : WLCAL=1 (channels: 1 2), WLEXPAND=1
#   polarisation correction : POLCOR=1
#   radiometric correction  : RADCAL=1, DEGCOR=1, PELTIER=1
#   radiances converted to reflectivities (albedos)
....
```

And the line specifying the units is adapted, saying "`Reflectivity [-]`" rather then "`Radiance [....]`".

If the user specifies that reflectivities have to be written, a line in the header of the logfile says:

```
....
#    radiance or reflectivity : reflectivity
....
```

Note that the reflectivity is not computed for unchanged files that are copied to the output directory.

The reflectivity is defined as: (radiance/irradiance) * (pi/cos(SZA))
where SZA is the Solar Zenith Angle and the solar spectrum has been interpolated (spline interpolation) to the radiance grid. The absolute errors in the GDP level-1 spectrum files are converted accordingly:

$$r \pm dr = \frac{pi}{\mu 0}\frac{R \pm dR}{I \pm dI} \sim \frac{pi}{\mu 0}\frac{R}{I} \pm \frac{pi}{\mu 0}\left|\frac{RdI - IdR + dRdI}{I^2}\right|$$

with: r,dr = reflectivity and error
R,dR = radiance and error
I,dI = irrandiance and error; dI << I
μ0 = cos(SZA)

Output full or part of the data channels

By default **gomecal** will write the full spectral data of all the channels present in the input level-1 files to the new level-1 files. As input **gomecal** always requires the full data sets which are in the files coming from the GDP 01 extractor. For the output, however, the user can specify that only a part of the data channel is written to the new level-1 files, *e.g.* just the part of interest for subsequent level-2 retrieval, thus reducing the size of these files.

The way to do this is to specify a string giving either the wavelength range or the detector pixel range to write to the output. Detector pixel numbers and wavelength values are linked to one another on the basis of the wavelength grid of the solar spectrum that is input to **gomecal**, *i.e.* based on the wavelength grid before any wavelength calibration. Evidently, if there is no solar spectrum supplied, it is impossible to write only a part of the channel data to the new level-1 files.

When using **gomecal** with a configuration file the syntax for the selection is:

```
PARTCHAN#: string
```

and when using **gomecal** with command line option the syntax is:

```
-P# string
```

where `#=1,2,3,4` specifies the channel number. In this, the *string* has one of these four forms:

string	*meaning*
`auto`	full channel data written [default]
`none`	channel is skipped entirely
`p npl np2`	range of detector pixel numbers: from `npl` to `np2`
`w wl1 wl2`	range of wavelength values: from `wl1` to `wl2`

The following example, in the notation of the configuration file, shows the usage of each of four possibilities:

```
PARTCHAN1: auto
PARTCHAN2: w 325 340
PARTCHAN3: none
PARTCHAN4: p 788 980
```

When using command line arguments, strings with spaces must be enclosed in quotation marks. For this example:

```
-P1 auto -P2 "w 325 340" -P3 none -P4 "p 788 980"
```

If either of the channels in not specified, the default is used: the full channel is written to the new level-1 files. Any data after the data of channel 4 (the "blind" and "straylight" data; see the options for the GDP_01 extractor) is written to the new level-1 file without change.

Note that this selection of writing only a part of the channel data to the new level-1 files does in no way affect any of the calibration steps **gomecal** performs. More specifically, if for example channel 3 is, as in the above example, marked as not to be written to the output files, this channel will still be calibrated if it is in the data files! To avoid thus wasting CPU time the channel in question is best omitted from the input level-1 files (with the `-b` option for the extractor GDP_01 extractor). To just prevent the wavelength calibration to be performed on a certain channel, select to have no wavelength calibration windows in that channel.

If the user specifies that a part of the channel data is to be written, a line in the header of the logfile says:

```
....
#    output channel parts      : see below solar spectrum
....
```

and below the solar spectrum a table lists the selection (this is below the solar spectrum, because the solar spectrum values are used to link wavelength values and detector pixel numbers). In case of the above example, the logfile would show:

```
....
  writing new solar spectrum file to NewLv1/

Output channel parts
====================
  The output to the new files has only a part of the channels,
  both for the solar spectrum and the earthshine spectra.

      channel  output    pixels       wavelengths
     ------------------------------------------------
         1      full      1 -  695  237.0702 - 314.7146
         2      part    120 -  253  324.9293 - 340.0688
         3      none
         4      part    788 -  980  745.0536 - 785.0710

Earth spectra
=============
....
```

In the header of the new level-1 file, a line is added:

```
....
# Corrections applied by GomeCal -- version 1.0 -- (c) KNMI, 2003
#   wavelength calibration  : WLCAL=1 (channels: 1 2), WLEXPAND=1
#   polarisation correction : POLCOR=1
#   radiometric correction  : RADCAL=1, DEGCOR=1, PELTIER=1
#   only a part of the channels is written to this file
....
```

And the lines in the level-1 data files specifying the begin and end wavelengths and number of data points for each channel or band are adapted according to the specified values, using the wavelength values before any wavelength calibration, as listed in the table in the logfile.

Wavelength calibration windows

The recalibration of the wavelength grids of the level-1 spectra is done in selected wavelength windows by **gomecal**. The program contains a set of default wavelength windows and for each the resolution of the GOME slit function. These default values can be written to a file by typing:

```
[gomecal]~/gc-example> gomecal windefs      or: gomecal -Z
```

This writes the definitions to a file called `windefs.win` in the working directory (see also `examples/windefs.win`).

The user can instruct **gomecal** to use another set of window definitions by supplying a file constructed in the same way as the `windefs.win` file, specifying on one line begin and end wavelength of the window and the resolution of the slit function (which may be different for different windows in the same channel). For example:

```
275.0000    283.3400    0.1680
292.4500    305.0400    0.1700
325.0000    345.0000    0.1900
710.5000    745.8800    0.3400
```

When using **gomecal** with a configuration file the syntax for the selection is:

```
WINDEFS: filename
```

and when using **gomecal** with command line option the syntax is:

```
-W filename
```

where the filename must be given with the path relative to the working directory.

If this way of specifying windows is used, then the wavelength calibration will be performed only on the windows specified in the window definition file. This means that if there is no window in a certain channel (as in this example for channel 3), then that channel is skipped in the wavelength calibration. If for one of the channels the default windows should be used, their definition should be repeated in the window definition file.

It is not necessary to give the windows in the file in an ascending wavelength order, but it is nicer and more logical to do that. Wavelength windows may also partly overlap one another. There is a maximum of 10 windows per channel. The windows in channels 1 and 2 should be wider than 1 nm and the windows in channels 3 and 4 should be wider than 2 nm, so that at least about 10 data pixels are used. Each wavelength window cannot be wider than 95 nm.

When **gomecal** is running, it will write a file called `FILEMASK.win` containing the wavelength window definitons in the output directory.

Output of reference spectra

It is possible to have **gomecal** write the reference spectra in the windows used for the wavelength calibration to files in the output directory (default is not write these files), by using `REFSPEC: 1` in the configuration file or by using the `-Y` command line option. The output files are called `FILEMASK.NWW.ref`, where `N` is the number of the channel and `WW` a 2-digit window number. Example: `80502203.301.ref` for window 1 in channel 3. The file contains both the original high-resolution spectrum and the spectrum after convolution with the slit function.

Extracting the reference spectrum for a given wavelength window or a set of windows can also be done with the additional program **refspec**.

Using gomecal with a configuration file

The program **gomecal** has three ways of usage. This chapter describe the usage with a configuration file:

- Usage with a configuration file
  - Configuration options
  - Example configuration file

Usage with a configuration file

The usage of **gomecal** with a configuration file is:

```
[gomecal]~/gc-example> gomecal [argument]
```

where the `argument` is either a reserved word or a filename:

argument	*function*
`help`	show a help message and exit
`version`	show version information and exit
`example`	write example configuration file `./gomecal.xmpl`, with the default settings and some explanation, and exit
`averages`	show internally defined avering numbers for the wavelength calibration and exit
`windefs`	write internally defined wavelength windows for the wavelength calibration to file `./windefs.win` and exit
`screen`	ask the settings via screen and keyboard, rather then read a configuration file; see the chapter on using **gomecal** in interactive mode
`filename`	the name of the configuration file
<empty>	assume the configuration file is called `./gomecal.cfg`

The `argument` can thus *not* be one of the file names in use by **gomecal**: the above mentioned `./gomecal.xmpl` or `./windefs.win`, or the message file `./gomecal.msg`. If either of these three is used, an error message is written to the screen and **gomecal** exits. Note that the name of the configuration file cannot start with a '-' (minus sign), as that indicates to **gomecal** that command line options are used.

If the `argument` is accepted as the valid name of a configuration file, then **gomecal** opens a message file called `gomecal.msg` in the working directory and writes messages (warnings and errors) to that file while reading and checking the configuration file. The first check is, of course, whether the given configuration file exists: if not, this is reported in the message file and **gomecal** exits. If the configuration file exists it is read and the given settings are checked.

Configuration options

The configuration file consists `KEYWORD: VALUE` pairs, listed in the following table. These pairs may be given in any order in the configuration file. Note that the `VALUE` shoud *never* be enclosed between quotation marks!

Where in the table below gives "0,1" and "n,y" as possible values, this means "no,yes" (giving "no" or "yes" should work too). Links in the description of the functionality of the keyword lead to sections in the chapter on the main usage of the program **gomecal**.

keyword	*possible values*	*default value*	*function*
		Input and output matters	
`INPUTDIR`	*string*	[undefined]	the directory where the input level-1 spectra are located, *i.e.* the directory where the GDP_01 extractor wrote its output to
`FILEMASK`	*string*	[undefined]	the "filemask" used in the arguments to the GDP_01 extractor
`OUTPUTDIR`	*string*	[undefined]	the directory where the new level-1 spectra are written to; this must be different from the `INPUTDIR`
`COPYFILES`	0,1 n,y	1	copy unchanged level-1 spectrum files also to the `OUTPUTDIR`
`REFLECTIV`	0,1 n,y	0	write reflectivities (albedos) rather then radiances to the new level-1 spectrum files

`PARTCHAN1`
`PARTCHAN2`
`PARTCHAN3`
`PARTCHAN4`
	string	auto
		auto
		auto
		auto
			write full or part of the data channels to the new level-1 spectrum files, where "auto" means write the full channel and "none" means skip the channel entirely

`REFSPEC`	0,1 n,y	0	write the reference spectra of the specified wavelength windows to files in the `OUTPUTDIR` (relevant only if `WLCAL=1`)

VERBOSE	0,1,2	1	set the verbose level: 1. be completely quiet (for batch purposes) 2. show a progress metre (nice to see) 3. show more info than that (disadviced)
LOGFILE	*string*	auto	the name of the logfile, where "auto" means using `FILEMASK.log` and "none" means no logfile is made (the file is written in the `OUTPUTDIR`)
Calibration settings			
WLCAL	0,1 n,y	0	perform the wavelength calibration
WINDEFS	*string*	auto	name of the file to read with window definitions for the wavelength calibration, where "auto" means using the internal default definitions (relevant only if `WLCAL=1`)
WLEXPAND	0,1 n,y	1	expand the wavelength calibration results from the individual windows to the entire channel (relevant only if `WLCAL=1`)
NAVERAGE	*set of four values*	20 160 160 160	the number of earthshine spectra to average before the wavelength calibration; one number per channel; giving a 0 (zero) means: do not calibrate that channel (relevant only if `WLCAL=1`)
POLCOR	0,1 n,y	0	perform the polarisation correction
RADCAL	0,1 n,y	0	perform the radiometric calibration
DEGCOR	0,1 n,y	0	perform the degradation correction
PELTIER	0,1 n,y	0	perform the correction to remove residual effects of the interference of the Peltier cooler signals
Comment lines			
COMMENT	*string*	<empty>	a set of lines (maximum 10) which are copied to the logfile (if made)

Example configuration file

Typing:

```
[gomecal]~/gc-example> gomecal example
```

produces an example configuration file `gomecal.xmpl` in the working directory. That file shows the default settings of **gomecal** and some explanation. It is also available in the `examples/` directory of the GomeCal package, and it looks like this:

```
[gomecal]~/gc-example> cat examples/gomecal.xmpl
# ********************************************************************
# gomecal.xmpl ==  1 August 2003
#
# example configuration file for gomecal
# ********************************************************************
#
# This files lists the configuration options for the gomecal program.
# The options are given as:
#    KEYWORD: VALUE
# For a list of KEYWORDs and their valid VALUEs, see the documentation.
# For a list with the default settings, type 'gomecal example'.
#
# Notes:
#  - lines starting with a '#' and open lines are skipped
#  - lines (KEYWORD and VALUE) are read with a character*120 string and
#    should therefore not contain more characters and not contain quotes
#  - the colon ':' behind the KEYWORD is required and there may not be
#    a space separating them
#  - the lines may be given in any order
#  - configuration lines that cause a 'Warning' to appear in the message
#    file gomecal.msg are skipped; an error resulting from a configuration
#    line means that gomecal will not run
#

# Input and output matters:
 INPUTDIR: [undefined]
 FILEMASK: [undefined]
OUTPUTDIR: [undefined]
COPYFILES: 1
REFLECTIV: 0
PARTCHAN1: auto
PARTCHAN2: auto
PARTCHAN3: auto
PARTCHAN4: auto
  REFSPEC: 0
  VERBOSE: 1
  LOGFILE: auto

# Calibration settings:
    WLCAL: 0
  WINDEFS: auto
 WLEXPAND: 1
 NAVERAGE:    20  160  160  160
   POLCOR: 0
   RADCAL: 0
```

```
   DEGCOR: 0
  PELTIER: 0

# Comment lines, copied to the log file (if made):
  COMMENT: Comment strings do not have to be between quotes, as they are read
  COMMENT: with a character*120 string.
  COMMENT: Comments longer than one line must be spread over several COMMENT
  COMMENT: lines, with a maximum of 10 lines.

#
# ****************************************************************
[gomecal]~/gc-example>
```

The second line of this example contains the file name and the date it was made, given in the layout used for listing the date of last modification of the files in the package.

An example of a working configuration file is `examples/gomecal.cfg`, which has the same functionality as the example shell script with command line options `examples/gomecal.sh`.

Using gomecal with command line options

The program **gomecal** has three ways of usage. This chapter describe the usage with command line options:

Usage with command line options

The usage of **gomecal** with command line options is:

```
[gomecal]~/gc-example> gomecal -option [argument] ...
```

where the `-option` is either a "single-use option":

option	*function*
`-H`	show a help message and exit
`-V`	show version information and exit
`-E`	show extended help (list all options) and exit
`-A`	show internally defined avering numbers for the wavelength calibration and exit
`-Z`	write internally defined wavelength windows for the wavelength calibration to file `./windefs.win` and exit
`-S`	ask the settings via screen and keyboard, rather then read a configuration file; see the chapter on using **gomecal** in interactive mode

or a set of configuration options (some with arguments) listed below. If either of these single-use options appears later on in the list on the command line, its functionality overrules all settings thus far.

Note:

gomecal currently accepts options in upper and lower case, but this may change in a future version to upper case only; this documentation lists all options in upper case.

If the first argument on the command line starts with a '-' (minus sign), **gomecal** assumes usage with command line options, and **gomecal** opens a message file called `gomecal.msg` in the working directory and writes messages (warnings and errors) to that file while reading and checking the command line options.

Command line options

All command line options start with a '-' (minus sign). Some options require an argument, others do not. There are four options with arguments that are required, *i.e.* options which have no default functionality. The other options are optional, as they have default values. The options can be given in any order. If a string which contains spaces must be supplied, then this string should be placed between quotation marks. Links in the description of the functionality of the keyword lead to sections in the chapter on the main usage of the program **gomecal**.

option & argument	*function*
	Required options
`-I directory`	the directory where the input level-1 spectra are located, *i.e.* the directory where the GDP_01 extractor wrote its output to
`-F filemask`	the "filemask" used in the arguments to the GDP_01 extractor
`-O directory`	the directory where the new level-1 spectra are written to; this must be different from the input directory
`-C WPRDC`	calibration options to turn on (by default all options are off): `W` Wavelength calibration `P` Polarisation correction `R` Radiometric calibration `D` Degradation correction `C` peltier Cooler correction
	Options with default values
`-U`	do not copy unchanged level-1 spectrum files also to the `OUTPUTDIR` [*default:* do copy]
`-R`	write reflectivities (albedos) rather then radiances to the new level-1 spectrum files [*default:* write radiances]

`-P1 string`
`-P2 string`
`-P3 string`
`-P4 string`
	write full or part of the data channels to the new level-1 spectrum files, where "auto" [*default*] means write the full channel and "none" means skip the channel entirely

-Y	write the reference spectra of the specified wavelength windows to files in the OUTPUTDIR [*default:* do not write reference spectra] (relevant only if W is given at the -C option)
-Q / -M / -T	set the verbose level: -Q be completely quiet (for batch purposes) -M show a progress metre (nice to see) [*default*] -T show more info than that (disadviced)
-L filename	the name of the logfile, where "auto" means using filemask.log [*default*] and "none" means no logfile is made (the file is written in the output directory)
-W filename	name of the file to read with window definitions for the wavelength calibration, where "auto" [*default*] means using the internal default definitions (relevant only if W is given at the -C option)
-X	do not expand the wavelength calibration results from the individual windows to the entire channel [*default:* do expand] (relevant only if W is given at the -C option)
-N values	the number of earthshine spectra to average before the wavelength calibration; one number per channel, to be given between quotes (*e.g.* "20 160 160 160"); giving a 0 (zero) means: do not calibrate that channel [*default:* type gomecal -A] (relevant only if W is given at the -C option)

Note that there is no command line option with which user comment lines can be written to the logfile, as can be done with the COMMENT keyword in case of using **gomecal** with a configuration file.

Example command line options

An example of the use of command line options is given in the shell script examples/gomecal.sh, which has the same functionality as the example configuration file examples/gomecal.cfg; all options using their default settings are omitted. The example script looks like this:

```
[gomecal]~/GomeCal> cat examples/gomecal.sh
#!/bin/sh

# ********************************************************************
# gomecal.sh == 1 August 2003
#
# example shell script to run gomecal with command line options
# ********************************************************************

# The options used here have the same result as the settings in the
# gomecal.cfg configuration file, with only the necessary options given.

gomecal -I Lv1 -F 80502203 -O NewLv1 -C WPRDC
```

```
# See the documentation for details on the usage; to get a brief list
# of available options, type "gomecal -H" and "gomecal -E".

# end of 'gomecal.sh' script
# ***************************************************************
[gomecal]~/GomeCal>
```

Running gomecal on multiple filemasks

If a user wishes to process sets of spectra with different filemasks, *i.e.* spectra from different orbits, the use with command line options is the easiest usage: one can simply create a loop over the filemasks and call **gomecal** for each filemask.

To do this, first a list of filemasks must be made. This can be done by hand if there are not many filemasks. If there are many filemask, it is easier to create a list from the input directory by extracting the filemasks from the solar spectrum filenames, which look like `FILEMASK.sun`, as the following example does (assuming the input files are in the `Lv1/` directory and assuming the output directory `NewLv1/` exists):

```
#!/bin/sh
FILEMASKLIST=`ls Lv1 | grep 'sun' | awk ' { FS="." ; print $1 } '`
for FILEMASK in ${FILEMASKLIST} ; do
   gomecal -I Lv1 -F ${FILEMASK} -O NewLv1 -C WPRDC -Q -N "20 160 0 0"
   mv -f gomecal.msg NewLv1/${FILEMASK}.msg
done
```

The `-Q` option is used to keep **gomecal** quiet, as showing a progress metre is not really handy for such usage. The "`mv -f ...`" line is added to save the message file `gomecal.msg` in the output directory, as else the messages get lost: each time **gomecal** is run it makes a message file called `gomecal.msg`, overwriting an existing file of that name.

Note that the `FILEMASK.sun` files only exist if the `-n` option of the GDP_01 extractor was used, not when `-d` was used; in the latter case the list of filemasks to process must be made differently.

Using gomecal in interactive mode

The program **gomecal** has three ways of usage. This chapter describe the usage in interactive mode :

Usage in interactive mode

The usage of **gomecal** in interactive mode is:

```
[gomecal]~/gc-example> gomecal screen    or: gomecal -S
```

When starting the program like this, it will ask for the settings via brief explanations on the screen and input via the keyboard. If the user supplies input that is incorrect or unacceptable, **gomecal** will say what the problem is and ask the question again (if that is possible, at least: in combination with earlier input, **gomecal** may have to deside to aboard because it does not know how to continue, in which case the user will have to start from the beginning). Some of the questions offer a choice of two or three, in which case one is marked "default": to select that choice, simply hitting the return-key is enough.

When using this method of configuration, the questions are asked in a certain order, and previously given settings cannot be changed, making this way of usage rather inflexible. The reason it is still offered is that a first-time user may wish to see all that is asked step by step.

If **gomecal** has asked for all input needed and is ready to start running, it first writes the given settings to an example configuration file called `gomecal.xmpl`, which the user can use next time, and the processing of the spectra is started.

Example of interactive mode

As an example consider the input as used in the example configuration file `examples/gomecal.cfg`, making a mistake on the name of the input directory and breaking off the program once the processing starts. Information to the screen is marked by `---` and questions by `>>>`, with a space in front. The input should be given starting at the first position (*i.e.* without leading spaces).

```
[gomecal]~/gc-example> gomecal screen

 gomecal -- program to perform a spectral re-calibration of GOME/GDP level-1
 =======    spectra with a wavelength calibration, a radiometric correction
```

```
            and/or a polarisation correction.
            For more details see the documentation.

 === Input and output matters ===

 >>> Give the directory with the level-1 files.
     The files with earth spectra are expected to have as name
     <filemask>_####.el1, with #### the number of the ground;
     pixel; the file with the solar spectrum is <filemask>.sun
inputLv1
 *** Error: directory does not exist; try again
Lv1

 >>> Give the <filemask> of the file names
80502203
 --- Gathering input files ...
     > Number of solar spectra: 1
     > Number of earth spectra: 21

 >>> Give output directory for the new the level-1 files.
     It must be different from the input directory, as the new
     level-1 files will have the same name as the old files.
     Notes: - If the output directory is not present, it is made.
            - Existing files with the same name are overwritten!
NewLv1

 >>> Some input level-1 files may remain unchanged for some reason;
     copy these unchanged file to the output directory?
       0 = no
       1 = yes [default]
1

 >>> Write radiance values to output level-1 files (as is in the input
     files) or reflectivities (solar file remains irradiance)?
       0 = radiances [default]
       1 = reflectivities
0

 --- By default the data of the full channels is written to
     the new level-1 data files. It is possible to only write
     a part of a channel (one window per channel) to the new
     data files. To use this option, run gomecal either with
     a configuration file or with command line options.

 >>> Write reference spectra to files in the output directory?
       0 = no [default]
       1 = yes
0

 >>> Set the verbose level, i.e. what to do when processing the files:
       0 = be completely quiet      = for batch purposes
       1 = show some progress info  = nice to see [default]
       2 = show more info than that = disadviced
1

 >>> Give log file name [default is "80502203.log"];
```

```
     using "none" means no log file is made and
     using "auto" means the default file name.
     The file ends up in the output directory;
     do not supply a directory name here!
auto

 === Calibration settings ===

 >>> Is a wavelength calibration wanted?
       0 = no
       1 = yes, using default wavelength windows [default]
       2 = yes, using user supplied wavelength windows
     To see the default windows, run "gomecal windefs" or
     "gomecal -Z".
1

 >>> Expand the calibration results from the individual windows
     to the entire channel?
       0 = no
       1 = yes [default]
1

 >>> Give the number of earthshine spectra to average before
     the wavelength calibration; one number per channel, i.e.
     four numbers:  NA1  NA2  NA3  NA4  [they must be > 0].
     Example:    20  160  160  160
     Giving a 0 (zero) means: do not calibrate that channel.
20  160  160  160

 >>> Is a polarisation correction wanted?
       0 = no
       1 = yes [default]
1

 --- The radiometric correction consists of three parts:
       a) a radiometric calibration
       b) a correction for the degradation of the instrument
       c) a correction to remove residual effects of the
          interference of the Peltier cooler signals

 >>> Is a radiometric calibration wanted?
       0 = no
       1 = yes [default]
1

 >>> Is a degradation correction wanted?
       0 = no
       1 = yes [default]
1

 >>> Is a Peltier cooler signal correction wanted?
       0 = no
       1 = yes [default]
1

 --- Writing configuration settings to 'gomecal.xmpl'
```

```
 --- All configuration settings done
 --- Initialising the logfile 80502203.log
....
[gomecal]~/gc-example>
```

and **gomecal** will process the input spectra as specified.

Some of the questions asked in the interactive usage depend on previously supplied information and/or the data files found in the input directory. If, for example, the input directory does not contain the PCD file needed for the polarisation correction, this interactive usage of **gomecal** says:

```
 *** The polarisation data file 80502203.pcd does not exist
     hence gomecal cannot perform the polarisation correction.
```

and automatically switches this calibration step off. If no solar spectrum is found in the input directory, then the polarisation and the radiometric corrections cannot be applied, and so **gomecal** will skip asking whether to apply these corrections.

Note that this interactive usage does not permit selecting whether to write full or part of the data channels to the new level-1 files; for that option, use ***gomecal*** *either with a configuration file or with command line options.*

Additional programs

The main program of the GomeCal package is **gomecal**. The package contains several additional programs that may be useful for users:

The program complv1

The program **complv1** is written to be able to compare two level-1 spectrum files. It reads two such files and writes the spectral data to a file in a format suitable for plotting with **gnuplot**. The output file can be used *e.g.* to plot the difference in wavelength grid before and after a wavelength calibration, or the difference in radiance due to the polarisation correction and/or radiometric correction of **gomecal**.

The input level-1 spectra files for **complv1** are of the same type as the input and output files of **gomecal**: either files with a single spectrum (when option `-n` of the GDP 01 extractor was used), or files with both a solar and an earthshine spectrum (option `-d` was used). In the latter case, **complv1** asks which of the two spectra in the file to use for the comparison.

Note that **complv1** is somewhat of a quick-hack to compare spectra and therefore rather inflexible when actually used for studying level-1 spectra. For more possibilities see, for example, the far more general Basic Envisat Atmospheric Toolbox (BEAT) (http://www.science-and-technology.nl/beat/).

Usage of complv1

The program asks via screen and keyboard for the names of the two level-1 spectrum files and for the name of the output file, as the following example shows for the case when both input files contain an earthshine spectrum. Information to the screen is marked by `---` and questions by `>>>`, with a space in front. The input should be given starting at the first position (*i.e.* without leading spaces).

```
[gomecal]~/gc-example> complv1

 --- Program that reads two GDP level-1 files and writes
     the spectral data to a file in a format suitable for
     plotting with Gnuplot.

 >>> Give the name of the 1st file (incl. path)
Lv1/80502203_0308.el1
 --- reading earth spectrum ...
 >>> Give the name of the 2nd file (incl. path)
NewLv1.cfg/80502203_0308.el1
 --- reading earth spectrum ...
 >>> Give the name of the output file [default: complv1.out]
     (an existing file of the same name is overwritten!)
NewLv1.cfg/80502203_0308.cmp

[gomecal]~/gc-example>
```

The "`default`" output filename mentioned above is used after just pressing the return-key, *i.e.* with an empty input string.

Note that via redirection of input, it is possible to use a small input file giving the three input strings for **complv1**. When using this redirection method, the messages written to the screen will appear, but **complv1** does not wait for answers. And in this case an output file name *must* be given as pressing the return-key does not work. In terms of the above example the contents of such an input file is:

```
Lv1/80502203_0308.el1
NewLv1.cfg/80502203_0308.el1
NewLv1.cfg/80502203_0308.cmp
```

without spaces in front of the strings! These three lines are given in the small redirection input file `examples/complv1.xmpl` and its usage is like this:

```
[gomecal]~/gc-example> complv1 < complv1.xmpl

 --- Program that reads two GDP level-1 files and writes
     the spectral data to a file in a format suitable for
     plotting with Gnuplot.

 >>> Give the name of the 1st level-1 file (incl. path)
 --- reading earth spectrum ...
 >>> Give the name of the 2nd level-1 file (incl. path)
 --- reading earth spectrum ...
 >>> Give the name of the output file [default: complv1.out]
     (an existing file of the same name is overwritten!)

[gomecal]~/gc-example>
```

If either of the two input files contains two spectra (the first being solar, the second earthshine), then **complv1** asks which of the spectra to use:

```
[gomecal]~/gc-example> complv1

 --- Program that reads two GDP level-1 files and writes
     the spectral data to a file in a format suitable for
     plotting with Gnuplot.

 >>> Give the name of the 1st level-1 file (incl. path)
Lv1d/80502203_0306.el1
 --- The level-1 file contain two spectra:
       1) solar spectrum
       2) earthshine spectrum
 >>> Which of these two to use?
2
 --- reading earth spectrum ...
 >>> Give the name of the 2nd level-1 file (incl. path)
Lv1d/80502203_0308.el1
 --- The level-1 file contain two spectra:
       1) solar spectrum
       2) earthshine spectrum
 >>> Which of these two to use?
2
 --- reading earth spectrum ...
 >>> Give the name of the output file [default: complv1.out]
     (an existing file of the same name is overwritten!)
Lv1d/earth0306_0308.cmp

[gomecal]~/gc-example>
```

Remark

With **complv1** it only possible to compare the same type of level-1 spectra files: either two files containing the full channel data or two files containing a part of the channel data (the latter can be the output of **gomecal**; see the section on output full or part of the data channels). If one level-1 spectrum contains the full data of a given channel while the other one contains only a part of the data of that channel, **complv1** will skip that channel and write a message to the output file saying: "`unequal number of data points; skipping`".

Data format

The output file (`NewLv1.cfg/80502203_0308.cmp` in the first example above) is in a format suitable for plotting with **gnuplot**. The file starts with a comment header, which is followed by the data:

```
[gomecal]~/gc-example> more NewLv1.cfg/80502203_0308.cmp
# Data file made by complv1.f
# data read from 1: Lv1p/80502203_0308.el1
#            and 2: NewLv1.cfg/80502203_0308.el1
#  spectral type 1: Earthshine
#  spectral type 2: Earthshine
#
#
# Output format:
#
```

```
# channel  ------ file 1 ---------    ------ file 2 ---------
# index     wavel.    (ir)radiance     wavel.    (ir)radiance

# Band 1a
      1    237.0704   0.380224E+10    237.4973   0.324840E+10
      2    237.1930   0.107352E+10    237.6159   0.906914E+09
      3    237.3156   0.100724E+10    237.7345   0.844898E+09
....
    624    307.0359   0.173945E+12    307.0236   0.122019E+12
    625    307.1438   0.186065E+12    307.1311   0.131044E+12

# Band 1b
      1    307.2517   0.180462E+12    307.2386   0.127542E+12
      2    307.3596   0.172142E+12    307.3461   0.122008E+12
....

# Band 2b
....
    830    404.9573   0.250650E+14    405.0816   0.238586E+14
    831    405.0667   0.243508E+14    405.1924   0.231808E+14
    832    405.1762   0.247835E+14    405.3032   0.235949E+14

# Band 3
#  --> missing in one or both files; skipping

# Band 4
#  --> missing in one or both files; skipping
#
# --- end of file.
[gomecal]~/gc-example>
```

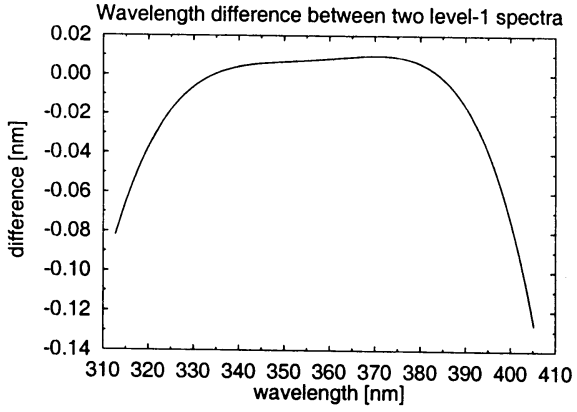
The data is organised per band or channel. If a band or channel is missing in one or both files, that band or channel is skipped and a comment about this is written to the output file.
If from an existing band or channel the (ir)radiance value of either of the two files is zero or negative at a certain wavelength value, this line is preceeded by a #, so that **gnuplot** does not plot this irrelevant value (this happens for example with Band 2a, which (always?) has zero radiance values).

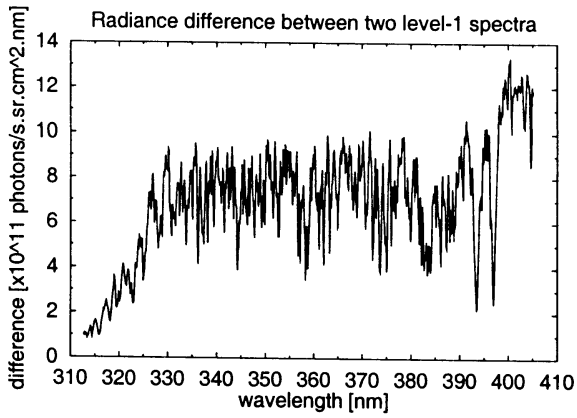
Plot example

The band or channel data are given with double open lines between them, to make it possible to plot one band or channel at a time using the `index` keyword of the `plot` command from **gnuplot**. For example, to plot the difference in wavelengths of band 2b, type within **gnuplot**:

```
plot "80502203_0308.cmp" index 2 using ($2):($2-$4) with lines
```

where the wavelength of the first level-1 spectrum (column 2) is plotted along the x-axis. Note that band 1a has index 0, band 1b has index 1, band 2a is empty and is absent, and so band 2b has index 2. Similarly, the difference in radiance (columns 3 and 5) can be plotted. After making the plot look nice with some formatting instructions, this are the resulting graphs:





The graph on the left shows a difference in wavelength at the end of band 2b of up to the width of the detector pixels (0.12 nm). This is actually an unrealisticly large difference, which is due to the fact that there are no windows for the wavelenght calibration above 370 nm: the calibration results should threfore not be just for wavelengths above, say, 385 nm. The difference in radiaces, plotted in the right graph, is due to a combination of the polarisation and radiometric corrections.

The program refspec

The program **refspec** writes the reference spectrum in selected wavelength windows to a file, containing both the original high-resolution solar reference spectrum and that spectrum after convolution with the GOME slit function at the specified resolution (the full-width at half-maximum, FWHM). The usage is either:

```
[gomecal]~/gc-example> refspec window_definition_file
```

or:

```
[gomecal]~/gc-example> refspec wlmin wlmax resol [outfile]
```

In the first case the command line argument is a window definition file used by **gomecal**. The output files are then named `refspec.NWW.ref`, where `N` is the number of the channel and `WW` a 2-digit window number. For example, using the file `examples/windefs.win`:

```
[gomecal]~/gc-example> refspec windefs.win
 --- Reading window definitions from file ...
 --- Writing reference spectra ...
     refspec.101.ref:  271.0000 -  277.0000 nm
     refspec.102.ref:  277.0000 -  287.0000 nm
....
     refspec.403.ref:  742.5000 -  757.5000 nm
     refspec.404.ref:  770.0000 -  785.0000 nm
[gomecal]~/gc-example>
```

This usage is similar to having **gomecal** write the reference spectra of the selected windows (using "`REFSPEC: 1`" in the configuration file or `-Y` as command line option).

In the second case `wlmin` and `wlmax` define the begin and end wavelength of the window and `resol` the resolution of the GOME slit function to use for the convolution. The fourth argument gives the name of the output file; if it is omitted, then `respec.N01.ref` is used, with N the number of the channel. For example:

```
[gomecal]~/gc-example> refspec 750 770 0.37
 --- Writing reference spectra ...
     refspec.401.ref:  750.0000 -  770.0000 nm
[gomecal]~/gc-example> more refspec.401.ref
# High resolution solar reference spectrum from sunref4.f
#
#    wavelength window: 749.000 - 771.000 nm
#    wavelength step  :   0.010 nm
#    resolution slit  :   0.370 nm
#
# Column 1: index number
#        2: wavelength [nm]
#        3: high-res. irradiance [photons/s.cm^2.nm]
#        4: convolved irradiance [photons/s.cm^2.nm]
#
    1  749.000   0.4902180E+15   0.4879775E+15
    2  749.010   0.4869120E+15   0.4878456E+15
    3  749.020   0.4904520E+15   0.4877535E+15
    4  749.030   0.4915260E+15   0.4876743E+15
....
 2198  770.970   0.4809600E+15   0.4808428E+15
 2199  770.980   0.4817290E+15   0.4808554E+15
 2200  770.990   0.4808000E+15   0.4808688E+15
 2201  771.000   0.4789640E+15   0.4808309E+15
#
# ---- end of file
[gomecal]~/gc-example>
```

As the above example shows, 1 nm is added to either side of the specified window. This is done in order to make sure that the convolution of the spectrum with the slit function works well also at the edges of the specified window.

The data in the `respec.NWW.ref` files can be used for quickly comparing the original high-resolution and the convolved reference spectrum with each other and/or with a measured spectrum.

There are some restrictions on the wavelength windows, similar to the restrictions for **gomecal**: a window should be wider than 1 nm for channels 1 and 2, and wider than 2 nm for channels 3 and 4 (to assure that there are at least a few data points available), but not wider than 95 nm. Note that a window should fall entirely within one of GOME's detector channels.

The program wlcomp

The program **wlcomp** is a small interactive program that computes a wavelength value from 5 polynomial coefficients that make up the wavelength grid of a channel in a level-1 file and a pixel index number within that channel using the following formula:

```
wl(ip) = sum (i=1...5) [a(i)*dble(ip-1)**(i-1)]
```

with `ip` the pixel index number within the channel (which in general starts counting at 1 for the first pixel) and `a(i)` the five coefficients. This is the formula used in the wavelength calibration routines of **gomecal** to characterise the wavelength grid of a given channel. The five coefficients of the recalibrated wavelength grid can be found in the logfile made by **gomecal** (if the user choose to have it made).

The **wlcomp** program was made to be able to quickly check whether the wavelength calibration routines were doing their job properly, and as **wlcomp** exists it is added to the GomeCal package, though it is probably not very useful to most users.

When running **wlcomp** the five coefficients are asked for, followed by a pixel number. Here is an example of the usage:

```
[gomecal]~/gc-example> wlcomp

 --- Program that computes a wavelength from a 5-th degree polynomial:
          wl(ip) = sum (i=1...5) [a(i)*dble(ip-1)**(i-1)]
     with ip the pixel index number and a(i) the coefficients.

 >>>  Give the five wavelength coefficients: a(i) for i=1,2,3,4,5 resp.
0.3111786209515E+03  0.1164546012767E+00 -0.7943133436766E-05
0.3003717816705E-08  0.1954538000562E-15

 >>> Give the pixel number index ip;
     a 0 (zero) or <return> ends the program
110
 ===> wl(110) = 323.7816900515685 nm

 >>> Give the pixel number index ip;
     a 0 (zero) or <return> ends the program
208
 ===> wl(208) = 334.9710106550839 nm

 >>> Give the pixel number index ip;
     a 0 (zero) or <return> ends the program

[gomecal]~/gc-example>
```

Note that on different machines the decimals in the values shown in this example may be different due to rounding and machine accuracy.

Program matters

Apart from the program matters discussed in the chapter on Installing the package and the documentation, there are some other relevant points that need to be mentioned:

- System requirements
  - External routines
  - Required programs
  - Memory and disk usage and computational time
- Exit codes
- System calls
  - Progress metre
  - Temporary files

System requirements

The GomeCal code is written in Fortran-77 to run on UNIX machines. Hence, the GomeCal package requires the presence of a Fortran-77 compiler; additional libraries are not necessary. To build the codes, the `make` utility used should understand the VPATH macro; see the section on `make`-ing the programs.

The code has been compiled and tested successfully on the following systems:

no.	*machine*	*operating system*	*compiler*	`make` *utility*
1	Red Hat Linux release 7.2	Kernel 2.4.9-13 on an i686	GNU Fortran compiler (v0.5.26)	GNU make utility (v3.79.1)
2	Red Hat Linux release 7.2	Kernel 2.4.9-13 on an i686	Portland Group's FORTRAN 77 compiler (v3.2-4)	GNU make utility (v3.79.1)
3	Silicon Graphics workstation	IRIX Release 6.5 IP32	MIPS and MIPSpro F77 compiler (v7.2.1)	GNU make utility (v3.78.1)
4	Sun	SunOS 5.8	Sun WorkShop(TM) 6 update 2 FORTRAN 77	SunOS make utility

External routines

Apart from the standard Fortran-77 coding and the routines in the GomeCal package, the program **gomecal** uses a small number of external routines. These routines, listed below, are not really standard Fortran-77, but are recognised and included by most (all?) modern compilers.

routine	*purpose*
`function iargc()`	Acquire the number of command line arguments `na` supplied to the program: `na=iargc()`
`subroutine getarg(n,arg)`	Read command line argument `n` and store it in string `arg`
`subroutine exit(nexit)`	Generate an exit code with value `nexit`; see the section on exit codes below for details
`subroutine flush(nunit)`	Flush all output to the file connected to unit `nunit`
`subroutine system(cmd)`	Send command `cmd` to the system; see the section on system calls below for details

Required programs

Some of the shell scripts in the package as well as the system calls that **gomecal** performs use the standard `sh` UNIX shell, while some other scripts use the `csh` UNIX shell or the `awk` script language, all of which should be available on any UNIX system. The system calls also use a number of other UNIX commands, listed in the relevant section below.

Printing the documentation requires the software package **html2ps**, which in turn requires **ghostscript** (`gs`) and `perl` to be in the users `$PATH` environment variable.

Memory and disk usage and computational time

The GomeCal package after installation and compilation of the programs and including the PostScript and PDF version of the documentation uses about 13 MB of disk space, depending somewhat on the compiler and the compiler options used. The executables of **gomecal** and **refspec** are quite large because these include large data blocks.

The process of extracting the GDP level-1 spectra takes up quite a lot of time, especially when extracting ground pixels near the end of an orbit. Extracting all ground pixels of the orbit `80502203` used for the examples in the `examples/` directory (being pixels `0001` through `2237`) with only channels 1 and 2 results in about 96 MB of data files.

Processing this set of data with `examples/gomecal.cfg` (wavelength calibration of channels 1 and 2 only) costs about 160 seconds while showing the progress metre and 95 seconds without the progress metre on machine 2 in the above list, which has a 1.7 GHz processor and compiling was done with "`pgf77 -fast`". While running the memory usage of **gomecal** was about 13 MB.

In the run without the progress metre, most of the computational time (about 50%) is spend on reading and writing the ASCII files containing the spectra, which depends on the length of the level-1 spectra (*i.e.* how many channels are extracted per spectrum) and the number of spectra. Nearly 40% of the time is spend on the wavelength calibration, which depends on the number of wavelength windows and their length, and the number of earthshine spectra averaged before calibration. The rest of the time is spend on the other computational tasks. The slower the processor, the more time the computational tasks will take and the less time (in terms of a percentage) is spend on input/output.

Exit codes

By default the exit code of a program that ends normally is '0' (zero), while the exit code is '1' (one) if the program ends abnormally, *e.g.* if it crashed due to a runtime error, and it is thus possible to check whether the program has ended normally or not. Under UNIX this can be done *e.g.* by typing:

```
echo "exit code: "$?
```

right after the program has ended.

In the case of **gomecal**, the '0' and '1' exit codes are generated in this standard way. Additional, two exit codes have been introduced, using a "`call exit(nexit)`" statement just before the very end of the main program. (Note that a "`call exit(nexit)`" means an immediate end of the program at that point in the program; any statement below that is not processed.)

The meaning of the exit codes of **gomecal** is as follows:

exit code	*meaning*
0	**gomecal** ended normally. The configuration of **gomecal** was successful and the processing of the spectra was started; the message file `gomecal.msg` lists some info on the configuration. Any possible warnings or errors from the processing are written to the logfile (if that file is made).
1	**gomecal** ended abnormally due to a runtime error.
2	**gomecal** ended normally, but an error occured *before* the configuration of **gomecal** started and **gomecal** exits without making a message file. The reason of the error is reported on the screen and is either an invalid name of the configuration file or an invalid single-use command line option; see the examples in the section Output to screen and message file
3	**gomecal** ended normally, but an error occured *during* the configuration of **gomecal**, making any processing of spectra impossible, *e.g.* because a configuration setting is used incorrectly, an essential configuration setting is missing, required files do not exist, etc. The reason of the error is reported in the message file `gomecal.msg`; a logfile is not made.

System calls

The program **gomecal** uses some "`call system()`" statements to execute UNIX sh-shell commands. The routines that perform such system calls are all put together in the source file `GCsrc/syscalls.f`, with "`shcommand()`" being the actual routine performing the system call. The system calls used by **gomecal** are listed in the following table.

routine name	*functionality*	*command*
`shcommand`	Perform the actual system call for command CMD	call system('/bin/sh -c CMD')
`rundate`	Put todays date in the "DD Month YYYY" format in a string, with the name of the month in full (*e.g.* "15 September 1997"), which is used in gomecal's message file and logfile	date +"%e %B %Y"
`existdir`	Check whether directory DIR exists	if [-d DIR] ; then \ echo "1"; else echo "0"; fi
`makedir`	Make directory DIR and all its non-existing parent directories	mkdir -p -m 755 DIR
`getinfiles`	Gather the input level-1 earthshine spectra in input directory DIR having filemask MSK	ls DIR | egrep "^MSK_.....el1S"
`metrebegin`	Start the progress metre with message string MSG -- *see below*	echo -n "MSG spectrum 0000"
`metrerun`	Update the last four characters of the progress metre with string NNNN -- *see below*	echo -n "\b\b\b\bNNNN"
`metredone`	End the progress metre -- *see below*	echo " -- done."

The system call itself (in `shcommand`) is programmed as general as possible and should work fine on UNIX machines. To make sure that the behaviour is always the same, the `/bin/sh` shell is used, with the `-c` flag to make sure that the command is read from a string. What does not behave exactly the same on all systems is the `echo` command used for the progress metre, hence the following subsection.

Progress metre

The progress metre of **gomecal** is activated by setting the verbose level to '1'. The progress metre shows in a nice way what the program is doing by *e.g.* showing a counter of the number of the earthshine spectrum being treated. To do this, the UNIX command `echo` is used, as listed in the table above.

It appears that `echo` can perform slightly different on different systems and it also seems to depend on the compiler and the options used for building the code. On some systems, for example, the `-n` for omitting a new-line at the end is absent. And in some cases the backslash should be given double to get it working as an escape sequence character, depending on the settings of the compiler.

All of this means that for `metrebegin` there are three possibilities given in the `GCsrc/syscalls.f` file:

```
call shcommand("echo -n '      "//msg(1:lnmsg)//"spectrum 0000'")
call shcommand("echo '      "//msg(1:lnmsg)//"spectrum 0000\c'")
call shcommand("echo '      "//msg(1:lnmsg)//"spectrum 0000\\c'")
```

where `msg(1:lnmsg)` is the message string to be printed, the double forward slash `//` is the Fortran code for concatenating strings, and `\c` (or `\\c`) is the escape sequence convention to print a line without a 'new-line' at the end.

One of these three lines should be enabled, depending on the system **gomecal** is running on and the compiler in use. For the machines 1, 2 and 3 listed above, the first of the three `echo`-statements works fine, while for machine 4 the third `echo`-statement has to be used. The GomeCal package is released with the first of the three lines enabled.

Similarly, for `metrerun` the possibilities are:

```
call shcommand("echo -n '\b\b\b\b"//specnum//"'")
call shcommand("echo '\b\b\b\b"//specnum//"\c'")
call shcommand("echo '\\b\\b\\b\\b"//specnum//"\\c'")
```

where `specnum` is the 4-digit spectrum number to show. In these lines `\b` (or `\\b`) is the escape convention for a backspace, used to overprint the previously shown spectrum number. Again, one of these three should be enabled.

The progress metre is ended with a simple command that comes in one possibility only:

```
call shcommand("echo ' -- done.'")
```

The following example of running **gomecal** (using the `examples/gomecal.cfg` configuration file) has verbose level '1' and thus shows the progress metre. In real all characters appear in the same font: *italic* is used here to indicate the parts on which the `echo` with the backspaces are used.

```
[gomecal]~/gc-example> gomecal

 --- Checking configuration options ...
 --- All configuration settings done.
 --- Initialising the logfile 80502203.log
 --- Gathering reference spectra ...
     channel 1, spectrum  ref -- done.
     channel 2, spectrum  ref -- done.
```

```
     channel 3, spectrum  ref -- done.
     channel 4, spectrum  ref -- done.
 --- Processing the Solar Spectrum ...
     channel 1, spectrum  sun -- done.
     channel 2, spectrum  sun -- done.
     channel 3, spectrum  sun -- done.
     channel 4, spectrum  sun -- done.
   > writing new solar spectrum file to NewLv1/
 --- Pre-processing the Earth Spectra ...
     checking spectrum 0326 -- done.
 --- Processing the Earth Spectra ...
   > wavelength calibration ...
     channel 1, spectrum 0326 -- done.
     channel 2, spectrum 0326 -- done.
   > applying wl-calibration ...
   > polarisation correction ...
   > radiometric corrections ...
     all channels, spectrum 0326 -- done.
 --- All done.

[gomecal]~/gc-example>
```

Note that a similar progress metre is used by the **keydat** program in the Data/ directory. As it is unlikely that a GomeCal user has to run that program, it is not set up as general as the above mentioned routines.

Temporary files

When running **gomecal**, some temporary files are used by some of the routines performing a system call (see above). These temporary files are placed in the working directory, where **gomecal** is run, and the files are deleted again after use by **gomecal**. Existing files with these names are overwritten.

The following three routines performing a system call use a temporary file:

routine name	*temporary file in working directory*
rundate	gmcl-date.tmp
existdir	gmcl-xdir.tmp
getinfiles	gmcl-list.tmp

The script docs/makedoc.sh used for printing the documentation writes to a temporary file called docs/gcdoc-tmp.ps, before making the actual documentation file docs/gcdoc.ps.

The programs **keydat** and **o3block** in the Data/ directory also use a temporary file (Data/keydat-date.tmp and Data/o3blck-date.tmp, respectively) to retrieve the date.

Final remarks

This chapter contains some final remarks, namely:

Contact name and address

For comments on the programs of the GomeCal package (usage, input, output, possible additional features, bugs, etc.) please contact the author of the package: Jos van Geffen at geffen@knmi.nl.

When reporting problems or bugs, please be as specific as possible and send at least the configuration file or list of configuration options, which GOME orbit was used, which ground pixels were extracted (and with which options), etc., and mention the version number of the package.

For details on the correction steps performed by the package, the user is referred to the references given below.

Postal address:

KNMI -- Royal Netherlands Meteorological Institute
Climate Reseach and Seismology Department
P.O. Box 201
3730 AE De Bilt
The Netherlands
tel.: +31 (0)30 2206 911 (= general)
fax.: +31 (0)30 2210 407

Acknowledgements

The author of the GomeCal package wishes to thank the following persons for help, suggestions, information and discussions:

Ronald van de A, Sidney Cadot, Sander Niemeijer, Roeland van Oss, Ankie Piters, Nick Schutgens, Gijs van Soest Rob Spurr, Piet Stammes, Pieter Valks, Pepijn Veefkind, Robert Voors, Claus Zehner.

The GomeCal package was developed and used at the Royal Netherlands Meteorological Institute (KNMI (http://www.knmi.nl/)) within the framework of and/or for the following projects:

= BEAT : Basic Envisat Atmospheric Toolbox (http://www.science-and-technology.nl/beat/)
= GOA : GOME Assimilated and Validated Ozone and NO2 Fields (http://www.knmi.nl/goa/)
= GOFAP : GOME Ozone Fast delivery and value-Added Products (http://www.knmi.nl/gome_fd/)
= OMI : Ozone Monitoring Instrument (http://eos-chem.gsfc.nasa.gov/instruments/omi/introduction.html)
= TEMIS : Tropospheric Emission Monitoring Internet Service (http://www.temis.nl/)

References

Details on the calibration steps can be found in the following papers:

Wavelength calibration -- Jos van Geffen (geffen@knmi.nl)

- Van Geffen, J.H.G.M. van Van Oss, R.F.: 2003, "Wavelength calibration of spectra measured by the Global Ozone Monitoring Experiment by use of a high-resolution reference spectrum," *Applied Optics* **42**, 2739-2753.
- Van Geffen, J.H.G.M.: 2003, "Wavelength calibration of spectra measured by the Global Ozone Monitoring Experiment: variations along orbits and in time," *Applied Optics* , submitted.

Polarisation correction -- Nick Schutgens (schutgen@knmi.nl)

- Schutgens, N.A.J. and Stammes P.: 2002, "Parametrisation of Earth's polarisation spectrum from 290 to 330 nm," *J. Quantitative Spectroscopy and Radiative Transfer* **75**, 239-255.
- Schutgens, N.A.J. and Stammes, P.: 2003, "A novel approach to the polarisation correction of space-born spectrometers," *J. Geophys. Res.* **108** (D7), doi 10.1029/2002JD002736.

Radiometric corrections -- Ronald van der A (avander@knmi.nl)

- Van der A, R.: 2001, "Recalibration of GOME spectra for the purpose of ozone profile retrieval," Technical Report TR-236, KNMI, De Bilt, The Netherlands.
- Van der A, R.J., Van Oss, R.F., Piters, A.J.M., Fortuin, J.P.F., Meijer, Y.J. and Kelder H.M.: 2002, "Ozone profile retrieval from recalibrated Global Ozone Monitoring Experiment data," *J. Geophys. Res.* **107**, 10.1029.2001JD000696.

Other papers and documents mentioned in this documentation or what may be otherwise relevant for the GomeCal user:

- Balzer, W., Aberle, B., Loyola, D. and Spurr, R.: 1996,
  GOME level 0 to 1 algorithm description,
  ER-TN-DLR-GO-0022, Iss./Rev. 4/A,
  Deutsches Centrum für Luft- und Raumfahrt, Oberpfaffenhofen, Germany.
- Balzer, W. and Loyola, D.: 2000,
  Product Specification Document of the GOME Data Processor,
  ER-PS-DLR-GO-0016, Iss./Rev. 3/D,
  Deutsches Centrum für Luft- und Raumfahrt, Oberpfaffenhofen, Germany.
- Burrows, J.P., Weber, M., Buchwitz, M., Rozanov, V., Ladstätter-Weißenmayer, A., Richter, A., Debeek, R., Hoogen, R., Bramstedt, K., Eichmann, K.-U., Eisinger, M. and Perner, D.: 1999,
  "The Global Ozone Monitoring Experiment (GOME): Mission concept and first results,"
  J. Atmos. Sci. **56**, 151-175.
- Chance K. and Spurr R.J.D.: 1997,
  "Ring effect studies: Rayleigh scattering, including molecular parameters for rotational Raman scattering, and the Frounhofer spectrum,"
  Applied Optics **36**, 5224-5230.
- De Haan, J.F., Bosma, P.B. and Hovenier, J.W.: 1987,
  "The adding method for multiple scattering calculations of polarized light,"
  Astron. Astrophys. **183**, 371-391.
- Fortuin, J.P.F. and Kelder, H.M.: 1998,
  "An ozone climatology based on ozonezonde and satellite measurements,"
  J. Geophys. Res. **103**, 31,709-31,734.
- Hahne A. *et al.*: 1995,
  The Global Ozone Monitoring Experiment (GOME) Users Manual,
  ESA publication SP-1182,
  European Space Agency, Noordwijk, The Netherlands.
- Hahne A.: 1997,
  The Global Ozone Monitoring Experiment},
  ESA publication SP-1212,
  European Space Agency, Noordwijk, The Netherlands.
- Koelemeijer, R.B.A., Stammes, P., Hovenier, J.W. and De Haan, J.F.: 2001,
  "A fast method for retrieval of cloud parameters using oxygen A-band measurements from the Global Ozone Monitoring Experiment,"
  J. Geophys. Res. **106**, 3475--3490.
- Koelemeijer, R.B.A., Stammes, P., Hovenier, J.W. and De Haan, J.F.: 2002,
  "Global distribution of effective cloud fraction and cloud top pressure derived from oxygen A band spectra measured by the Global Ozone Monitoring Experiment: Comparison to ISCCP data,"
  J. Geophys. Res. **107**, 10.1029/2001JD000840.
- Slijkhuis, S. and Loyola, D.: 1999,
  GOME Data Processor extraction software user's manual,
  ER-SUM-DLR-GO-0045, Issue 1,
  Deutsches Centrum für Luft- und Raumfahrt, Oberpfaffenhofen, Germany.
- Spurr, R.J.D.: 2001,
  Linearized Radiative Transfer Theory - A general discrete ordinate approach to the calculation of radiances and analytic weighting functions, with application to atmospheric remote sensing,
  Ph.D. thesis, Eindhoven University of Technology, Eindhoven, The Netherlands.

Some relevant or related web sites:

- Introduction to GOME: Global Ozone Monitoring Experiment (http://www.temis.nl/general/gomeintro.html) onboard the ERS-2 (European Remote Sensing) satellite (http://earth.esa.int/ers/); launched in April 1995 by the European Space Agency (ESA) (http://www.esa.int/)
- GOME home pages at:
  - ESA-ESRIN (http://earth.esa.int/ers/gome/)
  - DLR (http://auc.dfd.dlr.de/GOME/index.html)
  - University of Bremen (http://www-iup.physik.uni-bremen.de/gome/)

Disclaimer

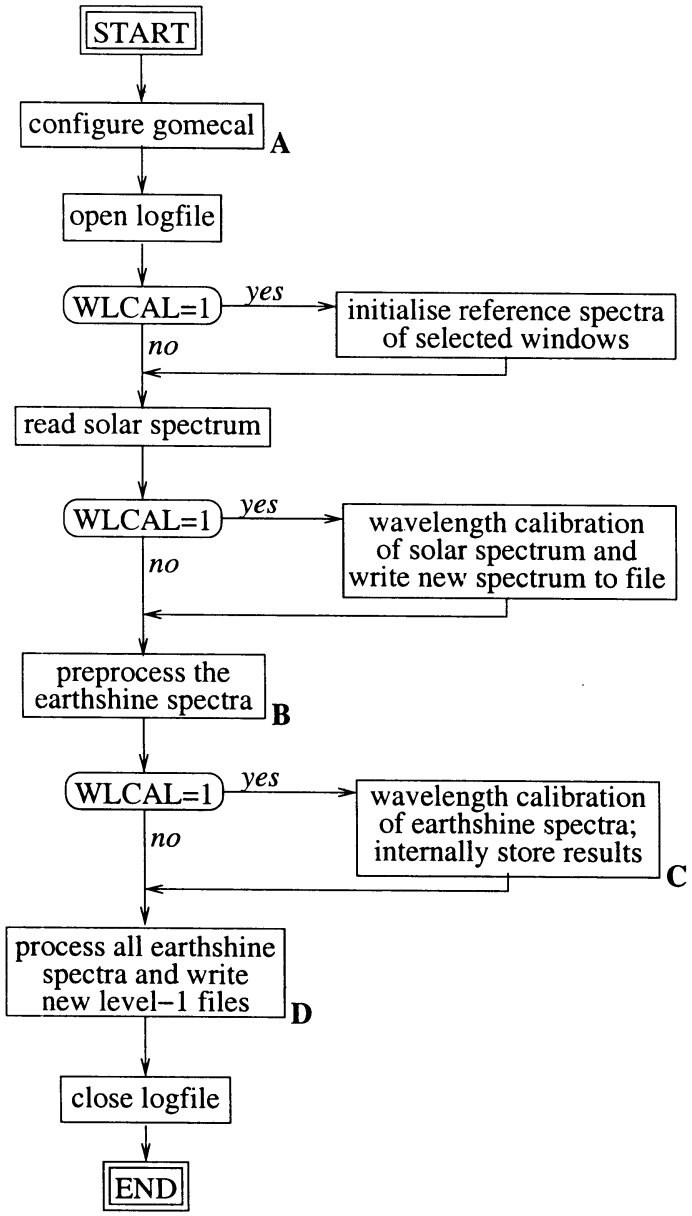
The copyright notice and disclaimer are given in the file `COPYING`, which reads as follows:

Appendix

This Appendix deals with the following subjects:

The structure of the gomecal program

The **gomecal** program is quite complicated: it performs a number of operations, depending on user selected configuration settings, on a set of GDP level-1 spectrum files. Below the structure of **gomecal** is schematically outlined to show in what order the steps are taken. The additional programs of the package are rather straightforward and are therefore not outlined here.

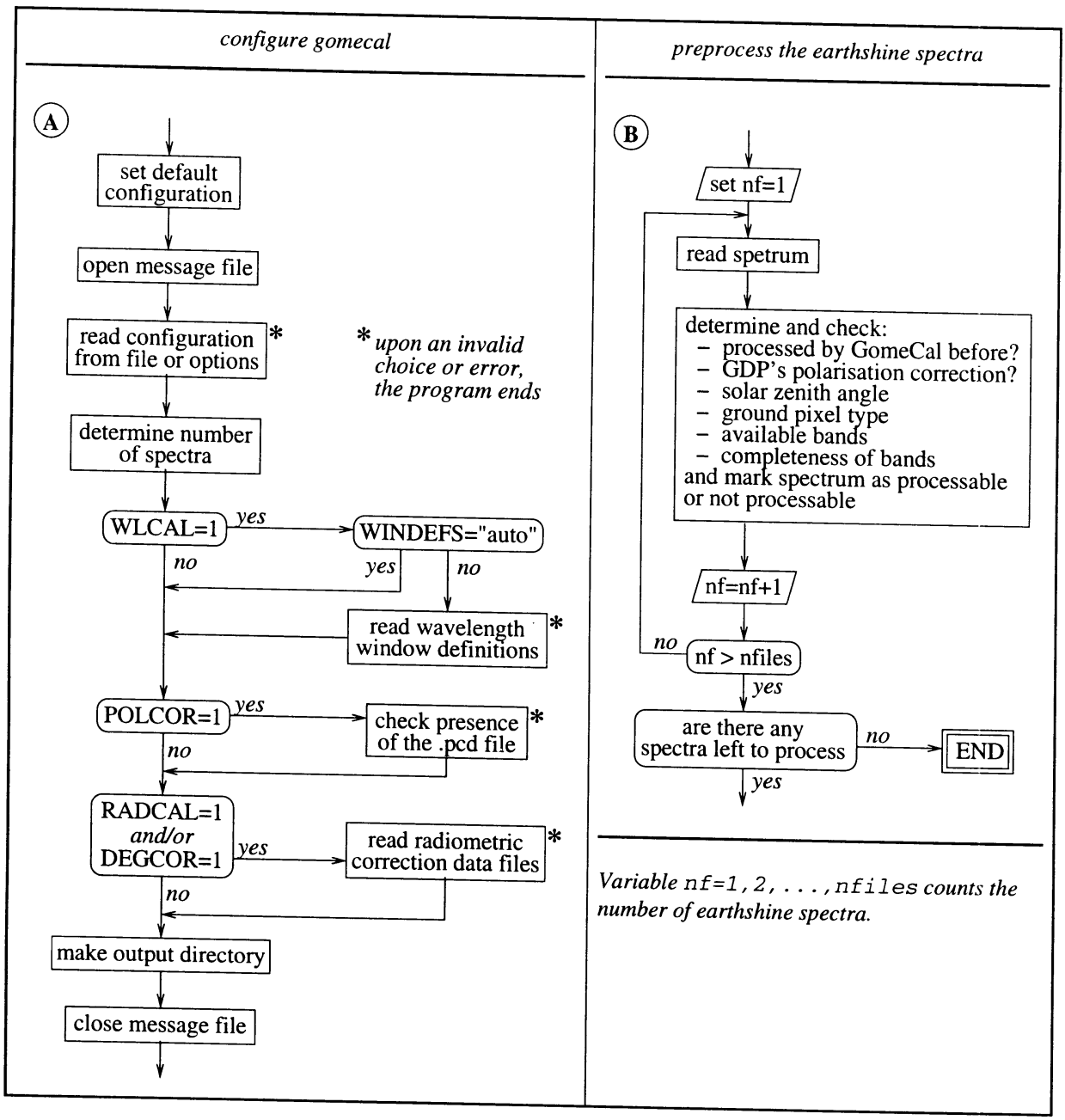


The diagram shows the order in which the different steps are taken in the **gomecal** program; the four blocks marked with a capital letter are given in more detail below. Some terminology is borrowed from the keywords of the configuration file.

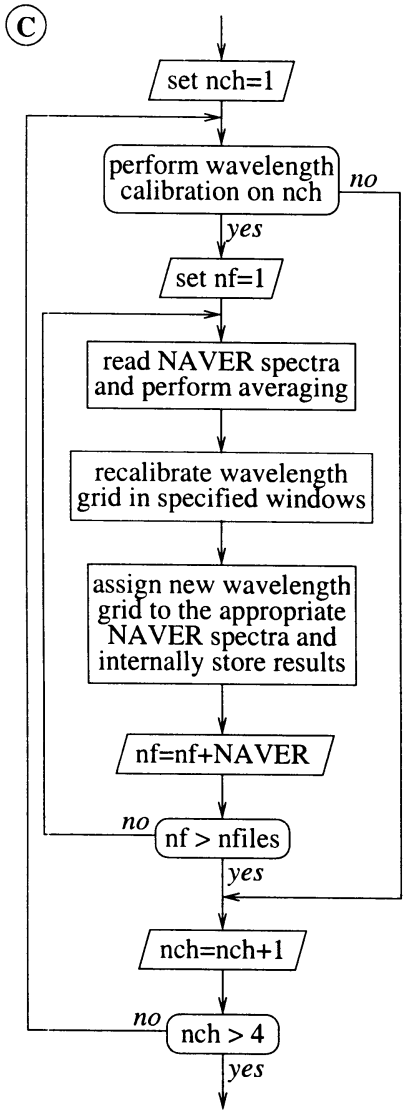
Assumptions for the process in the diagrams:

- > configuration shows no problems
- > a logfile is to be made
- > usage is to process data files: one solar spectrum and `nfiles` earthshine spectra
- > settings `COPYFILES` and `REFLECTIV` are zero
- > data of the full channel is written: `PARTCHAN#=auto` for `#=1,2,3,4`

Preprocessing of the earthshine spectra (checking whether the spectra can be processed) is done to make the subsequent processing more transparent.

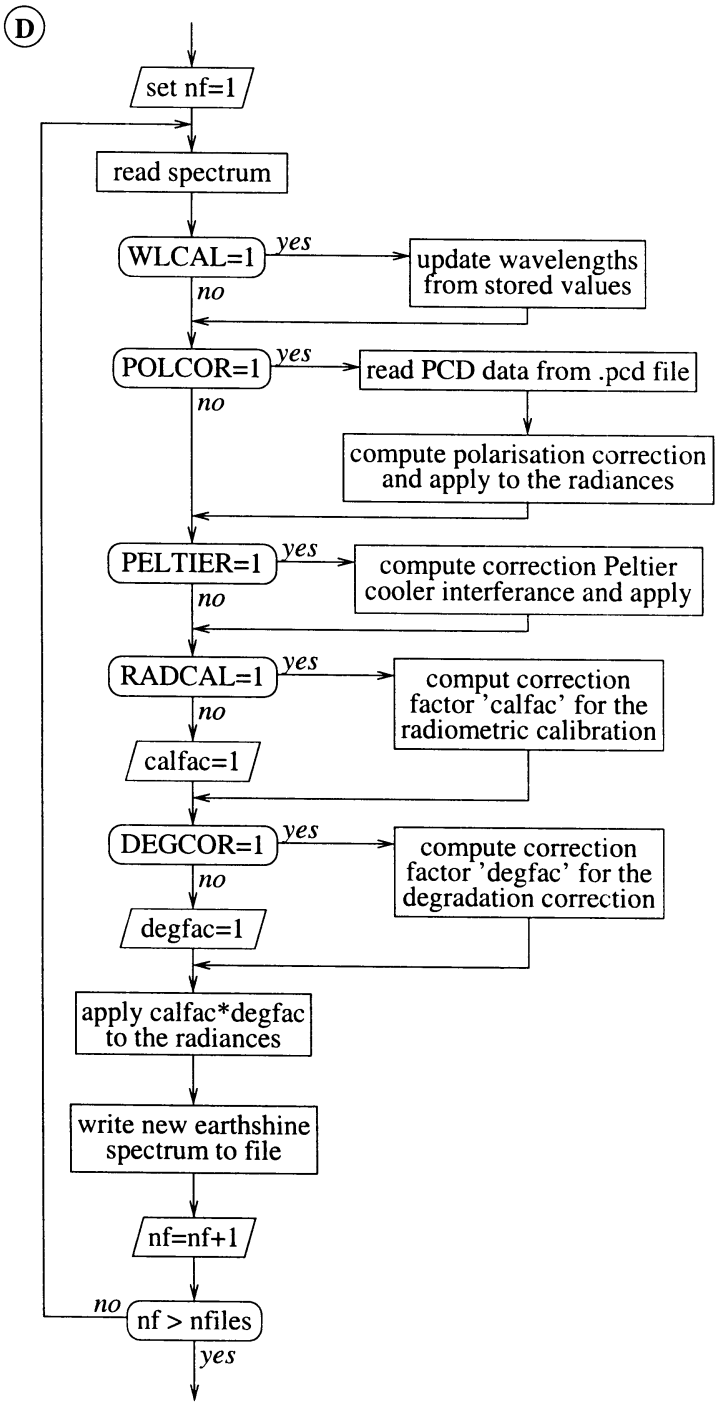

configure gomecal
A
set default configuration
open message file
read configuration from file or options
* upon an invalid choice or error, the program ends
determine number of spectra
WLCAL=1
yes
WINDEFS="auto"
no
yes
no
read wavelength window definitions
*
POLCOR=1
yes
check presence of the .pcd file
*
no
RADCAL=1 and/or DEGCOR=1
yes
read radiometric correction data files
*
no
make output directory
close message file
preprocess the earthshine spectra
B
set nf=1
read spetrum
determine and check:
- processed by GomeCal before?
- GDP's polarisation correction?
- solar zenith angle
- ground pixel type
- available bands
- completeness of bands
and mark spectrum as processable or not processable
nf=nf+1
no
nf > nfiles
yes
are there any spectra left to process
no
END
yes
Variable nf=1,2,...,nfiles counts the number of earthshine spectra.

wavelength calibration of the earthshine spectra



Variable `nch=1, 2, 3, 4` *counts the number of GOME channels and variable* `NAVER` *holds the number of earthshine spectra to average for the wavelength calibration.*

process all the earthshine spectra



The GomeCal directory structure

The directory structure of the GomeCal package has been mentioned when discussing installing the package. In a little more detail the structure is as follows:

`./`	main directory, contains a `README` file with some basic information on getting started, the main `Makefile`, the disclaimer, etc. and after compilation the executables of the programs
`docs/`	contains this documentation in HTML, PostScript and PDF format
`examples/`	contains some example scripts and such; see below
`GCsrc/`	contains the source files of the programs and two text files: `GCsrc/Routines.txt` = overview of which subroutine/function is in which file `GCsrc/Units.txt` = overview of the unit numbers and connected input or output files used
`GCobj/`	will contain the object files after building the programs
`Data/`	contains some additional data files; see below

The `examples/` directory

The `examples/` directory contains some examples regarding the use of the **gomecal** program mentioned in this documentation.

`Makefile`
: To clean up the directory, used when typing `make clean` and such in the main directory.

`getlvl1.csh`
: Example script to extract level-1 spectrum files from the binary GDP orbit files. The script makes the directory for these files, `Lv1`, but does not do more as the call to the extractor has been commented out: the user will have to adapt the paths and such before using the script.

`Lv1/`
: Set of level-1 spectrum files, extracted with a script like `getlvl1.csh`, and ready for use with the configuration file `gomecal.cfg` or the configuration script `gomecal.sh`.

`Lv1d/`
: Set of level-1 spectrum files, extracted with a script like `getlvl1.csh`, where the `-d` option of the GDP_01 extractor was used, instead of `-n`.

`gomecal.xmpl`
: Example configuration file written by **gomecal** when typing `gomecal example`; it shows the default configuration settings.

`gomecal.cfg`
: Ready-to-use configuration file for processing the files in the `Lv1/` subdirectory. Some of the settings in the file are the default settings: these can be omitted, but are left in the file as example. The output goes to a directory called `NewLv1/`.

`NewLv1.cfg/`
: Directory with the output of the `gomecal.cfg` configuration file; the directory is renamed to avoid overwriting when rerunning **gomecal** with `gomecal.cfg`.
NOTE: The files in this directory are made on machine no. 2 in the table in the section on the system requirements. When running **gomecal** on another machine there will no doubt be small differences in the last decimals of the double-precision values given in the logfile due to differences in rounding and machine accuracy. In the results in the level-1 spectrum files there should be for almost all values no difference, because of the low accuracy of the numbers in these files.

`gomecal.msg.cfg`
: Message file produced when running **gomecal** with the `gomecal.cfg` configuration file, renamed to prevent overwriting when rerunning **gomecal** with `gomecal.cfg`.

`gomecal.sh`
: Shell script calling **gomecal** with command line options to get the same result as using the `gomecal.cfg` configuration file.

`windefs.win`
: File showing the default wavelength windows of **gomecal**; it is obtains by typing "`gomecal windefs`" or "`gomecal -Z`".

`complv1.xmpl`
: Example "input file" for the **complv1** program; its output file is `NewLv1.cfg/80502203_0308.cmp`.

The `Data/` directory

This directory contains some data files used for making data blocks for **gomecal**. There are two programs that make these data blocks; the user will probably not ever have to use these two programs. But if needed, these programs can be built by typing `make Data` in the main directory.

`Makefile`
: For building the programs **keydat** and **o3block** in the `Data/` directory by typing "`make Data`" in the main directory. Also used for cleaning-up and such.

`scdegrad.108`
: Latest available properly working data file for the corrections applied by the `-e` and `-f` options of the GDP_01 extractor.

`keydat08.10`
: Latest available KeyData Data file from the GDP level 0-to-1 processing. Some of the data in this file is needed by **gomecal** and the **keydat** program extracts this information. The "08.10" specifies the version of the data file. To save disk space, the file is gzip-ed; extracted it is about 1.2 MB.

`keydat.f`
: Program that extracts the necessary data from the KeyData Data file and stores this in a data block for use **gomecal**. The input file name is asked for, as the KeyData file might get an update in the future. The "KeyData Version" is checked to see it is at least 8.10, the most resent version known when writing this program.
The output of the program is two data block files:

- `keyeta.f`, with some data needed by the polarisation correction;
- `keyresp.f`, with some data needed by the radiometric corrections.

To make the headers of the these two data block files, `keydat.f` reads the `keyeta.txt` and `keyresp.txt` text files.

`o3mean.dat`
: Data file with the Fortuin & Kelder (1998) ozone climatology. The file is read by the **o3block** program. To save disk space, the file is gzip-ed; extracted it is about 35 kB.

`o3block.f`
: Program that converts the ozone climatology in `o3mean.dat`, given at 19 pressure levels, into a block data file for use with **gomecal** containing ozone column densities in Dobson Units (DU). The output of the program is the data block file `o3climcol.f`; to make the header of this file, `o3block.f` reads the `o3climcol.txt` text file.

Note that after producing them, the data block files `keyeta.f`, `keyresp.f` and `o3climcol.f` need to be copied by hand to the `GCsrc/` source directory.

List of all `make` targets

The 'targets' in the main `Makefile` are mentioned at the appropriate places in this documentation. For the sake of completeness, the following table lists all 'targets' and mentions their functionality; the usage being:

```
[gomecal]~/GomeCal> make <target>
```

The targets marked with an asterisk (*) require that the `make` utility understands the VPATH macro; see the section on make-ing the programs

target	*functionality*	*subject*
<empty> *	build the main programs: **gomecal, complv1, refspec** and **wlcomp**	make-ing the programs
`all` *	idem	make-ing the programs
`gomecal` *	build the **gomecal** program	make-ing the programs
`complv1` *	build the **complv1** program	make-ing the programs
`refspec` *	build the **refspec** program	make-ing the programs
`wlcomp` *	build the **wlcomp** program	make-ing the programs
`in.source`	build all four main programs with the object files ending up in the source directory `GCsrc/` rather then in the object directory `GCobj`	make-ing the programs
`Data`	build the programs in the `Data/` directory	The `Data/` directory
`doc`	make a PostScript and PDF version of the documentation for printing	Printing the documentation
`rmdoc`	remove the PostScript and PDF version of the documentation from the `docs/` directory	Printing the documentation
`clean`	clean the package directories from exectables and object files	Cleaning up
`rmbak`	clean the package directories from `*~` and `*.bak` editor backup files	Cleaning up
`exec`	make scripts executable	Shell scripts
`version`	update the `version.txt` version file	Last modification dates