

The EUMETSAT
Network of
Satellite
Application
Facilities



ROM SAF

Radio Occultation Meteorology

ROM SAF CDOP-2

The Radio Occultation Processing Package (ROPP) User Guide Part III: Pre-processor module

Version 9.0

31 December 2016

Danish Meteorological Institute (DMI)
European Centre for Medium-Range Weather Forecasts (ECMWF)
Institut d'Estudis Espacials de Catalunya (IEEC)
Met Office (MetO)

Document Author Table

	Name	Function	Date	Comment
Prepared by:	I. Culverwell	ROM SAF Project Team	31 December 2016	
Reviewed by:	D. Offiler	ROM SAF Project Team	31 December 2016	
Approved by:	K.B. Lauritsen	ROM SAF Project Manager	31 December 2016	

Document Change Record

Issue/Revision	Date	By	Description
Version 2.0	01 Dec 2008	HL	Module-specific documentation for second full release (v2.0)
Version 3.0	01 Jun 2009	HL	Third full release (v3.0)
Version 4.0	01 Nov 2009	HL	Fourth full release (v4.0)
Version 4.1	01 Jun 2010	HL	Updates to fourth full release (v4.1)
Version 5.0	14 Jun 2011	IC	Fifth full release (v5.0)
Version 6.0	31 Oct 2011	IC	Sixth full release (v6.0)
Version 6.1	31 Jan 2013	IC	Updates to sixth full release (v6.1)
Version 7.0	31 Jul 2013	IC	Seventh full release (v7.0)
Version 7.1	31 Dec 2013	IC	Updates to seventh full release (v7.1)
Version 8.0	31 Dec 2014	IC	Eighth full release (v8.0)
Version 8.1	31 Dec 2015	IC	Update to eighth full release (v8.1)
Version 9.0	31 Dec 2016	IC	Ninth full release (v9.0)

ROM SAF

The ROM SAF is the EUMETSAT Satellite Application Facility responsible for operational processing of radio occultation data from the Metop satellites. It delivers bending angle, refractivity, temperature, pressure, and humidity profiles in near-real time and offline for NWP and climate users. The offline profiles are further processed into climate products consisting of gridded monthly zonal means of bending angle, refractivity, temperature, humidity, and geopotential heights together with error descriptions.

The ROM SAF also maintains the Radio Occultation Processing Package (ROPP) which contains software modules that will aid users wishing to process, quality-control and assimilate radio occultation data from any radio occultation mission into NWP and other models.

The ROM SAF Leading Entity is the Danish Meteorological Institute (DMI), with Cooperating Entities: i) European Centre for Medium-Range Weather Forecasts (ECMWF) in Reading, United Kingdom, ii) Institut D'Estudis Espacials de Catalunya (IEEC) in Barcelona, Spain, and iii) Met Office in Exeter, United Kingdom. To get access to our products or to read more about the project please go to <http://www.romsaf.org>.

Intellectual Property Rights

All intellectual property rights of the ROM SAF products belong to EUMETSAT. The use of these products is granted to every interested user, free of charge. If you wish to use these products, EUMETSAT's copyright credit must be shown by displaying the words "copyright (year) EUMETSAT" on each of the products used.

Contents

1	Introduction	1
1.1	Purpose of this document	1
1.2	ROPP	1
1.3	User documentation	3
	References	3
2	GNSS-RO data processing	4
2.1	Radio occultation geometry	4
2.2	Radio occultation observations	5
2.3	Bending angle	6
2.3.1	Preprocessing amplitude and excess phase data	6
2.3.2	Geometric optics	8
2.3.3	Wave optics — CT2 algorithm	9
2.4	Ionospheric correction	11
2.5	Abel inversion	12
2.6	Refractivity	12
2.7	Dry temperature	13
2.8	Summary	13
	References	14
3	ROPP Pre-processor: amplitude and phase to bending angle	15
3.1	ROPP occultation tool	17
3.1.1	Implementation	17
3.1.2	Code organisation	19
3.2	Configuration options	20
3.3	Preprocessing ropp_pp_preprocess	20
3.3.1	Model excess phase	21
3.3.2	Raw sampling and open loop data pre-processing	22
3.3.3	Data cutoff	23
3.3.4	Compute spectra	23
3.3.5	Correction of L2 data	23
3.4	Geometric optics ropp_pp_bending_angle_go	24
3.5	Wave optics (CT2) ropp_pp_bending_angle_wo, ropp_pp_DCT	24
3.6	Spatial spectra ropp_pp_spectra_tool, ropp_pp_spectra	26
3.7	Processing GRAS raw sampling data	26
	References	26

4	ROPP Pre-processor: L1 and L2 bending angle to refractivity and dry temperature	29
4.1	ROPP inversion tool	30
4.1.1	Implementation	31
4.1.2	Code organisation	31
4.2	Configuration options	33
4.3	Observation data	33
4.3.1	Data order	35
4.3.2	Standard impact parameter levels <code>ropp_pp_merge_profile</code>	35
4.4	Linear combination <code>ropp_pp_linear_combination</code>	35
4.5	Ionospheric correction	36
4.5.1	Climatology bending angle data	36
4.5.2	Statistical optimization <code>ropp_pp_ionospheric_correction</code>	38
4.6	Inversion	40
4.6.1	Climatology bending angle data	41
4.6.2	Inverse Abel algorithm	41
4.6.3	Hydrostatic integration	43
	References	44
A	<code>ropp_utils</code> library	45
A.1	Missing data values	45
A.2	<code>ropp_messages</code>	45
A.3	Unitconvert	46
A.4	Coordinates	46
A.5	Datetime	46
A.6	Geodesy	46
A.7	Arrays	47
A.8	Misc	47
A.8.1	<code>typeSizes</code>	47
B	Installing and using ROPP	48
B.1	Software requirements	48
B.2	Software release notes	48
B.3	Third-party packages	48
B.3.1	NetCDF	49
B.3.2	BUFR (optional)	49
B.3.3	GRIB_API (optional)	50
B.3.4	netCDF4/HDF5 (optional)	50
B.3.5	RoboDoc (optional)	51
B.3.6	autoconf and automake (optional)	51
B.4	BUILDPACK script	51
B.5	Building and installing ROPP manually	52
B.5.1	Unpacking	53

B.5.2	Configuring	53
B.5.3	Compiling	54
B.5.4	Installing	55
B.5.5	Cleaning up	55
B.6	Linking	55
B.7	Testing	56
B.7.1	ropp_utils	56
B.7.2	ropp_io	56
B.7.3	ropp_pp	58
B.7.4	ropp_apps	58
B.7.5	ropp_fm	58
B.7.6	ropp_1dvar	58
B.8	Troubleshooting	59
C	ropp_pp program files	60
D	ROPP extra data	63
D.1	ropp_io_addvar	63
D.2	PPDiag	64
D.3	ropp_fm_bg2ro	64
D.4	VarDiag	64
E	ROPP user documentation	66
F	Acronyms and abbreviations	69
G	Definitions	72
H	Authors	73
I	Copyrights	74

1 Introduction

1.1 Purpose of this document

This document provides a User Guide for the Pre-processor module of the Radio Occultation Processing Package (ROPP). The Pre-processor is designed to compute ionospheric corrected bending angle and refractivity profiles either from excess phase or L1 and L2 channel bending angle data measured during a radio occultation.

The ROPP User Guide Part I (2016d) provides an overview of the generic ROPP data format and software to read and write radio occultation data provided as part of ROPP. The ROPP User Guide Part IIa (2016c) describes the ROPP forward models, while User Guide Part IIb (2016a) provides details of the ROPP retrieval software for users who wish to perform 1D–Var retrievals using refractivity or bending angle data. The ROPP User Guide Part IV (2016b) describes the ROPP applications module, which currently contains tools to calculate tropopause height and planetary boundary layer height from profiles of radio occultation data.

An overview of the ROPP modules and the installation, build and test procedures for ROPP are provided in the appendices. Detailed build and install instructions are contained in the release notes of the individual ROPP software modules.

1.2 ROPP

The aim of ROPP is

... to provide Users with a comprehensive software package, containing all necessary functionality to pre-process RO data from Level 1a (Phase), Level 1b (Bending Angle) or Level 2 (Refractivity) files, plus RO-specific components to assist with the assimilation of these data in NWP systems.

ROPP is a collection of software modules (provided as source code), supporting data files and documentation, which aids users wishing to assimilate radio occultation data into their NWP models. As far as is practical, the software is generic, in that it can handle any GNSS–LEO configuration radio occultation mission (CHAMP, GRACE, SAC–C, GRAS, COSMIC, etc).

The software is distributed in the form of a source code library written in Fortran 90. ROPP is implemented using Fortran modules and derived types, enabling the use of object oriented techniques such as the overloading of routines. The software is split into several modules. Figure 1.1 illustrates the inter-relationships between each module. Users may wish to integrate a subset of ROPP code into their own software applications, individually linking modules to their own code. These users may not require the

complete ROPP distribution package. Alternatively, users may wish to use the executable tools provided as part of each module as stand-alone applications for RO data processing. These users should download the complete ROPP release.

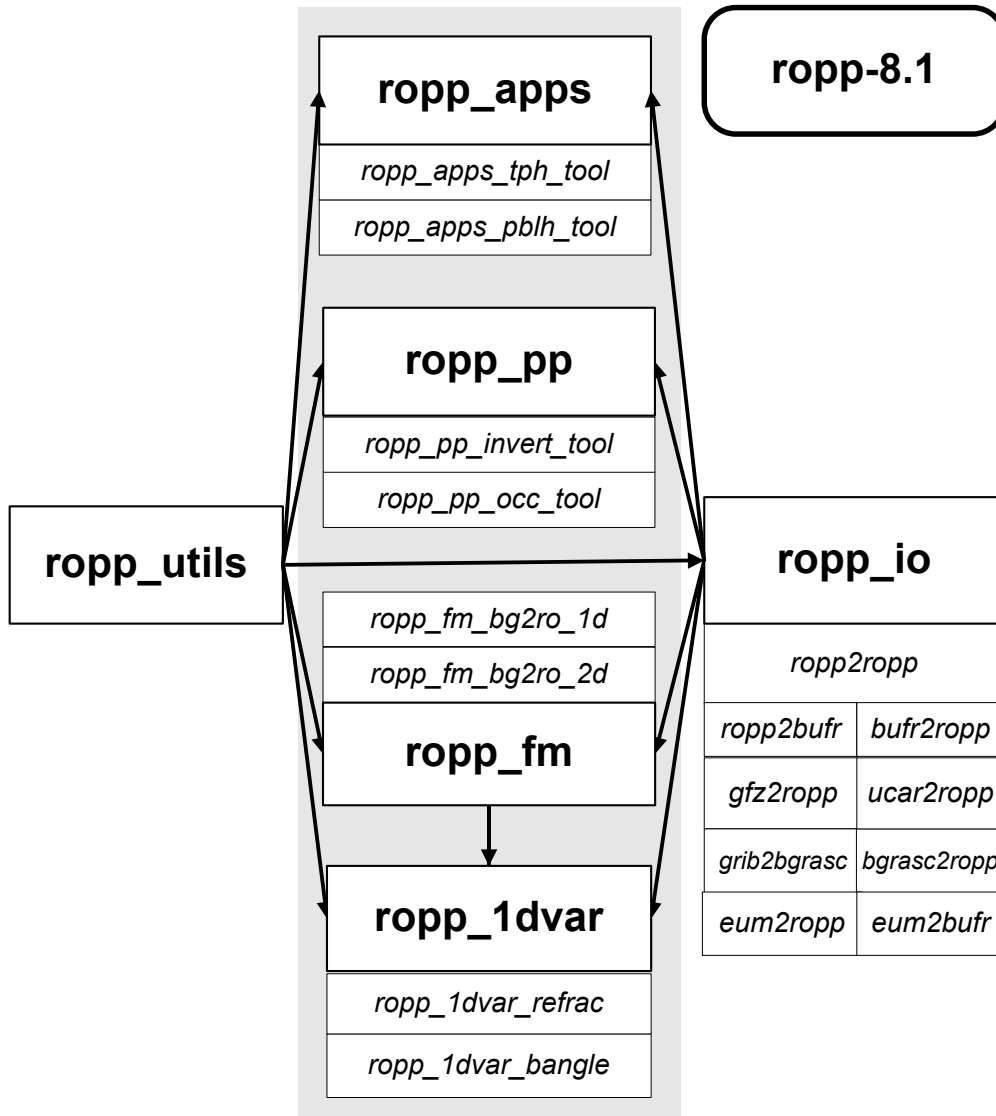


Figure 1.1: The **modules** and **tools** within ROPP-8.1. The module at the head of an arrow depends directly on the module at its tail.

ROPP contains support for a generic data format for radio occultation data (*ropp_io*), pre-processor routines (*ropp_pp*), application routines (*ropp_apps*), one- and two-dimensional forward models (*ropp_fm*), tools for quality control (immersed within the other modules, principally *ropp_1dvar*) and routines for the implementation of 1D-Var retrievals (*ropp_1dvar*). Utility routines used by some or all of the ROPP modules are provided in an additional module (*ropp_utils*). This structure (Figure 1.1) reflects the various degrees of interdependence of the difference ROPP modules. For example, the subroutines and functions in *ropp_io* and *ropp_fm* modules are mutually independent, whereas routines in *ropp_1dvar* depend on *ropp_fm*. Sample standalone implementations of *ropp_pp*, *ropp_fm* and *ropp_1dvar* (which then require *ropp_io* for file interfaces, reading and writing data) are provided with those modules and documented in

the relevant User Guides.

1.3 User documentation

A full list of user documentation is provided in Tables E.1, E.2 and E.3. These documents are available via the ROM SAF website at <http://www.romsaf.org>.

The ROPP distribution website has a Release Notes file in the root directory which provides a 'Quick Start' guide to the package. This should be read before downloading the package files. Detailed build and install instructions are contained in the release notes of the individual ROPP software modules.

This ROPP Pre-processor User Guide provides documentation of the pre-processor software provided with ROPP. An overview on how to install, build and use it, along with a description of the ROPP tools used by other ROPP modules are also provided.

Module-specific user guides for the input/output (ROM SAF, 2016d), forward model (ROM SAF, 2016c), 1D-Var (ROM SAF, 2016a) and applications (ROM SAF, 2016b) modules describe the algorithms and routines used in those modules. These provide the necessary background and descriptions of the ROPP software for users to process read and write radio occultation data and retrieve atmospheric variables from bending angle or refractivity profiles and background data.

More detailed reference manuals are also available for each module for users wishing to write their own interfaces to the ROPP routines, or to modify the ROPP code. These are provided in the associated module distribution files.

Further documentation can be downloaded from the ROPP section of the ROM SAF web site <http://www.romsaf.org>. The full user documentation set is listed in Table E.1.

In addition to these PDF documents, most of the stand-alone application programs have Unix-style 'man page' help files which are installed during the build procedures. All such programs have summary help information which is available by running the command with the `-h` switch.

Any comments on the ROPP software should in the first instance be raised via the ROM SAF Helpdesk at <http://www.romsaf.org>.

References

ROM SAF, The Radio Occultation Processing Package (ROPP) User Guide. Part IIb: 1D-Var module, SAF/ROM/METO/UG/ROPP/003b, Version 9.0, 2016a.

ROM SAF, The Radio Occultation Processing Package (ROPP) User Guide. Part IV: Applications module, SAF/ROM/METO/UG/ROPP/005, Version 9.0, 2016b.

ROM SAF, The Radio Occultation Processing Package (ROPP) User Guide. Part IIa: Forward model module, SAF/ROM/METO/UG/ROPP/003a, Version 9.0, 2016c.

ROM SAF, The Radio Occultation Processing Package (ROPP) User Guide. Part I: Input/Output module,
SAF/ROM/METO/UG/ROPP/002, Version 9.0, 2016d.

2 GNSS-RO data processing

The ROPP pre-processor module `ropp_pp` provided as part of the ROPP software enables users to process radio occultation data to derive atmospheric bending angle and refractivity profiles which may be used for NWP or climate applications. Additionally, the module also enables the derivation of the so-called dry temperature. Several distinct stages are required to process radio occultation data from excess phase measurements to refractivity and dry temperature. These stages are described in more detail in this Section.

1. Retrieve L1 and L2 channel bending angles as a function of impact parameter from excess phase measurements with time and orbit position and velocity information.
2. Retrieve atmospheric bending angle from L1 and L2 channel bending angles by removing the ionospheric contribution.
3. Retrieve atmospheric refractivity profile from the ionospheric corrected bending angle by inversion of the Abel integral, taking account for the infinite upper boundary condition.
4. Retrieve atmospheric dry temperature from refractivity assuming hydrostatic equilibrium and ignoring water vapour.

2.1 Radio occultation geometry

A detailed review of GNSS-RO measurements was provided by Kursinski et al. (1997). The typical radio occultation geometry is sketched in Figure 2.1. A ray passing from a GNSS to a LEO satellite through the atmosphere is refracted due to its vertical gradient of density. Under the assumption of spherical symmetry, the perpendicular distances from the origin to the ray asymptote of the ray at the GNSS satellite p_G and LEO satellite p_L are equal. This distance is termed the impact parameter p . The refraction by the atmosphere at a point on the ray is then described by Bouguer's Law, which states (e.g., Born and Wolf, 1980)

$$p = n(r)r \sin \phi = \text{constant} \quad (2.1)$$

Here, n is the refractive index of the atmosphere along the ray path, r is the radial distance from the symmetry centre (also referred to as centre of curvature) to a point on the ray, and ϕ denotes the angle between the position vector and the ray tangent at the same point (Figure 2.1). If the tangent point is defined as that point on the ray where the ray tangent is perpendicular to the position vector ($\phi = 90^\circ$), we obtain $\sin \phi = 1$, and therefore

$$p = n(r_t)r_t \quad (2.2)$$

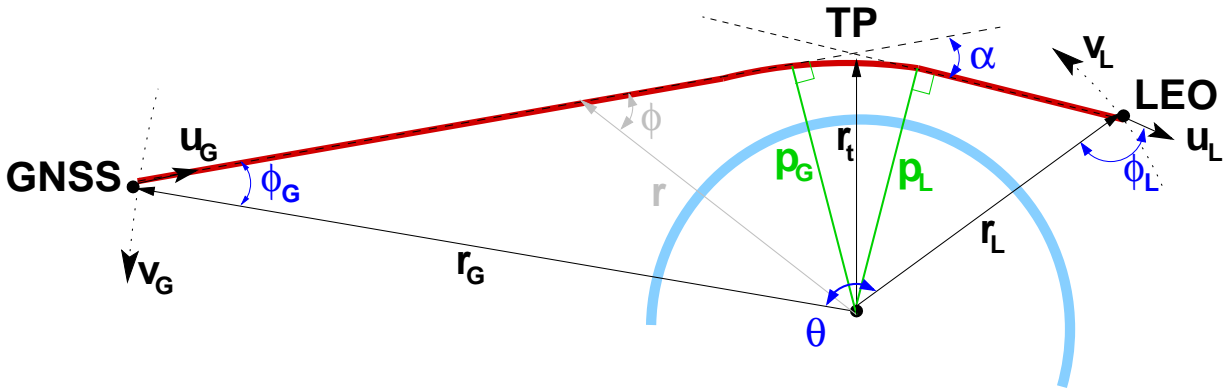


Figure 2.1: Radio occultation geometry. Shown are the bending angle α , the GNSS and LEO side impact parameters (p_G and p_L), the GNSS and LEO coordinate vectors (\mathbf{r}_G , \mathbf{r}_L), the ray path (solid red line) and the satellite side asymptotes of the ray path (dashed). Radius r_t shows the radial distance between the centre of curvature and the ray tangent point.

with r_t being the distance between the centre of curvature and the tangent point. This allows the calculation of the impact parameter from given refractivity and geometric altitude

$$p = (h + R_c) \cdot n(h + R_c) \quad (2.3)$$

where R_c is the local radius of curvature of the Earth (ellipsoid) at the tangent point and h is the geometric height with respect to the ellipsoid.

Note that the two satellite side impact parameters p_G and p_L will, in general, not be equal if the spherical symmetry assumption is not met. In addition, the impact parameter p calculated from Equation (2.2) at some hypothetical tangent point is not necessarily related to the satellite side impact parameters in the non-symmetrical case.

2.2 Radio occultation observations

A measured signal at time t can be described as a wave field with amplitude $A(t)$, wave vector $k = 2\pi f/c$ and phase path (optical path length) $\Psi(t)$

$$u(t) = A(t) \exp(ik\Psi(t)) \quad (2.4)$$

The amplitude and phase of the received GNSS signal are measured during the period of a radio occultation event. Each GNSS satellite transmits at two L-band frequencies, or channels, $f_{G1} = 1575.42$ MHz (L1) and $f_{G2} = 1227.6$ MHz (L2). Comparison of the L1 and L2 measurements enables the contribution by the ionosphere and lower atmosphere to be separated (Section 2.4).

Given precise positions and velocities of the GNSS and LEO satellites, it is possible to estimate the phase path *in vacuo* Ψ_0 for the GNSS–LEO straight line. The contribution to the measured phase path from refraction of the ray as it passes through the atmosphere is then determined as the excess phase path $\Delta S(t) = \Psi(t) - \Psi_0(t)$. This gives the difference between the accumulated optical path length measured at

the LEO and that which would be observed in a vacuum with the same observation geometry.

The raw GNSS-RO and satellite orbit data are processed in near-real time by satellite data providers. The `ropp_pp` module provides software which enables users to process L1 and L2 excess phase or bending angle data to derive ionospheric corrected bending angle and refractivity profiles. The software is generic in that it is appropriate for processing user data from any GNSS-LEO configuration radio occultation mission.

2.3 Bending angle

Two distinct approaches may be used to determine bending angle profiles as a function of impact parameter from excess phase measurements of a GNSS signal. Assuming local spherical symmetry, the excess phase can be combined with satellite position and velocity data to determine the bending angle α at a given impact parameter p from the geometry at a given time during an occultation event (e.g. Kursinski et al., 1997). This geometrical optics approach is relatively simple, but depends on the assumption that only one ray is observed at a given moment of time. For periods when a ray passes through regions where high refractivity gradients occur above a region of lower gradients, multiple signal paths may interfere. In this multipath case, the geometric optics approach results in non-physical profiles. An alternative wave optics approach, analysing the Fourier transform of the received signal is more appropriate (e.g. Gorbunov and Lauritsen, 2004). In regions where multiple rays are observed, the wave field is then approximated as a sum of the fields corresponding to different interfering rays.

$$u(t) = \sum_j u_j(t) = \sum_j A_j(t) \exp(ik\Psi_j(t)) \quad (2.5)$$

The `ropp_pp` module includes algorithms to compute bending angles from excess phase and amplitude measurements using geometric optics and wave optics. These routines are developed from software written by Michael Gorbunov as a ROM SAF visiting scientist.

2.3.1 Preprocessing amplitude and excess phase data

Processing of radio occultation data requires filtering and quality control to handle noise reduction, data gaps and corrupted data, before computing bending angle profiles. This takes care of the loss of signal lock or data corruption in multipath regions, for example. The preprocessing implemented in `ropp_pp` follows that described by Gorbunov et al. (2005).

1. Compute model excess phase from the MSIS refractivity climatology (Hedin, 1991).
2. Mission-specific preprocessing.
 - CHAMP channel L2 data contain a high level of instrument noise, so L2 amplitude is simply replaced by L1 amplitude data.
 - COSMIC and GRAS data for a ray height below 8-11 km are measured in open loop and raw sampling mode respectively (identified by a lost carrier flag in the input data). Retrieval of excess phase measurements in this region requires removal of the pseudo-random 0 or π which is added to

each data sample during the measurement. This is achieved using internally derived navigation bits, or by reference to an external source of navigation bit data.

3. Data cutoff to reject data where signal-tracking errors are very large. The cutoff criteria are based on the measured amplitude and a rough estimate of the bending angle and impact parameter profile computed by geometric optics with strong smoothing.

- The data starting from the time when the amplitude falls below a specified fraction (e.g. 0.1) of the peak measured amplitude are rejected.

- The data starting from the time when the smooth bending angle profile above a specified impact height and straight line tangent height reaches a critical value (e.g. 0.05) are rejected.

4. Correct channel L2 data since signal tracking errors for L2 significantly exceed those in channel L1 and L2 data are much noisier.

- Replace L2 amplitude by a smoothed amplitude computed from MSIS climatology using geometric optics.

- Apply a radioholographic filter in the time domain and filter L2 phase data (Gorbunov and Lauritsen, 2006). The radio occultation measurements are multiplied by a reference signal, which compresses the signal spectrum. A Fourier filter is applied to the narrow-banded signal and its spectrum finally decompressed.

- Analyse local spatial spectra $v_{1,2}(t, \eta)$ of L1 and L2 signals (radiooptic analysis).

- Compute the mean value of the impact parameter

$$\bar{p}_{1,2}(t) = \frac{\int |v_{1,2}(t, \eta)|^2 p(t, \eta) d\eta}{\int |v_{1,2}(t, \eta)|^2 d\eta} \quad (2.6)$$

- Compute the spectral width

$$\delta p_{1,2}(t) = \frac{\int |v_{1,2}(t, \eta)|^2 (p(t, \eta) - \bar{p}_{1,2}(t))^2 d\eta}{\int |v_{1,2}(t, \eta)|^2 d\eta} \quad (2.7)$$

- Compute a penalty function for L2 data as

$$W(t) = 1 - \exp \left[- \left(\frac{|\bar{p}_1(t) - \bar{p}_2(t)|}{\Delta p_A} + \frac{\delta p_2(t)}{\Delta p_D} \right)^2 \right] \quad (2.8)$$

where parameters $\Delta p_A = 0.20$ km and $\Delta p_D = 0.15$ km.

- Compute difference of smoothed excess phase paths $\Delta \bar{\Phi}(t) = \bar{\Phi}_2(t) - \bar{\Phi}_1(t)$.

- Correct L2 phase data as a linear combination of L1 and L2 data as

$$\Phi_2^{\text{cor}}(t_i) = \sum_{j=1}^{i-1} D_j \Psi_2^{\text{cor}} \quad (2.9)$$

$$D_i \Phi_2^{\text{cor}} = D_i \Phi_2 (1 - W(t_i)) + (D_i \Phi_1 + D_i \Delta \bar{\Phi}) W(t_i) \quad (2.10)$$

$$D_i F = F(t_{i+1} - t_i) \quad (2.11)$$

- Correct L2 amplitude data as

$$A_2^{\text{cor}}(t) = A_2(t)(1 - W(t)) \quad (2.12)$$

2.3.2 Geometric optics

In the geometric optics approximation it is assumed that only one ray is observed at each time t , uniquely defined by the impact parameter a .

The excess phase path $\Delta s(t) = \Psi(t) - \Psi_0(t)$ gives the difference between the accumulated phase path (in m) measured at the LEO and that which would be observed in a vacuum with the same observation geometry.

$$\Delta s(t) = \frac{c}{f_G} \int_{t_0}^t \left(f_L^{(0)}(t') - f_L(t') \right) dt' \quad (2.13)$$

The atmosphere-free Doppler frequency $f_L^{(0)}$ is the frequency which would be observed at the LEO in the straight-line approximation (no atmospheric bending of the GNSS signal). From Vorob'ev and Krasil'nikova (1994), this is related to the LEO and GNSS velocities \mathbf{v}_L and \mathbf{v}_G and the signal frequency emitted at the GNSS f_G as

$$f_L^{(0)} = f_G \frac{c - \mathbf{v}_L \cdot \mathbf{u}^{(0)}}{c - \mathbf{v}_G \cdot \mathbf{u}^{(0)}} \sqrt{\frac{c^2 - v_G^2}{c^2 - v_L^2}} \quad (2.14)$$

where $\mathbf{u}^{(0)} = \widehat{\mathbf{r}_L - \mathbf{r}_G}$ is the unit vector in the straight-line GNSS–LEO direction. The frequency observed at the LEO in the presence of the atmosphere with bending is given by

$$f_L = f_G \frac{c - \mathbf{v}_L \cdot \mathbf{u}_L}{c - \mathbf{v}_G \cdot \mathbf{u}_G} \sqrt{\frac{c^2 - v_G^2}{c^2 - v_L^2}} \quad (2.15)$$

where \mathbf{u}_L and \mathbf{u}_G are the unknown ray directions (unit vectors) at the LEO and GNSS positions at time t . The bending angle α is the angle between ray directions \mathbf{u}_L and \mathbf{u}_G (as shown in Figure 2.1):

$$\cos \alpha = \mathbf{u}_L \cdot \mathbf{u}_G \quad (2.16)$$

Since the satellite velocities are much less than c , the relative Doppler shift can be approximated by

$$d = \frac{f_L - f_G}{f_G} = \frac{f_L}{f_G} - 1 = \frac{c - \mathbf{v}_L \cdot \mathbf{u}_L}{c - \mathbf{v}_G \cdot \mathbf{u}_G} - 1 \quad (2.17)$$

The relative Doppler shift is also related to Δs as

$$d = \frac{f_L - f_G}{f_G} = \frac{f_L}{f_G} - 1 = \frac{f_L^{(0)}}{f_G} - \frac{1}{c} \frac{d\Delta s}{dt} - 1 \quad (2.18)$$

Thus the bending angle α as a function of impact parameter p can be found in the geometric optics approximation from measurements of \mathbf{v}_L , \mathbf{r}_L , \mathbf{r}_G , \mathbf{v}_G and excess phase path Δs at each sample time as

follows:

1. Compute straight line path direction $\mathbf{u}^{(0)} = \widehat{\mathbf{r}_L - \mathbf{r}_G}$
2. Compute atmosphere-free relative Doppler shift $d^{(0)} = \frac{c - \mathbf{v}_L \cdot \mathbf{u}^{(0)}}{c - \mathbf{v}_G \cdot \mathbf{u}^{(0)}} - 1$
3. Compute observed relative Doppler shift from the time derivative of excess phase $d = d^{(0)} - \frac{1}{c} \frac{d\Delta s}{dt}$
4. Find ray directions \mathbf{u}_G and \mathbf{u}_L by iteratively solving set of equations:

$$\frac{c - \mathbf{v}_L \cdot \mathbf{u}_L}{c - \mathbf{v}_G \cdot \mathbf{u}_G} - 1 = d \quad (2.19)$$

$$\mathbf{r}_L \times \mathbf{u}_L - \mathbf{r}_G \times \mathbf{u}_G = \mathbf{0} \quad (2.20)$$

$$\mathbf{u}_L \cdot \mathbf{u}_L = 1 \quad (2.21)$$

$$\mathbf{u}_G \cdot \mathbf{u}_G = 1 \quad (2.22)$$

where \mathbf{r}_G and \mathbf{r}_L are expressed relative to the centre of local curvature of the Earth surface at the occultation point. (The second of these equations results from Bouger's Law, with the assumption that the refractive index is unity at the position of both satellites.)

5. Compute the bending angle α as the angle between \mathbf{u}_L and \mathbf{u}_G .
6. Find the angle ϕ_L between \mathbf{r}_L and \mathbf{u}_L to calculate the impact parameter $p = |r_L| \sin \phi_L$ (again assuming $n_L = 1$).

2.3.3 Wave optics — CT2 algorithm

The geometrical optics approximation is only valid in regions where the receiver detects only one ray. Interpretation of measurements where multiple rays are observed is more complex and requires an alternative approach to reconstruct the ray structure of wave fields. The `ropp_pp` module includes an implementation of a canonical transform (CT2) algorithm (Gorbunov and Lauritsen, 2004). This method is a generalisation of canonical transform and Full Spectrum Inversion (FSI) techniques (Jensen et al., 2003).

Wave optics processing aims to transform the measured wave field $u(t)$ to a coordinate system in which each ray is a single-valued function of the new coordinate. The transform of the wave field is achieved by applying a Fourier Integral Operator (FIO) $\hat{\Phi}_2$. In general, $\hat{\Phi}_2$ projects a source wave field $u(y)$ with momentum η to a new coordinate frame z with momentum ξ . By inverse canonical transform, the transformed ray field can then be mapped back to the representation of the source coordinate and momentum.

$$\hat{\Phi}_2 u(z) = \sqrt{\frac{-ik}{2\pi}} \int a_2(z, t) \exp(ikS_2(z, t)) u(t) dt \quad (2.23)$$

where $a_2(z, t)$ is the amplitude function and $S_2(z, t)$ is the phase function of the operator (Gorbunov and Lauritsen, 2004). The phase function gives the phase path along the ray between points (t) and (z) , defined by

$$dS_2 = \xi dz - \eta dt \quad (2.24)$$

The amplitude function is derived from energy conservation as

$$a_2(z, t) = \sqrt{\left| \frac{\partial^2 S_2(z, t)}{\partial z \partial t} \right|} \quad (2.25)$$

Each ray in a spherically symmetric medium is uniquely defined by the impact parameter p . By transformation of the wave field from a projection as a function of time t to a coordinate given by the impact parameter p (i.e. choose $z = p$), then all rays will be uniquely represented. The phase function $S_2(p, t)$ is derived by integrating $\eta(p, t) = d\Psi/dt = \dot{\Psi}$ over t for a fixed value of p .

$$\eta(p, t) = \mathbf{v}_L \cdot \mathbf{u}_L - \mathbf{v}_G \cdot \mathbf{u}_G = p\dot{\theta} + \frac{v_G}{r_G} \sqrt{r_G^2 - p^2} + \frac{v_L}{r_L} \sqrt{r_L^2 - p^2} \quad (2.26)$$

$$S_2(p, t) = -p\theta - \sqrt{r_G^2 - p^2} + a \cos^{-1} \left(\frac{p}{r_G} \right) - \sqrt{r_L^2 - p^2} + p \cos^{-1} \left(\frac{p}{r_L} \right) \quad (2.27)$$

The amplitude function $a_2(p, t)$ is

$$a_2(p, t) = \sqrt{\left| \mu(p, t) \frac{\partial^2 S_2(p, t)}{\partial p \partial t} \right|} \quad (2.28)$$

where $\mu(p, t)$ is the measure density

$$\mu = \sqrt{r_L^2 - p^2} \sqrt{r_G^2 - p^2} \frac{r_L r_G}{p} \sin \theta \left(\dot{\theta} - \frac{v_G}{r_G} \frac{p}{\sqrt{r_G^2 - p^2}} - \frac{v_L}{r_L} \frac{p}{\sqrt{r_L^2 - p^2}} \right) \quad (2.29)$$

For the ideal case of circular observation geometry, the FIO is reduced to a Fourier transform (Gorbunov and Lauritsen, 2004) and its numerical implementation can be very efficient. To reduce the FIO to the Fourier transform and enable efficient implementation for more arbitrary observation geometry, it is necessary to define the approximate value of the impact parameter \tilde{p} . The approximate impact parameter is itself a function of time t and momentum σ .

$$\tilde{p}(t, \sigma) = p_0(t) + \frac{\partial p_0}{\partial \sigma} (\sigma - \sigma_0(t)) = f(t) + \frac{\partial p_0}{\partial \sigma} \sigma \quad (2.30)$$

$$f(t) = p_0(t) - \left(\dot{\theta} - \frac{v_G}{r_G} \frac{p_0}{\sqrt{r_G^2 - p_0^2}} - \frac{v_L}{r_L} \frac{p_0}{\sqrt{r_L^2 - p_0^2}} \right)^{-1} \sigma_0(t) \quad (2.31)$$

where $\sigma_0(t)$ is a smooth model of the Doppler frequency shift and $p_0(t) = p(t, \eta_0(t))$.

A new coordinate $Y = Y(t)$ is now defined with momentum η , such that

$$dY = \left(\frac{\partial p_0}{\partial \sigma} \right)^{-1} dt \quad (2.32)$$

$$\eta = \frac{\partial p_0}{\partial \sigma} \sigma \quad (2.33)$$

$$\tilde{p} = f(Y) + \eta \quad (2.34)$$

$$\xi = -Y \quad (2.35)$$

Then,

$$S_2(\tilde{p}, Y) = \xi d\tilde{p} - \eta dY = -\tilde{p}Y + \int_0^Y f(Y') dY' \quad (2.36)$$

and

$$a_2(\tilde{p}, Y) = \left(\sqrt{r_L^2 - \tilde{p}^2} \sqrt{r_G^2 - \tilde{p}^2} \frac{r_L r_G \sin \theta}{\tilde{p}} \right)^{1/2} \quad (2.37)$$

The amplitude function $a_2(\tilde{p}, Y)$ is replaced by $a_2(\tilde{p}, Y_s(\tilde{p}))$ where $Y_s = -\xi$ is the stationary phase point of the Fourier integral. The resulting FIO is then given by

$$\hat{\Psi}_2 u(\tilde{p}) = \sqrt{\frac{-ik}{2\pi}} a_2(\tilde{p}, Y_s(\tilde{p})) \int \exp(-ik\tilde{p}Y) \exp(ik\tilde{Y}_0 f(Y') dY') u(Y) dY \quad (2.38)$$

which is the composition of a reference signal $\exp(ik \int f(Y) dY)$, the Fourier transform and an amplitude function.

Practically, the difference between exact and approximate impact parameters p and \tilde{p} is negligibly small, typically below 1 m but reaching 5 m in very strong multipath conditions (Gorbunov and Lauritsen, 2004).

2.4 Ionospheric correction

In order to remove the ionospheric contribution to the measured bending, measured profiles at the two GNSS frequencies f_1 and f_2 are combined. This is useful because, in the L-band, ionospheric refractivity is proportional to the inverse square of the frequency whereas the neutral atmosphere refractivity is largely independent of frequency. The simplest method to correct for ionospheric effects is by linear combination of bending angles at a common impact parameter (Vorob'ev and Krasil'nikova, 1994).

$$\alpha(a) = \frac{f_1^2 \alpha_1(a) - f_2^2 \alpha_2(a)}{f_1^2 - f_2^2} \quad (2.39)$$

The resulting linear combination bending angle profiles at altitudes in excess of about 50 km are very noisy. Residual errors of the ionospheric correction originate from small scale ionospheric turbulence and receiver noise. The relative error of the derived bending angle profile may be greater than 100% in these regions and it is not appropriate to derive refractivity from these data. Rather, it is preferable to use a combination of the measured L1 and L2 bending angle data with a climatological bending angle profile α_{CLIM} , calculated (currently) from the density profile of the MSISE-90 climatological model (Hedin, 1991) or the COSMIC-based BAROCLIM climatology (Scherllin-Pirscher et al., 2015).

The `ropp_pp` module includes the ionospheric correction algorithm devised by Gorbunov (2002). The method aims to combine measured and climatological bending angle data and reduce noise by the application of statistical optimization. The optimal linear combination is expressed as a matrix equation to compute the neutral atmospheric bending angle α_N and ionospheric bending angle α_I ,

$$\begin{pmatrix} \alpha_N(a) \\ \alpha_I(a) \end{pmatrix} = \begin{pmatrix} \alpha_{CLIM}(a) \\ \bar{\alpha}_I(a) \end{pmatrix} + \bar{K}^{-1} \begin{pmatrix} \Delta\alpha_1(a) - \bar{\alpha}_I(a) \\ \Delta\alpha_2(a) - (f_1/f_2)^2 \bar{\alpha}_I(a) \end{pmatrix} \quad (2.40)$$

The matrix \bar{K} is computed from a noise covariance matrix C_N and signal covariance matrix C_S (Gorbunov, 2002),

$$\bar{K}^{-1} = (K^T C_N^{-1} K + C_S^{-1})^{-1} K^T C_N^{-1} \quad (2.41)$$

where $K_{11} = K_{12} = K_{21} = 1$ and $K_{22} = f_1^2/f_2^2$. The climatological bending angle profile $\alpha_{CLIM}(a)$ is fitted to the observed data by regression with the linear combination solution between user-specified heights, typically 40 and 60 km. $\bar{\alpha}_I(a)$ is an estimate of the ionospheric bending angle computed as

$$\bar{\alpha}_I(a) = \frac{f_1^2}{f_2^2 - f_1^2} (\Delta\bar{\alpha}_1(a) - \Delta\bar{\alpha}_2(a)) \quad (2.42)$$

where $\Delta\alpha_i(a)$ is the deviation of L1 and L2 bending angles from the fitted model bending angle profile:

$$\Delta\alpha_i(a) = \alpha_i(a) - \alpha_{CLIM}(a). \quad (2.43)$$

2.5 Abel inversion

The bending angle can be derived from the refractive index n by the Abel integral (Fjeldbo et al., 1971; Melbourne et al., 1994; Kursinski et al., 1997).

$$\begin{aligned} \alpha(a) &= -2 \int_{r_t}^{\infty} d\alpha = -2a \int_{r_t}^{\infty} \frac{1}{\sqrt{r^2 n^2 - a^2}} \frac{d \ln(n)}{dr} dr \\ &= -2a \int_a^{\infty} \frac{1}{\sqrt{x^2 - a^2}} \frac{d \ln(n)}{dx} dx . \end{aligned} \quad (2.44)$$

The second expression is obtained by substituting $x = nr$ (and $a = n(r_t)r_t$). The negative sign in Equation (2.44) follows the convention that bending towards the Earth's surface is positive.

Remarkably, Equation (2.44) can be inverted, so that the refractive index profile $n(r)$ for a given corrected bending angle profile $\alpha(a)$ is given by

$$n(r) = \exp \left[\frac{1}{\pi} \int_x^{\infty} \frac{\alpha(a)}{\sqrt{a^2 - x^2}} da \right], \quad (2.45)$$

where again $x = nr$. To reduce the influence of the infinite upper boundary condition on the inversion of a finite depth bending angle profile, the corrected bending angle profile is extended above the highest measurement impact parameter using the climatological profile α_{CLIM} discussed in Sec 2.4.

The `ropp_pp` module includes algorithms to carry out the Abel transform and its inverse.

2.6 Refractivity

For convenience, refractivity N is often used instead of the refractive index n . These are related by

$$N = (n - 1) \times 10^6 . \quad (2.46)$$

At microwave frequencies in the Earth's atmosphere, N varies due to contributions from the dry neutral atmosphere, water vapour, free electrons in the ionosphere and particulates.

$$N = \kappa_1 \frac{P}{T} + \kappa_3 \frac{e}{T^2} + \kappa_4 \frac{n_e}{f^2} + \kappa_5 W . \quad (2.47)$$

The first term is the dry neutral atmosphere contribution where P is the total (dry air plus water vapour) atmospheric pressure and T is the temperature. The second term is the water vapour contribution where e is the partial water vapour pressure. The third ionospheric contribution results from free electrons in the ionosphere where n_e is the electron number density and f is the transmitter frequency of the radio signal. The final term results from scattering, mainly due to cloud droplets where W is the liquid water content.

For radio occultation soundings it is usually assumed that the scattering term is negligible, and that ionospheric effects have been removed during the pre-processing, such as by ionospheric correction of the bending angles (Vorob'ev and Krasil'nikova, 1994). The forward models in the ROPP module `ropp_fm` relate bending angle and refractivity profiles to pressure, temperature and humidity. This enables radio occultation data to be applied to NWP data assimilation for example. The ROPP 1D-Var module `ropp_1dvar` provides software for one-dimensional retrievals. Further details are provided in Part II of the ROPP User Guide (ROM SAF, 2016).

2.7 Dry temperature

The so-called dry temperature is obtained by ignoring the water vapour in Equation (2.47) such that

$$N = \kappa_1 \frac{P}{T} . \quad (2.48)$$

With this assumption, refractivity is proportional to density $\rho = P/RT$, where R is the specific gas constant for dry air. Pressure can be obtained by assuming hydrostatic equilibrium $dP/dz = -\rho g$, where g is the gravitational acceleration.

The `ropp_pp` module includes an algorithm to solve for the temperature with these assumptions. The resulting temperature is denoted *dry temperature* to distinguish it from real temperature. The dry temperature is practically identical to real temperature in the stratosphere and above, where the water vapour term in Equation (2.47) can be neglected, but it is significantly smaller than real temperature in moist regions of the troposphere.

Section 4.6.3 provides further details.

2.8 Summary

Several key processing stages are required to derive ionospheric corrected bending angle, refractivity, and dry temperature profiles from radio occultation measurements. ROPP provides software to perform this pre-processing which enables users to process radio occultation measurements from any mission in a consistent manner before the resulting data are used for NWP or climate applications.

References

- Born, M. and Wolf, E., *Principles of Optics*, Pergamon Press, Oxford, 1980.
- Fjeldbo, G., Kliore, G. A., and Eshleman, V. R., The neutral atmosphere of Venus as studied with the Mariner V radio occultation experiments, *Astron. J.*, *76*, 123–140, 1971.
- Gorbunov, M. E., Ionospheric correction and statistical optimization of radio occultation data, *Radio Sci.*, *37*, 10.1029/2000RS002 370, 2002.
- Gorbunov, M. E. and Lauritsen, K. B., Analysis of wave fields by Fourier integral operators and their application for radio occultations, *Radio Sci.*, *39*, doi:10.1029/2003RS002 971, 2004.
- Gorbunov, M. E. and Lauritsen, K. B., Radio holographic filtering of noisy radio occultations, in *Atmosphere and Climate Studies by Occultation Methods*, edited by U. Foelsche, G. Kirchengast, and A. Steiner, Springer, Berlin, Heidelberg, 2006.
- Gorbunov, M. E., Lauritsen, K. B., Rhodin, A., Tomassini, M., and Kornblueh, L., Analysis of the champ experimental data on radio-occultation sounding of the earth's atmosphere, *Izvestiya, Atmospheric and Oceanic Physics*, *41*, 798–813, 2005.
- Hedin, A. E., Extension of the MSIS thermosphere model into the middle and lower atmosphere, *J. Geophys. Res.*, *96*, 1159–1172, 1991.
- Jensen, A. S., Lohmann, M. S., Benzon, H.-H., and Nielsen, A. S., Full Spectrum Inversion of radio occultation signals, *Radio Sci.*, *38*, 1040, doi:10.1029/2002RS002 763, 2003.
- Kursinski, E. R., Hajj, G. A., Schofield, J. T., Linfield, R. P., and Hardy, K. R., Observing earth's atmosphere with radio occultation measurements using the Global Positioning System, *J. Geophys. Res.*, *102*, 23.429–23.465, 1997.
- Melbourne, W. G., Davis, E. S., Duncan, C. B., Hajj, G. A., Hardy, K. R., Kursinski, E. R., Meehan, T. K., and Young, L. E., The application of spaceborne GPS to atmospheric limb sounding and global change monitoring, Publication 94–18, Jet Propulsion Laboratory, Pasadena, Calif., 1994.
- ROM SAF, The Radio Occultation Processing Package (ROPP) User Guide. Part IIb: 1D–Var module, SAF/ROM/METO/UG/ROPP/003b, Version 9.0, 2016.
- Scherllin-Pirscher, B., Syndergaard, S., Foelsche, U., and Lauritsen, K. B., Generation of a bending angle radio occultation climatology (BAROCLIM) and its use in radio occultation retrievals, *Atmospheric Measurement Techniques*, *8*, 109–124, 2015.
- Vorob'ev, V. V. and Krasil'nikova, T. G., Estimation of the accuracy of the atmospheric refractive index recovery from doppler shift measurements at frequencies used in the NAVSTAR system, *USSR Phys. Atmos. Ocean, Engl. Transl.*, *29*, 602–609, 1994.

3 ROPP Pre-processor: amplitude and phase to bending angle

The ROPP pre-processor module (`ropp_pp`) includes routines to process measured L1 and L2 amplitude and excess phase as a function of time during an occultation to retrieve profiles of bending angle as a function of impact parameter. These data can be corrected for ionospheric effects and inverted to retrieve refractivity as described in Section 4. Routines are included to preprocess and filter the measured data (`ropp_pp_preprocess`) and compute bending angles using geometric optics (`ropp_pp_bending_angle_go`) and wave optics (CT2) (`ropp_pp_bending_angle_wo`) algorithms. Background information on these algorithms is provided in Section 2 and Gorbunov and Lauritsen (2004), Gorbunov et al. (2005).

Figure 3.1 shows example corrected refractivity and bending angle profiles resulting from applying the `ropp_pp` routines to process the input L1 and L2 amplitude and excess phase measurements provided to users.

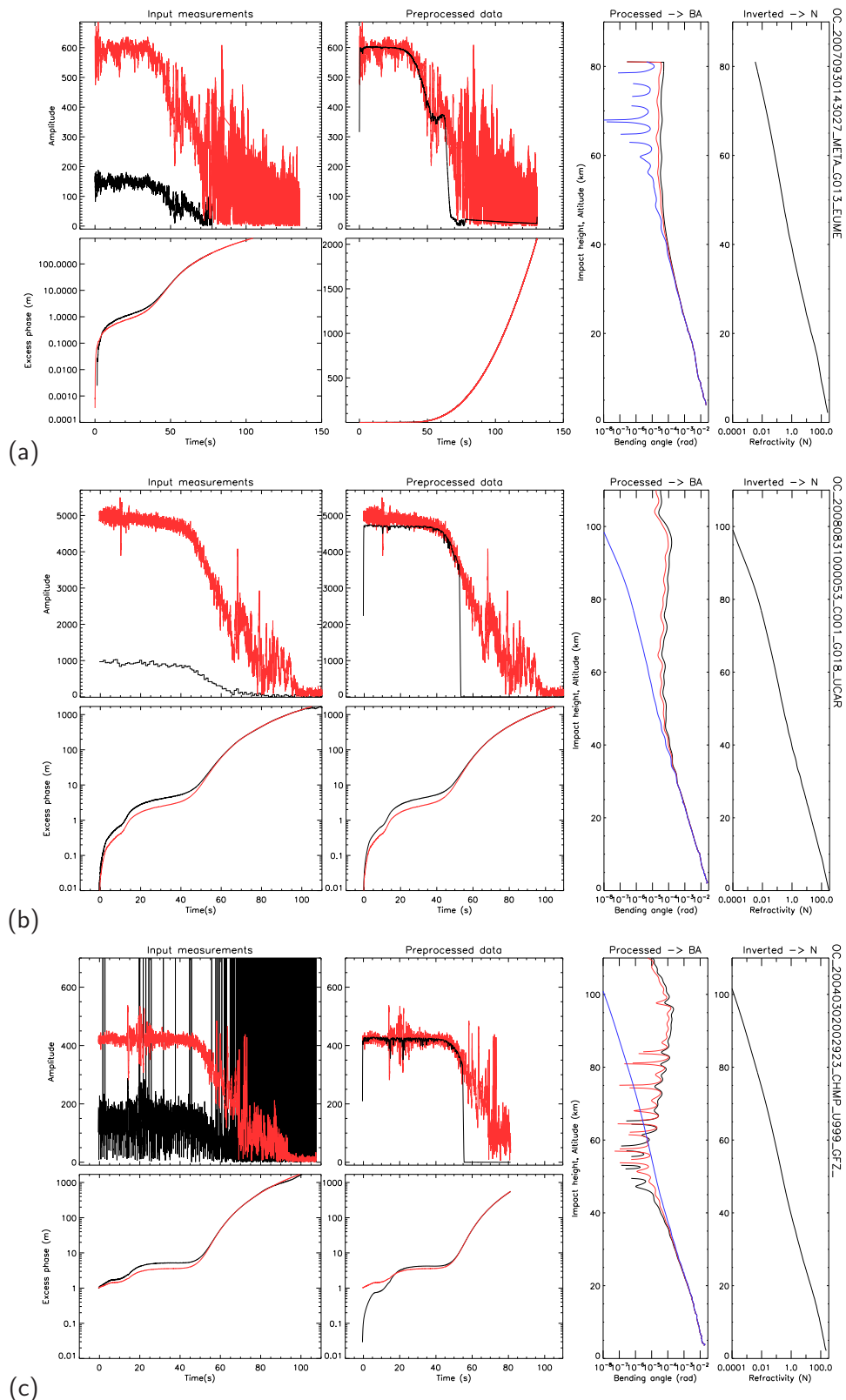


Figure 3.1: (a) A GRAS profile, (b) a COSMIC profile and (c) a CHAMP profile. Example L1 (red) and L2 (black) amplitude and phase measured data, pre-processed data and the resulting bending angle profiles computed using `ropp_pp` routines. The ionospheric corrected bending angle as a function of impact parameter is plotted in blue, from which the refractivity profile as a function of altitude is computed.

3.1 ROPP occultation tool

A stand-alone tool `ropp_pp_occ_tool` is provided in the `ropp_pp` module as an illustration of how the `ropp_pp` routines can be implemented to process amplitude and excess phase data and derive profiles of bending angle and refractivity. Figure 3.2 shows the general outline of `ropp_pp_occ_tool` processing.

3.1.1 Implementation

The `ropp_pp_occ_tool` executable is run using the command

```
ropp_pp_occ_tool <inputdatafile> [options] -o <outputfile>
```

where `<inputdatafile>` is a ROPP netCDF file (ROM SAF, 2016) containing the input radio occultation data and `<outputfile>` will contain the processed input data together with the computed L1 and L2 channel bending angle, corrected bending angle and refractivity profiles.

The following command line options can be used with `ropp_pp_occ_tool` tool:

<code>-h</code>	give help menu
<code>-o <outfile></code>	name of netCDF ROPP format output filename
<code>-c <config_file></code>	configuration parameters filename
<code>-m <method></code>	ionospheric correction method [NONE,MSIS,GMSIS,BG,BARO,GBARO] (default GMSIS)
<code>-mfile <mfile></code>	climatology coefficients filename (default local search for <code>MSIS_coeff.nc</code>)
<code>-bfile <bfile></code>	background atmospheric profile file (if using 'BG' method)
<code>-navfile <nfile></code>	external navigation bit file path (default internal correction)
<code>-occ <occ_method></code>	processing method, WO or GO (default WO)
<code>-filter <f_method></code>	filtering method, slpoly or optest (default slpoly)
<code>-d</code>	output additional diagnostics (VerboseMode)
<code>-v</code>	output version information

The method used to retrieve bending angles from phase and amplitude measurements can be specified as a command-line option using the `-occ` flag. As default (or using `-occ WO`), `ropp_pp_occ_tool` computes bending angles using wave optics processing (CT2) below 25 km and geometric optics above.

The method used to conduct the ionospheric correction can be specified as a command-line option using the `-m` flag. By default, or by using `-m GMSIS` or `-m GBARO`, `ropp_pp_occ_tool` applies the statistical optimization and ionospheric correction algorithm described by Gorbunov (2002), using the best fitting member of the MSISE-90 or BAROCLIM global climatology for both the ionospheric correction and the Abel inversion processing. Specifying `-m MSIS` or `-m BARO` also applies the statistical optimization and ionospheric correction, but using the *local* MSIS-90 or BAROCLIM climatology profile for the observation location and time of year. Specifying `-m BG` uses a background refractivity profile read in from the file given with the `-bfile` option. Specifying `-m NONE` as a command line option reverts to the linear combination of L1 and L2 bending angles (Equation (2.39)) and no climatological or background model data are used for the Abel inversion.

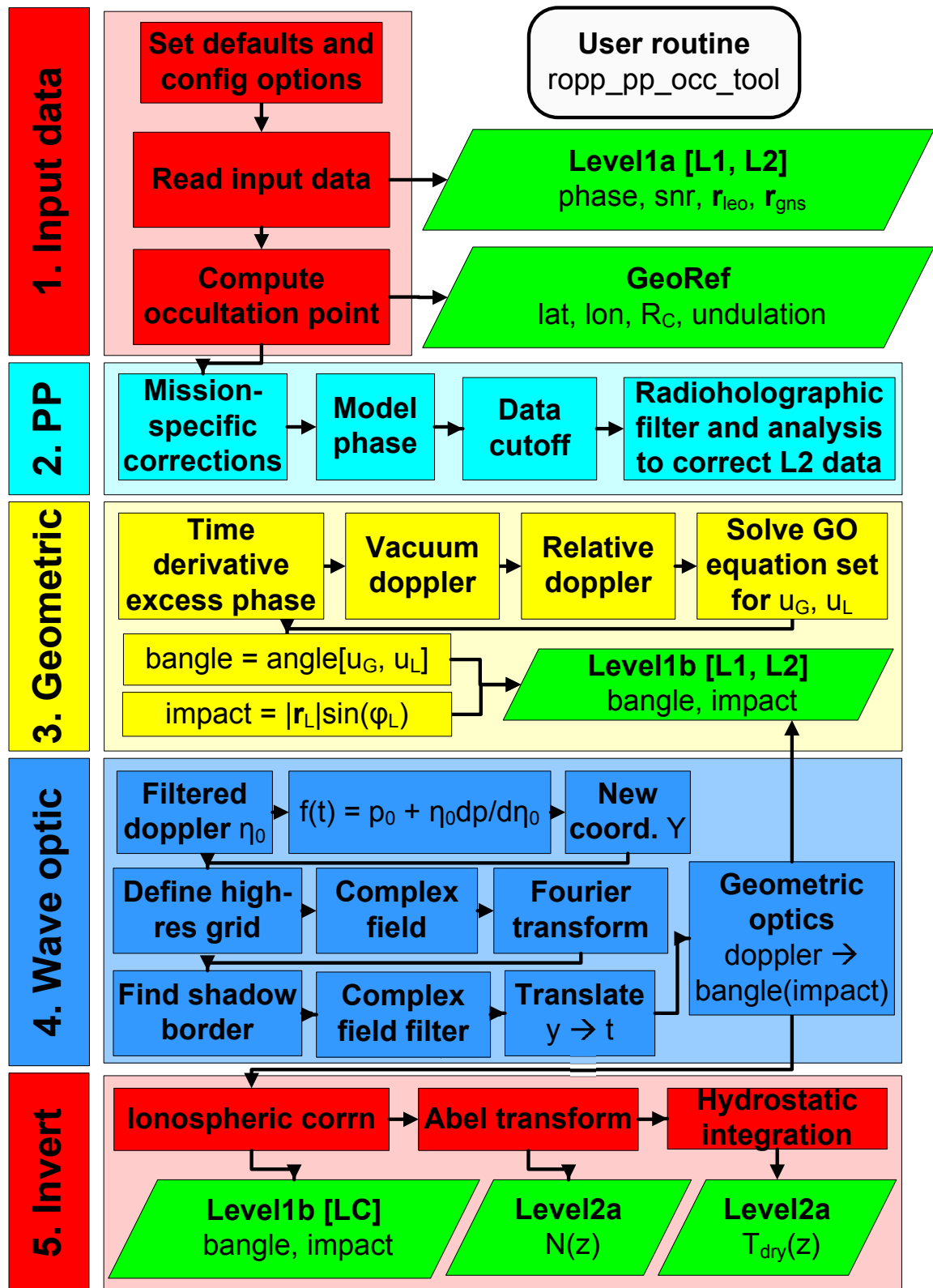


Figure 3.2: Flow chart illustrating calling tree of the ROPP pre-processor occ tool to compute ionospheric corrected bending angle and refractivity profiles from input L1 and L2 channel amplitude and phase measurements. See also Figure 4.2.

The filtering algorithm used in the wave optics, geometric optics and ionospheric correction processing may be set on the command line using the `-filter` flag. As default (or using `-filter slpoly`) a sliding polynomial filter is used. Specifying `-filter optest` causes an optimal estimation filter to be used.

3.1.2 Code organisation

Figure 3.2 shows how `ropp_pp_occ_tool` is composed of the following stages:

- **Input data access**

Setup the input data arrays, read the input data as a structure of type `R0data`. The sub-structure of type `L1atype` holds all satellite position, amplitude and phase data. Further details on the ROPP data structures can be found in the ROPP User Guide Part I (ROM SAF, 2016).

- **Compute occultation point**

The `ropp_utils` routine `occ_point` determines the lowest occultation perigee point projected to the Earth's surface to determine the latitude and longitude of the occultation point. Cartesian coordinates of the centre of curvature and the radius of curvature at the occultation point are also computed. The undulation, the difference between the WGS84 ellipsoid (NGA) and the EGM96 geoid (NASA/NIMA) at the occultation point, is computed using geoid coefficient files `egm96.dat` and `corrcoef.dat` obtained from the National Image and Mapping Agency (NIMA/NASA). These files are provided as part of the `ropp_pp` distribution in the `data/` directory. The explicit path to these files can be specified as a configuration option (see Table 3.1), or by setting the environment variables `GEOPOT_COEF` and `GEOPOT_CORR` appropriately. To use the files included in the `ropp_pp` distribution, the user would say

```
> export GEOPOT_COEF=$ROPP_ROOT/ropp_pp/data/egm96.dat
> export GEOPOT_CORR=$ROPP_ROOT/ropp_pp/data/corrcoef.dat
```

- **Preprocessing of L1 and L2 amplitude and excess phase**

Filtering and quality control procedures outlined in Section 2.3.1 are implemented by `ropp_pp_preprocess`. Mission-specific preprocessing, such as open loop phase modulation for GRAS and COSMIC data, is conducted, depending on the identity of the LEO satellite (variable `leo_id`). A smoothed bending angle profile is computed and used to identify where signal tracking errors dominate and unusable measured data are cutoff in routine `ropp_pp_cutoff`. Analysis of local spatial spectra is performed to derive a penalty function (Equation (2.8)) which is used to correct channel L2 amplitude and excess phase measurements in `ropp_pp_correct_L2`. Note that the corrected excess phase and amplitude data are written to the output file.

- **Computation of bending angle profiles by geometric optics**

The algorithm described in Section 2.3.2 is implemented in routine `ropp_pp_bending_angle_go`. The geometric optics set of equations are solved for a given time sample by calling `ropp_pp_geometric_optics`. The results are written to variables `bangle_L1`, `impact_L1`, `bangle_L2` and `impact_L2` of `Level1b` in the `R0data` structure.

- **Computation of bending angles below 25 km by wave optics (CT2)**

The CT2 algorithm developed by Gorbunov and Lauritsen (2004), outlined in Section 2.3.3 is implemented in routine `ropp_pp_bending_angle_wo` which calls `ropp_pp_DCT`. The results are written to variables `bangle_L1`, `impact_L1`, `bangle_L2` and `impact_L2` of `Level1b` in the `R0data` structure.

- **Perform ionospheric correction and statistical optimization**

A non-optimised bending angle profile (LC) is computed using L1 and L2 bending angles from Equation (2.39). The results are written to variables `bangle`, `impact`. The Gorbunov (2002) ionospheric correction with statistical optimization algorithm is implemented in `ropp_pp_ionospheric_correction`. The resulting optimized bending profile is written to variable `bangle_opt`, `impact_opt`. See Section 4 for further details.

- **Perform Abel inversion to derive the refractivity profile**

Routine `ropp_pp_invert` performs the Abel inversion (Equation (2.45)) to derive a refractivity profile from the corrected bending angles. See Section 4 for further details.

- **Integrate refractivity to derive dry temperature**

Routine `ropp_pp_tdry` performs the hydrostatic integration to derive a temperature profile from the refractivity profile, ignoring the contribution of water vapour. See Section 4 for further details.

3.2 Configuration options

A number of configuration options can be defined by the user in order to tune the preprocessing and wave optics algorithms. Table 3.1 lists these configuration options and their default values held in a structure of derived type `PPConfig` (see also Table 4.1). The use of these parameters within `ropp_pp` are described within this User Guide. A user can specify configuration options at run-time by setting their values in a configuration file and including the `'-c <config_file>'` command line option when running `ropp_pp_invert_tool`. Sample configuration files are provided in the `config` directory of the `ropp_pp` distribution.

The configuration file is read, if specified, and the elements of a variable of type `PPConfig` are overwritten by calling `ropp_pp_read_config`.

```
USE ropp_pp
TYPE(PPConfig) :: config
CALL ropp_pp_read_config(config_file, config)
```

3.3 Preprocessing `ropp_pp_preprocess`

Routine `ropp_pp_preprocess` is called to perform all L1 and L2 amplitude and excess phase data quality control, filtering and correction (Gorbunov et al., 2006). The modified data are written in the output file along with computed bending angle and refractivity profiles. Examples are plotted in Figure 3.1. Note that this routine requires that the `ropp_io` module is installed (ROM SAF, 2016).

PPConfig		
Structure element	Default	Description
...%occ_method	WO	Phase to bending angle method (GO = geometric, WO = wave optics)
...%filter_method	slpoly	Algorithm for data filtering (slpoly = sliding polynomial, optest = optimal estimation)
...%fw_go_smooth	3000.0 m	Filter width for smoothed GO bangle
...%fw_go_full	3000.0 m	Filter width for full res GO bangle
...%fw_wo	2000.0 m	Filter width for wave optics bangle above 7 km
...%fw_low	-1000.0 m	Filter width for wave optics bangle below 7 km
...%hmax_wo	25 km	Maximum height for WO processing
...%msisfile	MSIS_coef.nc	Climatology coefficients filename (as used in preprocessing)
...%Acut	0.0	Fractional cut-off limit for amplitude
...%Pcut	-2000.0 m	Cut off limit for impact height
...%Bcut	0.1 rad	Cut off limit for bending angle
...%Hcut	-250 km	Cut off limit for straight line tangent altitude
...%CFF	3	Complex filter flag (used in radioholographic filtering)
...%dsh	200 m	Shadow border width
...%opt_DL2	.TRUE.	Flag to indicate degraded L2 data
...%out_spectra	.FALSE.	Flag to compute and output spectra
...%egm96	egm96.dat	Path to EGM96 coefficients file
...%corr_egm96	corrcoef.dat	Path to EGM96 corrections file
...%navbit_file		Path to external navigation bit file

Table 3.1: Configuration options held as elements of the PPConfig structure which are used by ropp_pp preprocessing and wave optics routines. The default values are assumed unless overwritten by reading configuration options from an input file.

```
USE ropp_io_types
USE ropp_pp
TYPE(ROprof)   :: ro_data
TYPE(PPconfig) :: config
CALL ropp_pp_preprocess(ro_data, config)
```

3.3.1 Model excess phase

Correction and filtering of excess phase data requires a reference excess phase sample which describes the excess phase rate within about 10 Hz. Excess phase data are corrected and filtered using an algorithm devised

by Michal Gorbunov. The phase model is computed by calling `ropp_pp_modelphase`, which derives excess phase from the MSIS climatology model (Hedin, 1991). Since the MSIS model does not contain humidity, it is necessary to retrieve dry temperature and pressure from the MSIS refractivity before re-computing the MSIS refractivity profile with an assumed constant relative humidity of 90% in the troposphere (below 15 km). A model bending angle profile is derived by Abel transform of the refractivity profile. Routine `ropp_pp_bangle2phase` solves the following equation for $t(p)$ to find the sample time associated with a given impact parameter and MSIS bending angle value.

$$\alpha_{MSIS}(p) = \theta(t) - \cos^{-1} \frac{P}{r_G(t(p))} - \cos^{-1} \frac{P}{r_L(t(p))} \quad (3.1)$$

The relative Doppler shift is computed using Equation (2.17) where $\mathbf{u}_L(t(p))$ and $\mathbf{u}_G(t(p))$ are the ray vectors at the GNSS and LEO satellites, forming angles of $\cos^{-1}(P/r_G(t(p)))$ and $\cos^{-1}(P/r_L(t(p)))$ with the local verticals respectively. The model phase is calculated by integrating the relative doppler shift over time, so that model excess phase ΔS_{MSIS} is given by

$$\Delta S_{MSIS} = \Psi_{MSIS}(t) - \Psi_0(t) = \int_0^t d(p(t))dt - |\mathbf{r}_G - \mathbf{r}_L| \quad (3.2)$$

3.3.2 Raw sampling and open loop data pre-processing

GRAS data are a combination of raw sampling measurements for a ray height below 5-10 km and closed loop tracking measurements elsewhere. COSMIC data are a combination of open loop measurements for a ray height below 8-11 km and closed loop measurements elsewhere. The raw sampling or open loop phase measurements include a navigation message in the form of pseudo-random addition of 0 or π to each data sample. These are removed by `ropp_pp_openloop`.

The raw sampling and closed loop segments of input GRAS data are identified using the lost carrier flag profile variable in an input file. This is a bitwise variable containing navigation bit data, as follows:

- bit 0: 0=closed loop mode, 1=raw sampling mode
- bit 1: external navigation bits
- bit 2: external navigation bits quality
- bit 3: missing data
- bit 4: alternative navigation bits
- bit 5: alternative navigation bits quality
- bit 6: closed loop and raw sampling overlap
- bit 7: internal navigation bits
- bit 8: internal navigation bits quality

The open loop fragment of the input COSMIC data is identified by variable `XMDL` in the UCAR `atmPhs` user data, which is preserved as an additional variable when creating the ROPP `netCDF` data file using the `ucar2ropp` tool (ROM SAF, 2016). Open loop data are identified where `XMDL` \neq -999. If available, COSMIC data navigation bits may be read from a separate text file (path specified by the `navbit_file` config option), which contains a sequence of two-digit binary value navigation bits as a function of GNSS

time. These bits are interpolated to the occultation measurement times and stored as variable LCF.

The excess phase deviation from the MSIS model excess phase is then computed as,

$$\Delta\Phi(t_i) = k(\Delta S - \Delta S_{MSIS}) + \pi(LCF/2 \bmod 2) \quad (3.3)$$

This quantity may contain extra or missing cycles due to the navigation message. Therefore, the excess phase is re-accumulated as

$$\Delta\bar{\Phi}(t_i) = \Delta\bar{\Phi}(t_{i-1}) + ((\Delta\Phi(t_i) - \Delta\Phi(t_{i-1}) + \pi) \bmod 2\pi) - \pi \quad (3.4)$$

The corrected excess phase data are then computed for channels L1 and L2 in `ropp_pp_openloop` as

$$\Delta S = \Delta S_{MSIS} + \Delta\bar{\Phi} \quad (3.5)$$

3.3.3 Data cutoff

Configuration options `config%Acut`, `config%Bcut`, `config%Pcut` and `config%Hcut` are used in routines `ropp_pp_cutoff` and `ropp_pp_cutoff_amplitude` to set the amplitude, bending angle, impact height and straight-line tangent height criteria which are used to reject measured data corrupted by signal tracking errors. Data are cutoff if the time when L1 amplitude reaches fraction `config%Acut` of the maximum measured amplitude is more than 2 times the smoothing window from the data end (start) point for a setting (rising) occultation. A bending angle profile is computed by geometric optics with strong smoothing (defined by `config%fw_go_smooth`). Data are cutoff where the estimated bending angle exceeds `config%Bcut` at an impact height above `config%Pcut` and where the straight-line tangent height is above `config%Hcut`.

3.3.4 Compute spectra

If configuration option `config%opt_spectra` is set, localised spectra of the uncorrected wave fields in the time-frequency domain are computed by calling `ropp_pp_spectra`. See also Section 3.6.

3.3.5 Correction of L2 data

If configuration option `config%opt_d12` is set, the L2 channel data are assumed to be degraded. The L2 amplitude is replaced by a smooth geometric optics amplitude computed using the MSIS impact parameter levels in `ropp_pp_amplitude_go`. Routine `ropp_pp_correct_L2` follows the algorithms described by Gorbunov et al. (2006) (Equations (2.6)–(2.12)). First, a radioholographic filter is applied to the L2 data. By radioholographic analysis smoothed profiles of L1 and L2 impact parameter $\bar{p}_{1,2}(t)$, bending angle $\bar{\alpha}_{1,2}(t)$ and impact parameter spectral widths $\delta p_{1,2}(t)$ are computed. A quality control indicator is computed as

$$Q^{L2}(t) = \left(\frac{|\bar{p}_1(t) - \bar{p}_2(t)|}{\Delta p_A} + \frac{\delta p_2(t)}{\Delta p_D} \right)^2 \quad (3.6)$$

where parameters $\Delta p_A = 0.20$ km and $\Delta p_D = 0.15$ km. The quality control indicator determines a cut-off point where L2 data quality is very low, either due to lost closed loop tracking or overly noisy data. In the region of poor data quality the L2 bending angles are estimates as $\bar{\alpha}_1 + \Delta\alpha_I$ where $\Delta\alpha_I$ is the estimated ionospheric bending angle difference using data above the cutoff point. The resulting bending angle profiles are used to compute excess phase data $\Phi_{1,2}$ (using `ropp_pp_bangle2phase`), which are used to correct L2 excess phase and amplitude data according to Equations (2.8)–(2.12).

3.4 Geometric optics `ropp_pp_bending_angle_go`

Computation of L1 and L2 bending angles and impact parameters using the geometric optics approximation (Section 2.3.2) is implemented by calling `ropp_pp_bending_angle_go`.

```
USE ropp_pp
CALL ropp_pp_bending_angle_go(time,      &      ! Time of samples
                              r_leo,     &      ! LEO cartesian coordinates
                              r_gns,     &      ! GNSS cartesian coordinates
                              r_coc,     &      ! Centre of curvature coordinates
                              phase_L1,  &      ! L1 excess phase data
                              phase_L2,  &      ! L2 excess phase data
                              w_smooth,  &      ! Filter window size
                              filter,    &      ! Filter algorithm method
                              impact_L1, &      ! L1 impact parameters (OUT)
                              bangle_L1, &      ! L1 bending angles (OUT)
                              impact_L2, &      ! L2 impact parameters (OUT)
                              bangle_L2) ! L2 bending angles (OUT)
```

To avoid differentiating noisy excess phase data to compute the relative Doppler shift, the corrected excess phase data are filtered before differentiating the trend and detrended parts of the signal separately. The filter window size is defined by argument `w_smooth`, which is set by configuration options `fw_go_smooth` and `fw_go_full` for computation of smoothed and full resolution bending angle profiles respectively. Equations (2.22) are then solved for each time sample by calling `ropp_pp_geometric_optics`.

3.5 Wave optics (CT2) `ropp_pp_bending_angle_wo`, `ropp_pp_DCT`

Computation of L1 and L2 bending angles and impact parameters using the wave optics approximation (Section 2.3.3) is implemented using the CT2 algorithm developed by Gorbunov and Lauritsen (2004) in routine `ropp_pp_bending_angle_wo`. The CT2 processing is implemented by `ropp_pp_DCT`. The wave optics processing is only applied below a specific height, which is set by configuration option `config%hmax_wo`. The default is 25 km. Bending angles calculated by geometric optics are preserved above this height.

Wave fields $u_1(t)$ and $u_2^{cor}(t)$ are transformed by the Fourier Integral Operator (Equation (2.38)). The transformation from measured data to the new coordinate system (Equations (2.31)–(2.35)) is conducted

with a smooth model of the Doppler frequency shift σ_0 determined using a smoothing window set by configuration option `config%fw_go_smooth`. Bending angle and impact parameters are then computed using geometric optics processing given the relative Doppler shift derived from \tilde{p} .

Shadow borders $p_{1,2}$ are determined from the maximum of the correlation of $|\hat{\Phi}_{2u_{1,2}}(p)|$ with a unit step function. The height p_2 defines the border below which L2 data are unusable for WO processing. Below p_2 , geometric optics bending angles are retained.

An estimate of the errors of bending angle profiles in the lower troposphere is obtained by the analysis of local sliding spectra of the transformed wave field,

$$w(p, \xi) = \int_{p-\Delta p/2}^{p+\Delta p/2} \cos\left(\frac{\pi(p'-p)}{\Delta p}\right) \frac{\hat{\Phi}_{2u_1}(p')}{\exp(ik\bar{\Psi}'(p))} \exp(-ik\xi p') dp' \quad (3.7)$$

where $\Delta p = 1.0$ km and $\bar{\Psi}'(p)$ is a smoothed excess phase with smoothing window determined by configuration parameter `config%fw_wo`. The tropospheric error of bending angle is then estimated as the spectral width

$$\delta\alpha_T(p) = \left(\frac{\int |w(p, \xi)|^2 \xi^2 d\xi}{\int |w(p, \xi)|^2 d\xi} \right)^{1/2} \quad (3.8)$$

This value is written to the output results as variables `bangle.L1.sigma` and `bangle.L2.sigma`.

USE ropp_pp

```
CALL ropp_pp_bending_angle_wo(time,      &      ! Time of samples
                               r_leo,     &      ! LEO cartesian coordinates
                               r_gns,     &      ! GNSS cartesian coordinates
                               r_coc,     &      ! Centre of curvature coords
                               roc,       &      ! Radius of curvature
                               phase_L1, &      ! L1 excess phase data
                               phase_L2, &      ! L2 excess phase data
                               snr_L1,   &      ! L1 amplitude data
                               snr_L2,   &      ! L2 amplitude data
                               w_ls,     &      ! Large scale filter size
                               w_smooth, &      ! Filter window size above 7 km
                               w_low,    &      ! Filter below 7 km
                               hmax_wo,  &      ! Maximum height for WO
                               filter,   &      ! Filter algorithm method
                               opt_DL2   &      ! Degraded L2 flag
                               cff       &      ! Complex filtering flag
                               dsh       &      ! Shadow border width
                               impact_L1, &      ! L1 impact parameters (OUT)
                               bangle_L1, &      ! L1 bending angles (OUT)
                               ba_sigma_L1, &      ! L1 bending angle error (OUT)
                               impact_L2, &      ! L2 impact parameters (OUT)
                               bangle_L2, &      ! L2 bending angles (OUT)
```

```
ba_sigma_L1, & ! L1 bending angle error (OUT)
```

3.6 Spatial spectra `ropp_pp_spectra_tool`, `ropp_pp_spectra`

A stand-alone tool `ropp_pp_spectra_tool` is provided with the `ropp_pp` module in order to compute local sliding spectra of an occultation measurement. Spectra are computed in both the (time-frequency) and (bending angle-impact parameter) domains and the results output to ascii data files (default filenames `R0analysis_dt*.dat`, `R0analysis_ep*.dat`). The `test/` directory of the `ropp_pp` distribution contains an example IDL routine `it_pp_spectra.pro` which illustrates how users may wish to plot the output. Examples for the occultations shown in Figure 3.1 are plotted in Figure 3.3.

3.7 Processing GRAS raw sampling data

`ropp_pp` includes functionality to process GRAS measurements in raw sampling and closed loop mode. The raw data are processed by EUMETSAT and provided (by request) in netCDF format. A tool `ropp_pp_grasrs2ropp` is provided with ROPP to read the EUMETSAT format data, merge the raw sampling and closed loop, define the lost carrier flag bits (see Section 3.3.2) and output the data in a standard ROPP format netCDF file (ROM SAF, 2016). The output file is then suitable for processing using the `ropp_pp_occ_tool`.

```
ropp_pp_grasrs2ropp <inputGRASfile> -o <ROPPfile>  
ropp_pp_occ_tool <ROPPfile> -c config/gras_pp.cf -o <outputfile>
```

References

- Gorbunov, M. E., Radio-holographic analysis and validation of Microlab-1 radio occultation data in the lower troposphere, *J. Geophys. Res.*, *107*, 10.1029/2001JD000 889, 2002.
- Gorbunov, M. E. and Lauritsen, K. B., Analysis of wave fields by Fourier integral operators and their application for radio occultations, *Radio Sci.*, *39*, doi:10.1029/2003RS002 971, 2004.
- Gorbunov, M. E., Lauritsen, K. B., Rhodin, A., Tomassini, M., and Kornblueh, L., Analysis of the champ experimental data on radio-occultation sounding of the earth's atmosphere, *Izvestiya, Atmospheric and Oceanic Physics*, *41*, 798–813, 2005.
- Gorbunov, M. E., Lauritsen, K. B., Rhodin, A., Tomassini, M., and Kornblueh, L., Radio holographic filtering, error estimation, and quality control of radio occultation data, *J. Geophys. Res.*, *111*, D10 105, 2006.
- Hedin, A. E., Extension of the MSIS thermosphere model into the middle and lower atmosphere, *J. Geophys. Res.*, *96*, 1159–1172, 1991.

NASA/NIMA, NASA and NIMA joint geopotential model EGM96 website,
<http://cddis.nasa.gov/926/egm96/egm96.html>.

NGA, National Geospatial-Intelligence Agency WGS84 website,
<http://earth-info.nga.mil/GandG/wgs84/index.html>.

NIMA/NASA, National Image and Mapping Agency website,
<http://earth-info.nima.mil/GandG/wgs84/gravitymod/egm96/egm96.html>.

ROM SAF, The Radio Occultation Processing Package (ROPP) User Guide. Part I: Input/Output module,
SAF/ROM/METO/UG/ROPP/002, Version 9.0, 2016.

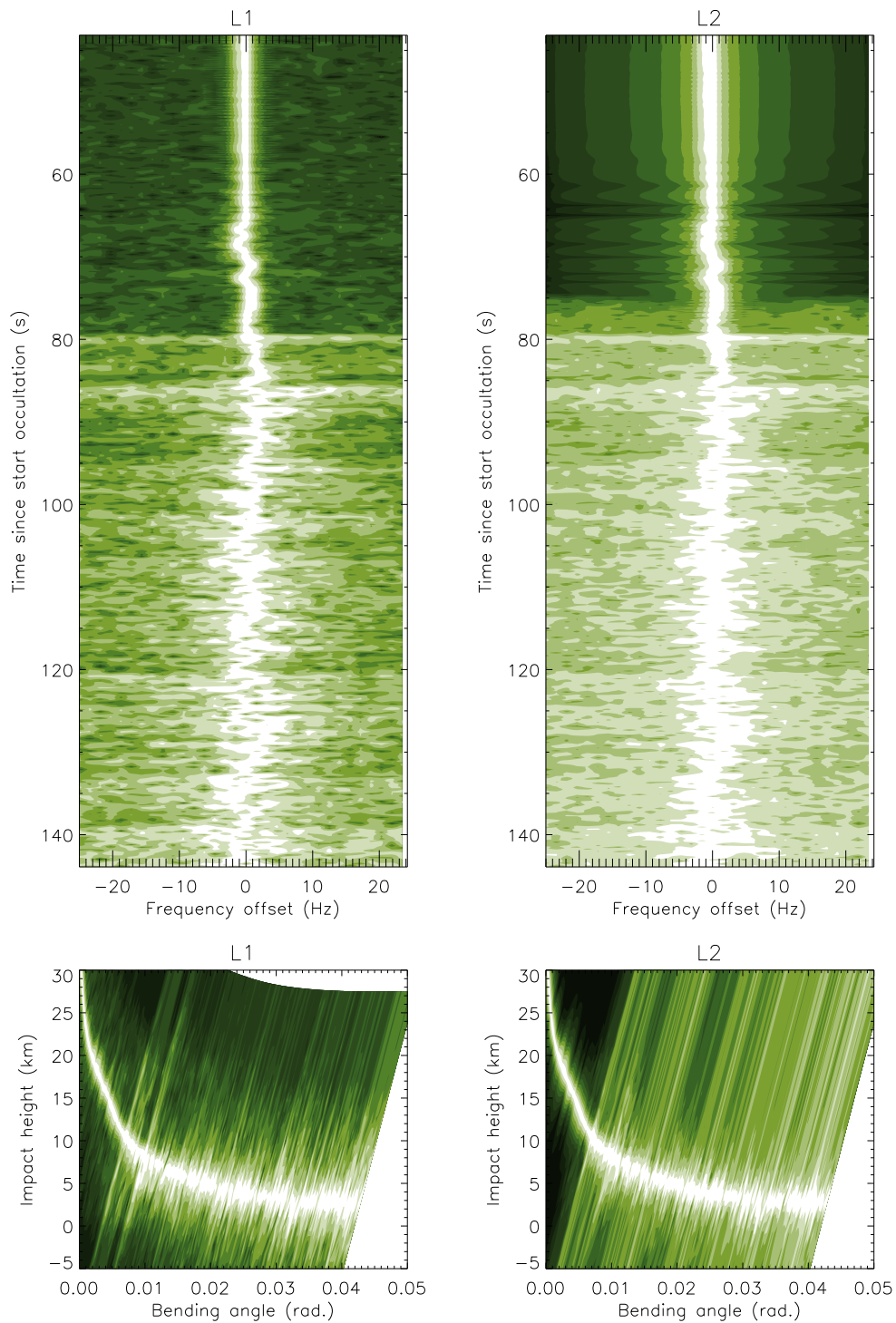


Figure 3.3: Local spatial spectra for channel L1 and L2 data for the COSMIC profile in Figure 3.1. Spectra are computed in the (time-frequency) and (bending angle-impact parameter) domains.

4 ROPP Pre-processor: L1 and L2 bending angle to refractivity and dry temperature

The ROPP pre-processor module (`ropp_pp`) includes routines to correct measured L1 and L2 bending angle data for ionospheric effects (`ropp_pp_ionospheric_correction`) and invert a corrected bending angle profile to derive refractivity (`ropp_pp_invert_refraction`) and dry temperature (`ropp_pp_tdry`).

Figure 4.1 shows example refractivity and bending angle profiles computed from measured L1 and L2 bending angle data. The results of the simple linear combination (Equation (2.39)) and the Gorbunov (2002b) ionospheric correction and statistical optimization algorithm are plotted. This example shows that errors of up to 100% may result at altitudes above 50 km due to inadequate ionospheric correction and not accounting for the infinite upper boundary condition of the Abel inversion integral (Equation (2.45)).

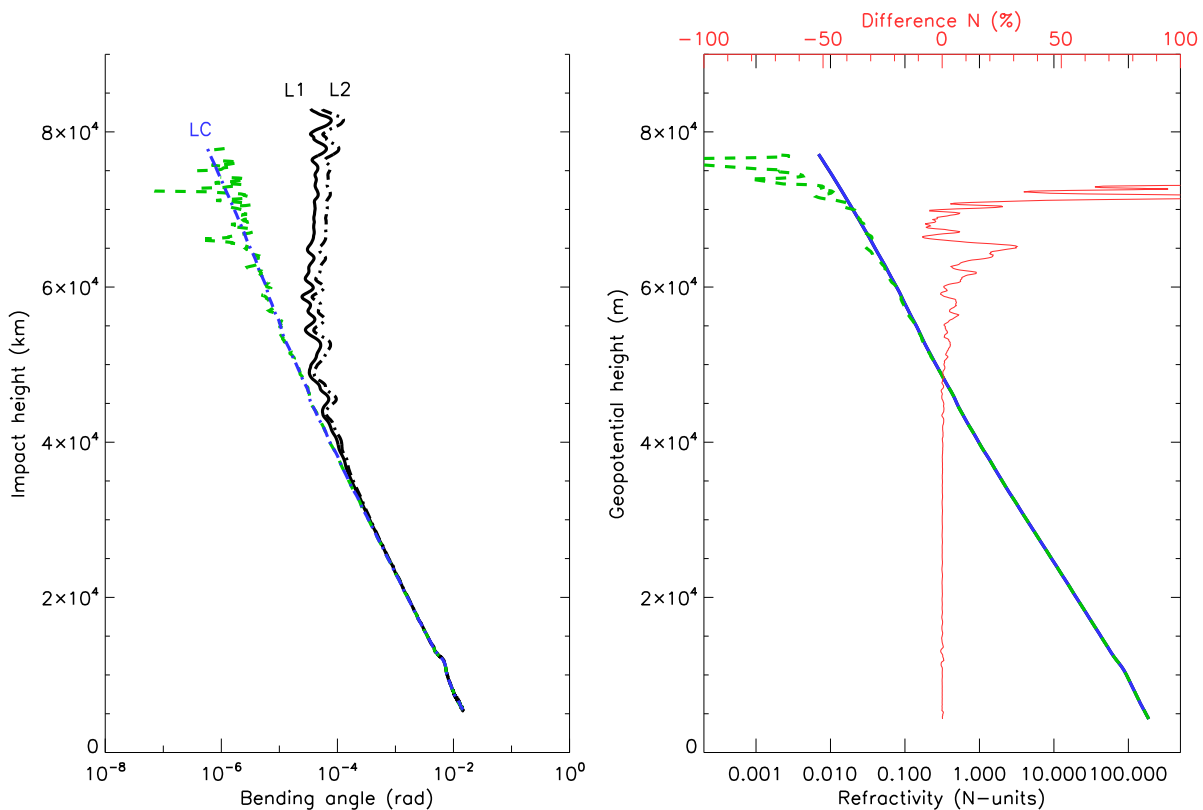


Figure 4.1: Example refractivity and bending angle profiles computed using the linear combination (green) and statistical optimization (blue) of measured bending angle data at L1 and L2 channel frequencies during a radio occultation. The difference between refractivity profiles computed using linear combination with inversion and statistical optimization with MSIS-90 climatology is plotted in red.

4.1 ROPP inversion tool

A stand-alone tool `ropp_pp_invert_tool` is provided in `ropp_pp` as an illustration of how the `ropp_pp` routines can be implemented to derive profiles of ionospheric corrected bending angle and refractivity from L1 and L2 channel bending angle data. Figure 4.2 shows how the `ropp_pp` routines are integrated in the `ropp_pp_invert_tool` code.

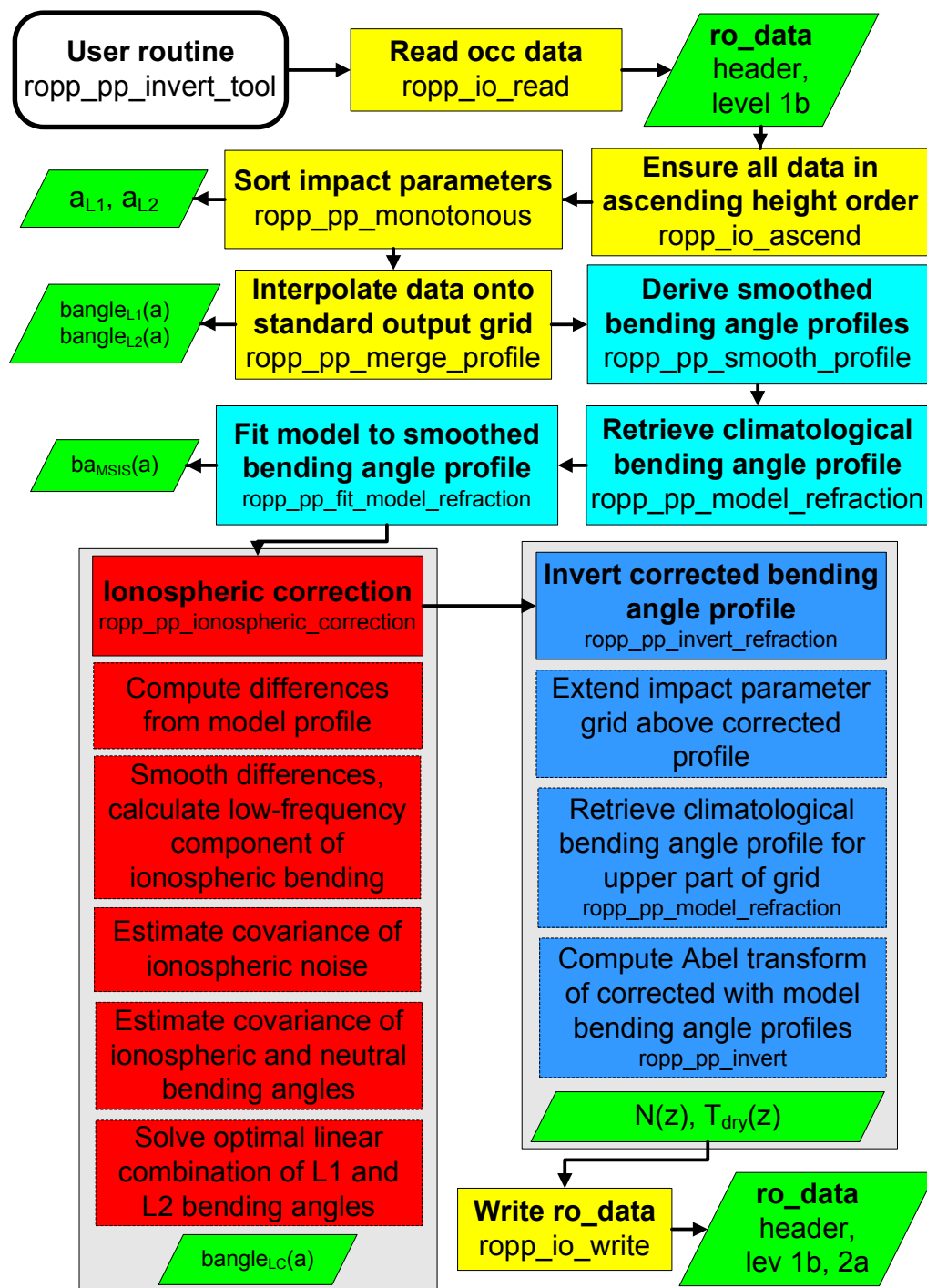


Figure 4.2: Flow chart illustrating calling tree of the ROPP pre-processor invert tool to compute ionospheric corrected bending angle and refractivity profiles from input L1 and L2 channel bending angles.

4.1.1 Implementation

The `ropp_pp_invert_tool` executable is run using the command

```
ropp_pp_invert_tool <inputdatafile> [options] -o <outputfile>
```

where `<inputdatafile>` is a ROPP netCDF file (ROM SAF, 2016) containing the input radio occultation data and `<outputfile>` will contain the input data together with the corrected bending angle and refractivity profiles.

The following command line options can be used with `ropp_pp_invert_tool` tool:

<code>-h</code>	give help menu
<code>-o <outfile></code>	name of netCDF ROPP format output filename
<code>-c <config_file></code>	configuration parameters filename
<code>-m <method></code>	ionospheric correction method [NONE,MSIS,GMSIS,BG,BARO,GBARO] (default GMSIS)
<code>-mfile <mfile></code>	climatology coefficients filename (default local search for <code>MSIS_coeff.nc</code>)
<code>-bfile <bfile></code>	background refractivity profile (if using <code>-m BG</code>)
<code>-d</code>	output additional diagnostics (VerboseMode)
<code>-v</code>	output version information

The method used to conduct the ionospheric correction can be specified as a command-line option using the `-m` flag. By default, or by using `-m GMSIS` or `-m GBARO`, `ropp_pp_invert_tool` applies the statistical optimization and ionospheric correction algorithm described by Gorbunov (2002a), using the best fitting member of the MSISE-90 or BAROCLIM global climatology for both the ionospheric correction and Abel inversion processing. Specifying `-m MSIS` or `-m BARO` also applies the statistical optimization and ionospheric correction, but using the *local* MSIS-90 or BAROCLIM climatology profile for the observation location and time of year. Specifying `-m BG` uses a background refractivity profile read in from the file given with the `-bfile` option. Specifying `-m NONE` as a command line option reverts to the linear combination of L1 and L2 bending angles (Equation (2.39)) and no climatological or background model data are used for the Abel inversion.

4.1.2 Code organisation

Figure 4.2 shows how `ropp_pp_invert_tool` is composed of the following stages:

- **Input data access**

Setup the input data arrays, read the input data as a structure of type `R0data`. Define sub-structures of type `L1btype` to hold all bending angle data and structures of type `L2atype` to hold output refractivity data. Further details on the ROPP data structures can be found in the ROPP User Guide Part I (ROM SAF, 2016).

- **Initial data checks**

The `ropp_pp` software assumes that data are in ascending height order (i.e. data array index 1 is closest to the surface). This requirement is ensured by applying the `ropp_io` routine `ropp_io_ascend` to the

input data. The Abel transform processing also requires impact parameter data to be a monotonically increasing function. This condition is ensured using `ropp_pp_monotonous`. Finally L1 and L2 bending angle data are considered on the same set of impact parameter levels, by defining a standard set of levels and interpolating the bending angle data to the standard grid in `ropp_pp_merge_profile`.

- **Ionospheric correction of bending angles by linear combination (LC)**

Equation (2.39) is applied to the input data on the standard impact parameter grid by calling `ropp_pp_linear_combination`. The resulting non-optimised bending angle profile is written to output variable `bangle` as a function of impact parameter `impact`. If the `-m NONE` command line option is used, only the non-optimized bending angle is computed.

- **Determine climatological bending angle profile**

A climatological bending angle profile is required for the statistical optimization and Abel inversion processing.

If configuration option `config%method = GMSIS` or `config%method = GBARO` is set, routine `ropp_pp_search_model_refraction` finds the closest bending angle profile to the observations from the full MSIS-90 or BAROCLIM climatology. An alternative method of doing the same thing, which is activated by setting the configuration parameter `sf_method = regular` (the default is `sf_method = convoluted`), is carried out in `ropp_pp_search_model_refraction_new`.

If configuration option `config%method = MSIS` is used, `ropp_pp_model_refraction` derives a bending angle profile for the location and month of the occultation from spherical harmonic coefficients held in an 'ancillary' data file which must have the same format as the example file `MSIS_coeffs.nc`, which is distributed in the `ropp_pp/data` directory.

If configuration option `config%method = BARO` is used, `ropp_pp_model_refraction` derives a bending angle profile for the location and month of the occultation from zonal harmonic coefficients held in an 'ancillary' data file which must have the same format as the example file `BAROCLIM_coeffs.nc`, which is distributed in the `ropp_pp/data` directory.

Note that `MSIS_coeffs.nc` and `BAROCLIM_coeffs.nc` have different structures, not least because there is no longitudinal variation in the BAROCLIM coefficients. Therefore one cannot, for example, run `-m BARO` with `-mfile MSIS_coeff.nc`.

To avoid systematic deviations of the observed profile from the climatology (MSIS or BAROCLIM), the model profile is scaled by a fitting coefficient in `ropp_pp_fit_model_refraction`. An alternative method of doing the same thing, again activated by setting the configuration parameter `sf_method = regular`, is carried out in `ropp_pp_fit_model_refraction_new`. Either fitting process may be implemented using a one- or two-parameter fitting method. The scaling coefficients are computed from regression of the climatological profile onto the smoothed observed bending angle profile.

- **Perform ionospheric correction and statistical optimization**

The Gorbunov (2002a) ionospheric correction with statistical optimization algorithm is implemented in `ropp_pp_ionospheric_correction`. This derives the elements of each matrix in Equation (2.40) and solves the equation to determine the neutral atmosphere and ionospheric bending angle profiles. The

optimized bending angle profile is output as variable `bangle_opt` as a function of impact parameter `impact_opt`. The ionospheric bending angle profile and covariances of the solutions can be output as additional diagnostics in the output file.

- **Perform Abel inversion to derive the refractivity profile**

Routine `ropp_pp_invert` performs the Abel inversion (Equation (2.45)) to derive a refractivity profile from the corrected bending angles. By default, this is called from `ropp_pp_invert_refraction` which first extends the corrected bending angle profile with the climatological bending angles (see above) to reduce the impact of the upper boundary condition in Equation (2.45) on refractivity values in the upper part of the output profile.

- **Integrate refractivity to derive dry temperature**

If `config%output_tdry` is `.TRUE.`, then routine `ropp_pp_tdry` performs the hydrostatic integration to derive a temperature profile from the refractivity profile, ignoring the contribution of water vapour. (Hence `config%output_tdry` really means “`config%calculate_tdry`”, but the old name is retained for back-compatibility with those configure files that still use `output_tdry`.)

- **Write results to generic RO data structure and output file**

4.2 Configuration options

A number of configuration options can be defined by the user in order to tune the ionospheric correction algorithm. Table 4.1 lists the configuration options and their default values held in a structure of derived type `PPConfig` (see also Table 3.1). The use of these parameters within `ropp_pp` are described within this User Guide. A user can specify configuration options at run-time by setting their values in a configuration file and including the ‘`-c <config_file>`’ command line option when running `ropp_pp_invert_tool`. A sample configuration file is provided in the `config` directory of the `ropp_pp` distribution.

The configuration file is read, if specified, and the elements of a variable of type `PPConfig` are overwritten by calling `ropp_pp_read_config`.

```
USE ropp_pp
TYPE(PPConfig) :: config
CALL ropp_pp_read_config(config_file, config)
```

4.3 Observation data

Data are read in from a ROPP netCDF input file using the `ropp_io` module routine `ropp_io_read`. A single profile of observation data are read and fill elements of the generic ROPP data structure of type `R0prof` (ROM SAF, 2016). The Level 1b bending angle data as a function of impact parameter within `R0prof` are held in a structure of type `L1btype`.

PPConfig		
Structure element	Default	Description
...%method	GMSIS	Ionospheric correction method (NONE = linear combination, MSIS/GMSIS = full with local/global MSIS search, BARO/GBARO = full with local/global BAROCLIM search, BG = full with background profile)
...%so_method	so	Statistical optimisation method (so = statistical optimisation, lcso = linear combination+stat. opt.)
...%abel	LIN	Assume LINear or EXPonential variation between levels in Abel integral algorithm?
...%mfile	MSIS_coef.nc	Climatology coefficients filename (as used in stat. opt. and Abel inversion)
...%bfile	n/a	Background atmospheric profile filename
...%npoints	<i>(from input file)</i>	Number of input data points
...%r_curve	<i>(from input file)</i>	Local radius of curvature
...%pmin	<i>(computed)</i>	Minimum impact parameter (IP)
...%pmax	<i>(computed)</i>	Maximum impact parameter (IP)
...%dpi	100.0 m	Step of standard impact parameter grid
...%np_smooth	3	Polynomial order for smoothing
...%fw_smooth	1000.0 m	Filter width for smoothing profile
...%sf_method	convoluted	Search and fit method (convoluted or regular)
...%nparm_fit	2	Number of parameters used for model fit regression
...%hmin_fit	20 km	Lower limit for model fit regression
...%hmax_fit	70 km	Upper limit for model fit regression
...%omega_fit	0.3	<i>A priori</i> s.d. of regression factor
...%f_width	2000.0 m	Ionospheric correction filter width
...%delta_p	20.0 m	Step of homogeneous IP grid
...%s_smooth	2000.0 m	External ionospheric smoothing scale
...%z_ion	50 km	Lower limit of ionospheric signal
...%z_str	35 km	Upper limit of stratospheric signal
...%z_ltr	12 km	Upper limit of tropospheric signal
...%n_smooth	11	Number of points for smoothing (odd)
...%model_err	-0.5	<i>A priori</i> model error std dev. (If negative then use dynamical estimate)
...%ztop_invert	150 km	Height of atmosphere for inversion
...%dzh_invert	50.0 m	Step size of inversion IP grid
...%dzc_invert	20 km	Profile depth for inversion regression

Table 4.1: Configuration options held as elements of the PPConfig structure which are used by ropp_pp ionospheric correction and Abel transform routines. The default values are assumed unless overwritten by reading configuration options from an input file.

4.3.1 Data order

ropp_pp calculations assume that data arrays are defined with index 1 closest to the surface. The input data are checked and reordered as required using ropp_io_ascend.

Further, it is assumed that input impact parameters are monotonically increasing. Routine ropp_pp_monotonous is used to transform a series of impact parameter values to a monotonic sequence.

```

USE ropp_io
USE ropp_pp
TYPE(ROprof)   :: ro_data
TYPE(L1btype)  :: bangle
CALL ropp_io_read(ro_data, inputfile, rec=iprofile) ! read data
CALL ropp_io_ascend(ro_data)                       ! check array direction
bangle = ro_data%lev1b                             ! structure assignment
CALL ropp_pp_monotonous(bangle%impact_L1)          ! sort L1 impact parameters
CALL ropp_pp_monotonous(bangle%impact_L2)          ! sort L2 impact parameters

```

4.3.2 Standard impact parameter levels ropp_pp_merge_profile

The ionospheric correction and subsequent ropp_pp processing is conducted using L1 and L2 bending angle data interpolated onto the same output impact parameter level grid. This is achieved by defining a new output variable structure of type L1btype and calling ropp_pp_merge_profile, which defines a monotonically increasing equally spaced impact parameter grid of size

$$n_grid = 1 + \frac{\max(bangle\%impact_L1) - \min(bangle\%impact_L1)}{config\%dpi} \quad (4.1)$$

and interpolates L1 and L2 bending angle data on L1 and L2 impact parameter levels to the new grid. The user may vary the resolution of the output grid using the configuration parameter dpi (Table 4.1).

4.4 Linear combination ropp_pp_linear_combination

ropp_pp_invert_tool performs a simple ionospheric correction by linear combination of L1 and L2 bending angles using Equation (2.39). The calculation of a corrected bending angle profile bangle as a function of impact parameter impact is performed in ropp_pp_linear_combination.

```

USE ropp_pp
CALL ropp_pp_linear_combination(impact_L1, bangle_L1, impact_L2, bangle_L2,
                               impact,    bangle)

```

(If the configuration option config%method is set to NONE, this linear combination of the L1 and L2 bending angles provides the statistically optimised bending angle.)

4.5 Ionospheric correction

As default, ionospheric correction in `ropp_pp` is conducted using the approach devised by Gorbunov (2002a). Observed L1 and L2 channel bending angle data are combined with a climatological bending angle profile derived from the MSISE-90 model or the BAROCLIM climatology in a statistically optimal method by solving Equation (2.40).

4.5.1 Climatology bending angle data

By default, `ropp_pp_search_model_refraction` is called to retrieve the best-fit climatology bending angle profile to the observations from the MSISE-90 (or BAROCLIM) model on the required standard impact parameter grid. This is the 'global search' (`config%method = GMSIS` or `config%method = GBARO`) implementation.

An alternative 'regular' global search method, which differs from the 'convoluted' one effected by `ropp_pp_search_model_refraction`, is implemented in `ropp_pp_search_model_refraction_new`. The basic difference is that the 'regular' method, which is activated by setting `sf_method = regular` in the `PPConfig` configuration structure, does the search and then the fitting (see below), whereas the older, 'convoluted' method, which is activated by setting `config%sf_method = convoluted`, does the fitting within the search. `sf_method = convoluted` remains the default, for backwards compatibility, but `sf_method = regular` is recommended.

Alternatively, if a 'local search' option is required, (`config%method = MSIS` or `config%method = BARO`) then `ropp_pp_model_refraction` is called to retrieve the *local* climatological bending angle profile for the month, latitude and (for MSIS) longitude of the observation.

To avoid systematic deviations from the observe profile, the climatological profile `model_bangle` is scaled to observations by a fitting coefficient in `ropp_pp_fit_model_refraction`. (If `config%sf_method = regular`, the fitting is undertaken in `ropp_pp_fit_model_refraction_new`.) The fit to observations may be performed using either a 1 or 2-parameter fitting method, as defined by the `config%nparm_fit` configuration parameter. The scaling coefficients are computed by regression of the climatological profile onto a linear combination of smoothed observed bending angle profiles.

The flow diagram of the ionospheric correction routines is therefore as follows.

```
USE ropp_pp
TYPE(PPConfig) :: config
CALL ropp_pp_linear_combination(impact_L1,  bangle_L1,  &
                               impact_L2,  bangle_L2,  &
                               impact,     bangle)
CALL ropp_pp_smooth_profile(impact, bangle, smooth_ba, config)

SELECT CASE(config%method)
CASE ('GMSIS', 'GBARO')
  SELECT CASE (config%sf_method)
```

```
CASE ('regular')
  CALL ropp_pp_search_model_refraction_new(config%mfile,impact,smooth_ba, &
                                          out_impact, model_bangle, config)
CASE ('convoluted')
  CALL ropp_pp_search_model_refraction(config%mfile,impact,smooth_ba, &
                                          out_impact, model_bangle, config)
END SELECT
CASE ('MSIS', 'BARO')
  CALL ropp_pp_model_refraction(config%mfile, month, lat, lon, &
                                out_impact, model_bangle, config)
END SELECT

SELECT CASE (config%sf_method)
CASE ('regular')
  CALL ropp_pp_fit_model_refraction_new(impact, smooth_ba, &
                                          out_impact, model_bangle, config)
CASE ('convoluted')
  CALL ropp_pp_fit_model_refraction(impact, smooth_ba, &
                                          out_impact, model_bangle, config)
END SELECT
```

The purpose of these routines is as follows.

- **ropp_pp_smooth_profile**

Smoothing of the L1 and L2 bending angle profiles is achieved by least-square fitting a polynomial to the data in sliding windows. Configuration parameter `fw_smooth` controls the filter width for smoothing and `np_smooth` defines the order of the polynomial expression applied.

- **ropp_pp_search_model_refraction**

The climatological data are included in `ropp_pp` as a netCDF file of spherical harmonic (zonal harmonic for BAROCLIM) coefficients, which are read and translated into a bending angle profile from subroutine `ropp_pp_bangle_MSIS` or `ropp_pp_bangle_BARO`. This is repeated for each month of the year, and a selection of latitude and longitude values. The best-fit profile to observations over the height range `config%hmin_fit` to `config%hmax_fit`, using either a 1 or 2-parameter fitting method, is selected.

- **ropp_pp_model_refraction**

Alternatively, for a local search, the climatological spherical/zonal harmonic coefficients are read and translated into a bending angle profile by calling `ropp_pp_bangle_MSIS` or `ropp_pp_bangle_BARO`.

Note that is different to the behaviour of `ropp_pp_model_refraction` before ROPP9.0 (that is, pre-BAROCLIM). In earlier versions, the climatological refractivities were generated from the coefficients in the `MSIS_coeffs` file, and a forward Abel transform (Equation (2.44)) was then used to derive the

MSIS bending angle profile. In practice, this difference in the generation of the climatological bending angles causes but small differences in the output fields.

- **ropp_pp_fit_model_refraction**

A linear combination of smoothed L1 and L2 bending angle profiles is performed using `ropp_pp_linear_combination` to obtain a corrected bending angle profile on the output impact parameters, α_{smooth} . The model profile is then fitted to the corrected smoothed observations by linear regression to obtain a scaling factor. The linear regression is only carried out between heights defined as configuration options `hmin_fit` and `hmax_fit` (Table 4.1). Configuration parameter `nparm_fit` specifies the number of parameters for the regression algorithm.

For 1-parameter fitting, configuration parameter `omega_fit` specifies the *a priori* standard deviation of the regression factor.

$$rf = \frac{\frac{\sum_{i=hmin}^{i=hmax} \alpha_{smooth}(i) \alpha_{CLIM}(i)}{\sum_{i=hmin}^{i=hmax} \alpha_{CLIM}(i)^2} + (\sigma/\omega_{fit})^2}{1 + (\sigma/\omega_{fit})^2} \quad (4.2)$$

where σ is the computed standard deviation of the smoothed to climatological bending angle profiles. The α_{CLIM} are then multiplied by `rf`.

For 2-parameter fitting, the logarithms of the α_{CLIM} are linearly regressed onto the logarithms of the smoothed observational bending angles, α_{smooth} . The α_{CLIM} are then given by $\exp(rf_1 + rf_2 \log(\alpha_{smooth}))$.

- **ropp_pp_search_model_refraction_new** and **ropp_pp_fit_model_refraction_new**

If `config%sf_method = regular`, then these two routines are invoked. Interested users are referred to the codes themselves for full details, but in essence these routines perform the same operations as their older cousins, except for some changes in the searching and the fitting, and that the 2-parameter fit is no longer a linear one of $\log \alpha_{CLIM}$, but of its scaling factor s to a two parameter function which is piecewise constant except for a 'round-shouldered' ramp between $h_{min} = \text{config}\%hmin_fit = 20$ km by default (40 km is better) and $h_{max} = \text{config}\%hmax_fit = 70$ km by default (60 km is better), thus:

$$s(h) = s(h \leq h_{min}) \cos^2 \delta + s(h \geq h_{max}) \sin^2 \delta \quad (4.3)$$

where $\delta = (\pi/2) \max(0, \min(1, (h - h_{min}) / (h_{max} - h_{min})))$. (The regression is performed using data between h_{min} and h_{max} .) After a successful fit, the α_{CLIM} are multiplied by s .

4.5.2 Statistical optimization `ropp_pp_ionospheric_correction`

The ionospheric correction with statistical optimization described by Gorbunov (2002a) is performed by calling `ropp_pp_ionospheric_correction`. This computes the optimized bending angle profile `bangl_opt` as a function of impact parameter `impact_opt`.

```
USE ropp_pp
TYPE(PPConfig) :: config
```

```

TYPE(PPDiag)   :: diag
CALL ropp_pp_ionospheric_correction( &
  impact_L1, bangle_L1,      ! L1 data
  impact_L2, bangle_L2,      ! L2 data
  impact_model, model_bangle, ! CLIM (climatological 'model') data
  config, impact_opt, bangle_opt, diag)

```

The ionospheric correction computation is conducted using data on a homogeneous impact parameter grid of size `nh`, defined as in Equation (4.1) with the spacing set by parameter `config%delta_p`. The L1, L2 and CLIM bending angle profiles are then interpolated to the new grid using `ropp_pp_interpol`.

Two smoothing windows are defined based on configuration parameters `config%f_width` and `config%s_smooth`.

$$whi = \frac{config\%f_width * (config\%npoints - 1) * (nh - 1)}{(config\%pmax - config\%pmin) * (nobs - 1)} \quad (4.4)$$

$$wei = \frac{config\%s_smooth * (nh - 1)}{config\%pmax - config\%pmin} \quad (4.5)$$

Scale `whi` sets the filter width for ionospheric smoothing used to estimate the ionospheric bending angle (Equation (8) in Gorbunov (2002a)). Scale `wei` sets a stronger smoothing filter used to compute the low-frequency component of ionospheric bending angle (Equation (6) in Gorbunov (2002a)).

The data are considered in distinct regions defined by configuration parameters `config%z_ion`, `config%z_str` and `config%z_ltr` (Table 4.1).

The outline stages of `ropp_pp_ionospheric_correction` are as follows.

- Compute difference between observed and fitted CLIM bending angles

$$\Delta\alpha_{L1} = \alpha_{L1} - \alpha_{CLIM} \quad (4.6)$$

$$\Delta\alpha_{L2} = \alpha_{L2} - \alpha_{CLIM} \quad (4.7)$$

- Smooth $\Delta\alpha_{L1}$ and $\Delta\alpha_{L2}$ with sliding window width `wei` to obtain $\Delta\bar{\alpha}_{L1}$, $\Delta\bar{\alpha}_{L2}$ (call `ropp_pp_filter`).
- Compute low frequency component of ionospheric bending angle for L1 channel.

$$\bar{\alpha}_I = \frac{f_2^2}{f_2^2 - f_1^2} (\Delta\bar{\alpha}_{L1} - \Delta\bar{\alpha}_{L2}) \quad (4.8)$$

- Smooth $\Delta\alpha_{L1}$ and $\Delta\alpha_{L2}$ with sliding window width `whi` to obtain $\Delta\tilde{\alpha}_{L1}$ and $\Delta\tilde{\alpha}_{L2}$.
- Estimate ionospheric noise covariance matrix components using all data above `config%z_ion`

$$C_N = \begin{pmatrix} \frac{1}{2N_{ion}} \sum (\Delta\tilde{\alpha}_{L1} - \Delta\tilde{\alpha}_{L2}(f_2/f_1))^2 & 0 \\ 0 & \frac{f_1^4}{2N_{ion}f_2^4} \sum (\Delta\tilde{\alpha}_{L1} - \Delta\tilde{\alpha}_{L2}(f_2/f_1))^2 \end{pmatrix} \quad (4.9)$$

- Estimate ionospheric signal covariance using all data above `config%z_ion`

$$\sigma^{IS} = \frac{1}{N_{ion}} \sum \left(\frac{1}{2} (\Delta\tilde{\alpha}_{L1} + \Delta\tilde{\alpha}_{L2}(f_2/f_1)) - \bar{\alpha}_I \right)^2 - C_N(1,1) \quad (4.10)$$

- Estimate the relative neutral signal covariance using data between `config%z_ltr` and `config%z_str`.

$$\sigma^{NS} = \frac{1}{N_{neut}} \sum \left(\frac{\Delta\alpha_{L1}}{\alpha_{CLIM}} \right)^2 \quad (4.11)$$

If `config%model_err` is positive and equal to σ_{static} , say, then use the fixed estimate

$$\sigma^{NS} = \sigma_{static} \quad (4.12)$$

instead.

- Estimate ionospheric signal covariance matrix

$$C_S = \begin{pmatrix} \sigma^{NS} \alpha_{CLIM}^2 & 0 \\ 0 & \sigma^{IS} \end{pmatrix} \quad (4.13)$$

- Compute quasi-inverse matrix

$$\bar{K}^{-1} = (K^T C_N^{-1} K + C_S^{-1})^{-1} K^T C_N^{-1} \quad (4.14)$$

where $K_{11} = K_{12} = K_{21} = 1$ and $K_{22} = (f_1/f_2)^2$.

- Calculate neutral bending angle α_N and ionospheric bending angle α_I by using \bar{K}^{-1} in Equation (2.40).

The Gorbunov (2002a) method results in a number of output diagnostics in addition to the corrected neutral bending angle profile required. These are passed to elements of a structure of type `PPDiag`, which holds (among other things): the ionospheric bending angle (`diag%ba_ion` = α_I , as given by the bottom row of Equation (2.40)); the neutral bending angle error covariance (`diag%err_neut` = $(K^T C_N^{-1} K + C_S^{-1})_{11}^{-1}$); the ionospheric bending angle error covariance (`diag%err_ion` = $(K^T C_N^{-1} K + C_S^{-1})_{22}^{-1}$); and the fraction of observed data (ie the ratio `data:data+clim`) at each point of the profile (`diag%wt_data` = $\bar{K}_{11}^{-1} + \bar{K}_{12}^{-1}$). If `config%output_diag` = `.TRUE.`, these additional variables are written to the output file using the `ropp_io_addvar` function (ROM SAF, 2016). The full composition of the `diag` structure is given in Table 4.2.

4.6 Inversion

A refractivity profile `refrac` is computed on geopotential height levels `geop_refrac` by applying an inverse Abel transform to the ionospheric corrected bending angle profile (Equation (2.45)). If the configuration parameter `config%method` is set to `NONE`, an inverse Abel routine is called to compute refractivity from the linear combination of the L1 and L2 bending angles which constitutes the ionospherically corrected bending angle in this case. But by default, the corrected bending angle data are combined with climatological bending angles above the top of the observed profile to limit the effect of the infinite upper boundary condition in Equation (2.45) on data quality in the region of interest. This is achieved by calling `ropp_pp_invert_refraction`.

PPDiag	
Structure element	Description
...%CTimpact	CT processing impact parameter (m)
...%CTamplitude	CT processing amplitude
...%CTamplitude_smt	CT processing smoothed amplitude
...%CTimpactL2	CT processing L2 impact parameter (m)
...%CTamplitudeL2	CT processing L2 amplitude
...%CTamplitudeL2_smt	CT processing smoothed L2 amplitude
...%ba_ion	Ionospheric bending angle in L1 (rad)
...%err_neut	Error covariance of neutral bending angle (rad ²)
...%err_ion	Error covariance of ionospheric bending angle (rad ²)
...%wt_data	Weight of data (data:data+clim) in profile
...%sq	SO badness score: $\text{MAX}[\text{err_neut}^{1/2}/\alpha_N] \times 100\%$
...%L2_badness	L2 phase correction badness score

Table 4.2: Elements of PPDiag structure

```
USE ropp_pp
TYPE(PPConfig)
CALL ropp_pp_invert_refraction(config%infile, month, lat, lon, &
                             impact_opt, bangle_opt, &
                             geop_refrac, refrac, config)
```

4.6.1 Climatology bending angle data

The ionospheric corrected bending angle profile is extended using climatology between the observed profile top and an altitude specified by `config%ztop_invert`. A climatological bending angle profile is derived following the method described in Section 4.5.1 where bending angles are retrieved on a homogeneous impact parameter grid with vertical spacing given by `config%dzh_invert`. In order to scale the α_{CLIM} profile to observations, a scaling factor is computed by regression of the observed data and a model profile above a height of `config%dzr_invert` below the observed profile top.

4.6.2 Inverse Abel algorithm

Two inverse Abel algorithms are provided in `ropp_pp`. In both cases the algorithms solve the inverse Abel integral (Equation (2.45)) by assuming that the input bending angle profile can be approximated as a known function of height between successive impact parameter levels, for which an analytical solution to the Abel integral can be found. Equation (2.45) is then solved for the refractive index n_i at a certain impact parameter a_i by summing contributions from the solutions to the known sub-integrals for each observation level between a_i and the top of the background profile, thus:

$$n_i = \exp \left[\frac{1}{\pi} \sum_{j=i}^n \Delta(\ln n)_j \right] \quad (4.15)$$

The sub-integrals can be evaluated in two ways.

- **ropp_pp_invert_lin** (config%abel=LIN)

If it is assumed that bending angles vary linearly between levels the Abel integral is solved using an analytical solution in routine `ropp_pp_invert_lin`. Contribution $\Delta(\ln n)_j$ is expressed as

$$\Delta(\ln n)_j = \frac{1}{\pi \Delta a_j} \times \left[(\alpha_j a_{j+1} - \alpha_{j+1} a_j) \ln \left(\frac{a_{j+1} + \sqrt{a_{j+1}^2 - a^2}}{a_j + \sqrt{a_j^2 - a^2}} \right) + (\alpha_{j+1} - \alpha_j) \left(\sqrt{a_{j+1}^2 - a^2} - \sqrt{a_j^2 - a^2} \right) \right] \quad (4.16)$$

The infinite upper boundary condition may be accounted for by adding a further correction to the summation of refractive index contributions, thus:

$$\Delta(\ln n)_{\text{top}} = \frac{\alpha_{\text{top}}}{\sqrt{\pi h'}} \exp \left(\frac{a_{\text{top}} - a_{\text{bot}}}{h} \right) \left[1 - \operatorname{erf} \left(\sqrt{\frac{a_{\text{top}} - a_{\text{bot}}}{h}} \right) \right] \quad (4.17)$$

where $h' = (a_{\text{top}} + a_{\text{bot}})/h$ and h is a scale height, computed as $h = -\Delta a / \Delta(\ln \alpha)$ with differences in impact parameter and $\ln \alpha$ estimated across the upper part of the input bending angle profile.

- **ropp_pp_invert_exp** (config%abel=EXP)

If it is assumed that bending angles vary exponentially between levels Abel integral is solved in routine `ropp_pp_invert_exp` in terms of the error function. Contribution $\Delta(\ln n)_j$ is expressed as

$$\Delta(\ln n)_j = \frac{1}{\sqrt{2\pi x k_j}} \alpha_j \exp(k_j(a_j - x)) \left[\operatorname{erf} \left(\sqrt{k_j(a_{j+1} - x)} \right) - \operatorname{erf} \left(\sqrt{k_j(a_j - x)} \right) \right] \quad (4.18)$$

where

$$k_j = \frac{\ln \alpha_j / \alpha_{j+1}}{(a_{j+1} - a_j)} \quad (4.19)$$

In this case, the ray bending above the observation top is accounted for by extrapolating when $j+1$ is at the top of the profile and evaluating

$$\Delta(\ln n)_j = \frac{1}{\sqrt{2\pi x k_j}} \alpha_j \exp(k_j(a_j - x)) \left[1 - \operatorname{erf} \left(\sqrt{k_j(a_j - x)} \right) \right] \quad (4.20)$$

Further details on the implementation and performance of the Abel integral algorithms in `ropp_pp` are provided by ROM SAF (2008). They could be called with code such as the following.

```
USE ropp_utils
USE ropp_pp
TYPE(PPConfig) :: config
SELECT CASE(config%abel)
  CASE ('LIN')
    scale = -(impact(n)-impact(3*n/4))/(log(bangle(n))-log(bangle(3*n/4)))
```

```

CALL ropp_pp_invert_LIN(impact, bangle, impact, refrac, scale)
CASE ('EXP')
CALL ropp_pp_invert_EXP(impact, bangle, impact, refrac)
END SELECT
alt = ( impact / (1.0 + refrac*1.e-6) ) - config%r_curve      ! compute altitude
geop_refrac = geometric2geopotential(lat, alt)              ! conversion

```

4.6.3 Hydrostatic integration

The dry temperature, T_{dry} , (and corresponding dry pressure, P_{dry}), are obtained by ignoring the water vapour contribution to refractivity (Gorbunov et al., 2005).

Using the equation of state for an ideal gas and assuming hydrostatic equilibrium, the dry pressure at each level, z , is obtained by solving

$$\frac{d \ln P_{\text{dry}}}{dz} = f(z, \ln P_{\text{dry}}(z)) = \frac{-g(z)N(z)}{R\kappa_1 \exp(\ln P_{\text{dry}}(z))} \quad (4.21)$$

using a fourth order Runge-Kutta method. The gravitational acceleration, g , is a function of z and also depends on the reference latitude of the occultation — see the Geodesy Section of Appendix A for details. The refractivity constant $\kappa_1 = 77.60$ N-unit K hPa⁻¹ in ROPP — see Chapter 2 for more. R is the dry gas constant (287.05 J kg⁻¹ K⁻¹ in ROPP).

For the Runge-Kutta integration, the initial value of $\ln P_{\text{dry}}$ at z_{top} is calculated as

$$\ln P_{\text{dry}}(z_{\text{top}}) = \ln \left(\frac{-g(z_{\text{top}})N(z_{\text{top}})}{R\kappa_1(d \ln N/dz)} \right) \quad (4.22)$$

which assumes the boundary condition that $dT_{\text{dry}}/dz = 0$ at z_{top} . Example use:

```

USE ropp_utils
USE ropp_pp
TYPE(PPConfig) :: config
IF (config%output_tdry) THEN
CALL message(msg_info, "Computing dry temperature \n")
shum = 0.0_wp
CALL ropp_pp_tdry(lat, alt, refrac, shum, t_dry, p_dry, Ztop)
ENDIF

```

The dry temperature T_{dry} is then derived from the dry pressure P_{dry} and the refractivity N by using (cf Eqn (2.48))

$$T_{\text{dry}} = \kappa_1 P_{\text{dry}}/N. \quad (4.23)$$

References

Gorbunov, M. E., Radio-holographic analysis and validation of Microlab-1 radio occultation data in the lower troposphere, *J. Geophys. Res.*, *107*, 10.1029/2001JD000 889, 2002a.

Gorbunov, M. E., Ionospheric correction and statistical optimization of radio occultation data, *Radio Sci.*, *37*, 10.1029/2000RS002 370, 2002b.

Gorbunov, M. E., Lauritsen, K. B., Rhodin, A., Tomassini, M., and Kornblueh, L., Analysis of the champ experimental data on radio-occultation sounding of the earth's atmosphere, *Izvestiya, Atmospheric and Oceanic Physics*, *41*, 798–813, 2005.

ROM SAF, Abel integral calculations in ROPP, SAF/GRAS/METO/REP/GSR/007, 2008.

ROM SAF, The Radio Occultation Processing Package (ROPP) User Guide. Part I: Input/Output module, SAF/ROM/METO/UG/ROPP/002, Version 9.0, 2016.

A ropp_utils library

The `ropp_utils` library contains a collection of utility routines which are used by other ROPP modules. They are not intended to be called directly by user applications, so they are not documented in any detail here. This chapter gives only an overview of the library which is divided into sub-libraries by related functionality. The reader is directed to the ROPP Utils Reference Manual (SAF/ROM/METO/RM/ROPP/001).

A.1 Missing data values

The `ropp_utils` module defines a set of parameters to indicate and test a 'missing' or 'invalid' data value used by any ROPP routine. These are set in the main module file `ropp_utils.f90`.

- `ropp_MDFV` ($=-99999000.0$) is used to set missing (invalid) data for real ROPP parameters.
- `ropp_MDTV` ($=-9999.0$) is used for testing invalid real parameter values. Any real number less than this value can be assumed to be 'missing'.
- `ropp_ZERO` ($=0.0$) is used to set parameters to zero.
- `ropp_ZDTV` ($=1.0e-10$) is used to test for (almost) zero values.
- `ropp_MIFV` ($=-999$) is used to set missing (invalid) data for integer ROPP parameters.
- `ropp_MITV` ($=-99$) is used for testing invalid integer parameter values. Any integer less than this value can be assumed to be 'missing'.

A.2 ropp_messages

These routines provide an interface to write information and error messages to `stdout` or `stderr` from ROPP routines. The utilities were originally written by Christian Marquardt as personal code, independent of the ROM SAF. The author grants a free-use licence for all of this code. The utilities have since been modified and extended for ROPP.

A `msg_MODE` parameter is used to control the level of output diagnostic information output by ROPP routines. The available options are

- `QuietMode` - only output error messages to `stdout`, no info/warnings
- `NormalMode` - output all info and warnings and errors to `stdout`
- `VerboseMode` - as `NormalMode`, but also output diagnostic/debug messages to `stdout`

The required `msg_MODE` may be altered either within a program routine, e.g.

```
USE messages
```

```
msg_MODE = VerboseMode          ! Enable all messages
```

```
CALL message(msg_diag, ‘‘The result is...’’)  
msg_MODE = NormalMode           ! Re-set to normal output level
```

or by implicitly setting the default value in the `ropp_messages/messages.f90` file before compiling, or by setting the environment variable `ROPP_MSG_MODE` on the command line. Note that `VerboseMode` can be selected when running any of the stand-alone tools provided with ROPP by running it with a ‘-d’ command-line option.

With effect from ROPP8.0, the interface to these routines has been extended to ensure that the unix return code `$?` produced by running any of the ROPP tools equals (0, 1, 2, 3) if the tool executed (OK, with at least one warning, with at least one error, with a fatal error) respectively.

A.3 Unitconvert

A collection of low-level F90 routines for converting data between standard units used within ROPP modules. The conversion scaling factors and offsets for a given unit conversion operation are defined in routine `ropp_unit_conversion.f90`. The set of available unit conversions provided may be extended by a user as required. These unit conversion routines are called automatically from within `ropp_io` module read and write routines, and from within `ropp_fm` and `ropp_ldvar` routines to ensure variables have required units.

A.4 Coordinates

A collection of low-level F90 coordinate manipulation routines. Functionality includes routines to convert cartesian position vectors between Earth Centred Fixed and Earth Centred Inertial reference frames, convert between cartesian and geodetic position description, compute impact parameter, occultation point, radius of curvature and undulation (ie, difference between the EGM96¹ geoid and the WGS84² ellipsoid). Vector manipulation routines (vector product, vector angle, rotation) are also included.

A.5 Datetime

A collection of low-level F90 date and time conversion routines. The utilities were developed within the Met Office outside the context of the ROM SAF and represents pre-existing software (PES). This code is Crown copyright.

A.6 Geodesy

A collection of low-level F90 geodetic conversion routines to convert to/from geometric/geopotential heights and compute Earth’s effective radius and gravity. These routines are based on Somigliana’s equation to compute height scales relative to the WGS-84 reference ellipsoid.

¹ See <http://cddis.nasa.gov/926/egm96/egm96.html>.

² See <http://earth-info.nga.mil/GandG/wgs84/index.html>.

A.7 Arrays

A collection of low-level F90 array manipulation routines supporting all data types. The utilities were written by Christian Marquardt as personal code, independent of the ROM SAF. The author grants a free-use licence for all of this code.

A.8 Misc

Miscellaneous utilities used by other ROPP modules. `FileDelete.f90` and `GetIOUnit.f90` are low-level F90 file handling routines. `ToUpper.f90` and `ToLower.f90` are low-level F90 string handling routines.

A.8.1 `typeSizes`

`typeSizes.f90` is a public-domain F90 module written by Robert Pincus (Cooperative Institute for Meteorological Satellite Studies, Madison) which provides named kind parameters for use in declarations of real and integer variables with specific byte sizes. It is a copy of the same file included in the 3rd party `netCDF` package, but is bundled with, and used by, ROPP as a stand-alone tool to provide a standardised type naming convention. This is 'freeware' provided complete and 'as-is' under the terms of usage. Users of `ropp_utils` must respect the same conditions in turn.

B Installing and using ROPP

B.1 Software requirements

ROPP is written in standard Fortran 95. Thus, compilation and use of the routines forming ROPP require the availability of standard ISO-conforming compilers. Fortran 95 was preferred over Fortran 90 because it has a number of convenient features. In particular, it allows elemental functions and pointers can be nullified when they are declared.

B.2 Software release notes

The latest ROPP distribution is available for download via the ROM SAF website <http://www.romsaf.org>. The ROPP Release Notes available from the ROPP download page and provided with the main ROPP download tarfile gives instructions for unpacking and installing the complete ROPP package, or individual modules. Users are strongly recommended to refer to the ROPP Release Notes and use the build and configure tools described therein. The information contained here are intended to complement the ROPP Release Notes. Where any contradiction between the User Guide and ROPP Release Notes exist, the ROPP Release Notes page is considered to be the most up-to-date latest information.

B.3 Third-party packages

To fully implement ROPP, the code uses some standard third-party packages. These are all non-commercial and cost-free. Note that third-party codes are only needed by the `ropp_io` and `ropp_pp` modules, so are optional if these modules are not required by the user.

All third-party code or packages used by ROPP are, by definition, classed as 'Pre-Existing Software' and all rights remain with the originators. Separate rights licences may be part of these distributions — some may have a licence which may impose re-distribution restrictions — and such licences must be adhered to by users.

If a third-party package is required, this must be built and installed before attempting to build the ROPP code. For convenience, these packages should be installed to the same root path as ROPP. It is highly recommended that the package is compiled using the same compiler and using the same compiler flags as will be used to build the ROPP code. Example configure scripts for supported compilers are provided in the `ropp_build` module available from the ROPP download website. See Section B.4 for further details.

B.3.1 NetCDF

The input/output library `ropp_io` uses Unidata's `netCDF` data format. Thus, the `netCDF` library and its associated utility programs (like `ncdump`, `ncgen`) are required and must be properly installed on the user's system before the compilation of the `ropp_io` package can be attempted. `netCDF` may also be used for reading MSIS or BAROCLIM climatology data as part of the `ropp_pp` module.

The SAF provides versions of the `netCDF` distribution, which have been successfully integrated with ROPP, alongside the ROPP distribution. This may not be the most recent distribution. Latest versions are freely available from

<http://www.unidata.ucar.edu/software/netcdf/>

Note that the `tests` subdirectory of the `ropp_io` distribution contains a simple test to check if the `netCDF` installation works; see Section B.7 for details.

A very useful complementary set of tools for handling and manipulating `netCDF` data files are the `netCDF Operators nco`.¹ While the latter are not required for using ROPP libraries and sample applications, we highly recommend them.

Some example and test programs provided with the `ropp_pp`, `ropp_fm` and `ropp_1dvar` packages read data via `ropp_io`. A complete installation of the `ropp_io` library is therefore required if the test programs or one of the sample applications are to be run. As a consequence, the complete installation of these packages also requires the availability of `netCDF`. Note, however, that the libraries `libropp_pp.a`, `libropp_fm.a` and `libropp_1dvar.a` can be compiled and installed without `ropp_io` and thus `netCDF`; the configuration script will recognise the absence of these libraries and only compile and install the core pre-processor, forward model or 1DVar routines (i.e. those with no dependencies on `netCDF` or `ropp_io`). dependencies of each routine contained within the ROPP modules.

B.3.2 BUFR (optional)

The GNSS-RO BUFR encoder/decoder tools `ropp2bufr` and `bufr2ropp` in `ropp_io` require either the Met Office's 'MetDB' or the ECMWF BUFR library to be pre-installed. If neither BUFR library is detected by the installation configure script, then these tools will not be built.

The MetDB BUFR package is available without charge on request from the ROPP Development Team but with some licence restrictions. The ECMWF package is licenced under the GNU/GPL and can be downloaded from:

<http://www.ecmwf.int/products/data/software/bufr.html>

Both libraries generate essentially identical data when decoded (there may be non-significant round-off differences due to use of single- vs. double-precision interfaces). While the MetDB library is easier to install from a portability point of view, the ROPP `buildpack` script makes the ECMWF installation compatibly with ROPP more transparent. Therefore users can employ whichever BUFR package they prefer. Thus, the MetDB library could be built with

¹See <http://nco.sourceforge.net/>.

```
> buildpack bufr <compiler>
```

or

```
> buildpack mobufr <compiler>
```

while the ECMWF library would be built with

```
> buildpack ecbufr <compiler>
```

In order to install BUFR tables and related files, and for the applications to find them at run-time, an environment variable must be pre-defined to the path to these files. For instance, for the MetDB library:

```
> export BUFR_LIBRARY=<path>/data/bufr/
```

or for the ECMWF library:

```
> export BUFR_TABLES=<path>/data/bufr/
```

Note that in both cases, the path must currently be terminated with a '/' character, although this restriction will be relaxed for later (v20+) releases of the MetDB BUFR library. By default, the buildpack script will set <path> to be ROPP_ROOT.

B.3.3 GRIB_API (optional)

The GRIB background reading tool `grib2bgrasc` in `ropp_io` requires the ECMWF GRIB_API library to be pre-installed. If it fails to be detected by the installation configure script, then this tools will not be built.

The ECMWF GRIB_API package is licenced under Apache (2.0), and can be downloaded from:

<https://software.ecmwf.int/wiki/display/GRIB/Releases>

The ROPP buildpack script allows installation of the GRIB_API by means of:

```
> buildpack grib <compiler>
```

B.3.4 netCDF4/HDF5 (optional)

The 'EUMETSAT-formatted' RO reading tools `eum2ropp` and `eum2bufr` in `ropp_io` require the installation of a netCDF4 library, with its attendant HDF5 and zlib libraries.

These can be found from

<http://www.unidata.ucar.edu/software/netcdf/>,

<http://www.hdfgroup.org/ftp/HDF5/releases>

and

<http://www.zlib.net/>

respectively.

The ROPP buildpack script allows installation of these libraries as follows:

```
> buildpack zlib <compiler>
> buildpack hdf5 <compiler>
> buildpack netcdf4 <compiler>
```

(They would need to be installed in this order, since each depends on the one before.) Note that the `ropp_io` tool `eum2bufc` has only been interfaced to the ECMWF BUFR library. Note too that the `zlib`, and possibly also the HDF5 libraries may already be installed as part of a standard Linux distribution, in which case the user need not of course build a local version.

B.3.5 RoboDoc (optional)

The ROPP Reference Manuals have been auto-generated using the RoboDoc documentation tool². All source code, scripts, etc. have standardised header comments which can be scanned by RoboDoc to produce various output formats, including LaTeX and HTML. If code (and in particular the header comments) is modified, RoboDoc can optionally be used to update the documentation. This tool is not required in order to build the ROPP software.

B.3.6 autoconf and automake (optional)

The `automake` and `autoconf` tools, common on most Linux and Unix systems, are not necessary to build the ROPP package as provided, but are useful if any modifications are made to the code or build systems to re-generate the package `configure` files. Versions at, or higher than, v1.9 are required to support some of the `m4` macros defined in the ROPP build system.

B.4 BUILDPACK script

The ROPP package distribution includes a collection of `configure` and `build` scripts for a number of compilers and platforms suitable for ROPP and the dependency packages. A top-level BASH shell script `buildpack` is provided which may be used to automate the build of any ROPP module or dependency package in a consistent way, using the appropriate `configure` scripts. Use of `buildpack` is therefore highly recommended for first time build and less experienced users. Summary usage can be obtained using

```
> buildpack -h
```

In general, to build and install a package,

```
> buildpack <package> <comp> [[NO]CLEAN]
```

²See <http://rfsber.home.xs4all.nl/Robo/robodoc.html>.

where `<package>` is one of the supported package names (e.g. `ropp_fm`, `ropp_io`, `netcdf`, `mobuf`, `ecbuf`, etc.) and `<comp>` is the required compiler (e.g. `ifort`, `nag`, `xlf95`, etc.).

The `buildpack` script assumes that all tarball files and configure scripts provided with the ROPP distribution are placed in the same working directory. Packages will be decompressed here and installed to the `ROPP_ROOT/<comp>` target directory. The script automates the `configure — make — make install` build cycle described below. Further information on the `buildpack` script are provided in the ROPP Release Notes.

Other shell scripts `build*` are provided which can be used to further automate the build process by calling `buildpack` with a pre-determined sequence of packages or compilers. Users should review and edit these to suit their requirements.

B.5 Building and installing ROPP manually

The low-level build sequence performed by `buildpack` may be implemented manually by more experienced users. After unpacking, all packages are compiled and installed following the `configure — make — make install` cycle.

1. First run the command `configure` to check for the availability of all required libraries. `configure` allows the user to specify compiler options, paths to libraries and the location where the software shall eventually be installed, on the command line or as environment variables. Based on this information, `configure` generates user specific Makefiles, allowing a highly customised configuration and installation of the software.
2. Compilation is then initiated with the command `make`.
3. If building the software was successful, a `make install` will install libraries, header and module files as well as any executables in the directories specified by the user via the `configure` step.

Note that the ROPP modules partially depend on each other. In particular, all packages require that `ropp_utils` has been installed successfully. This package therefore needs to be compiled and installed first. Most packages make use of the `ropp_io` package for sample applications and testing, and should therefore be installed next if these are required. Note that users wishing to use ROPP source code directly in their own applications need not install the `ropp_io` module. If the `ropp_io` module is not available at build time, only the source code libraries will be compiled. We thus recommend the following build order:

- i) Third-party packages
- ii) `ropp_utils`
- iii) `ropp_io` (if required)
- iv) `ropp_pp` (if required)
- v) `ropp_apps` (if required)
- vi) `ropp_fm` (if required)
- vii) `ropp_1dvar` (if required)

Note that *all* libraries need to be built with the same Fortran compiler, and preferably with the same version of the compiler as well.

Supported Fortran (and C) compilers are listed in the Release Notes distributed with the ROPP package.

B.5.1 Unpacking

Once the required third-party software packages have been installed successfully, the ROPP packages can be installed. The complete ROPP package and individual modules are distributed as gzipped tar (.tar.gz) files. The complete package file name consists of the version name (e.g. ropp-8.1.tar.gz). This file contains the complete ROPP distribution. The module file names consist of the package's name (e.g. ropp_utils) and version (e.g. 8.1), as in ropp_utils-8.1.tar.gz. If GNU tar is available (as on Linux systems), gzipped tar files can be unzipped with

```
> tar -xvzf ropp-8.1.tar.gz
```

Older, or non-GNU, versions of tar might need

```
> gunzip -c ropp-8.1.tar.gz | tar -xv
```

In all cases, a new subdirectory named (in the above example) ropp-8.1 will be created which contains the source code of the complete package.

B.5.2 Configuring

Details on the installation procedure for the individual packages can be found in the files README.unix and README.cygwin for the installation under Unix and Windows (with Cygwin), respectively. Here, we provide a brief example for a Unix or Linux system.

Unpacking the ropp_build package will create the configure/ sub-directory containing a number of mini-scripts for local build configuration. The files have names <package>_configure_<compiler>_<os> where <package> is the package name (ropp, netcdf), <compiler> is the compiler ID (ifort, nag, pgf, g95...) and <os> is the operating system ID, as output by the uname(1) command but entirely in lower case (linux, cygwin, hp-ux...). Note these configure mini-scripts are also used by the high-level buildpack script. The example configure scripts for specific platforms and compilers may need to be edited for optimal local use, or users may create their own following one of the examples.

The main configure scripts provided assume that the external libraries are all installed under /usr/local, i.e. the libraries can be found in the directory /usr/local/lib, and header and module files in /usr/local/include. It is further assumed that the individual ROPP packages are to be installed under \$ROPP_ROOT, i.e. libraries are intended to reside in \$ROPP_ROOT/lib, programs in \$ROPP_ROOT/bin, and headers and module files in \$ROPP_ROOT/include. The \$ROPP_ROOT location should be specified as an environment variable, e.g,

```
> export ROPP_ROOT=$HOME
```

For most compilers, this means that the two paths to the header and module files need to be specified via the proper compiler options – usually via the -I option. The linker also needs to know where libraries are

located; on most Unix systems, this can be achieved by specifying the `-L` option at link time. Users are referred to the examples provided in the `configure` package for further details.

Running the appropriate script from `configure/` will set the required compiler flags and specify the header, module and library paths before running the `configure` script. For example if the Fortran 95 compiler is named (say) `ifort`, the following command would be sufficient to configure a package for later compilation:

```
> cd ropp_<module>
> ../configure/ropp_configure_ifort_linux
```

The `configure` script will check for all required libraries and add the required options for the linker. If `configure` is not successful finding the required libraries, an error message will be produced, and further compilation will not be possible. Should the configuration step fail entirely, the file `config.log` created during the run of `configure` usually gives some clues on what went wrong; the most likely reason for failing is that compiler or linker options (and in particular paths to include files or libraries) are not set correctly.

Note that `ropp_io` may optionally use other external libraries in order to support additional features. For example, the `ropp_io` library will provide two conversion tools from ROPP to BUFR and back if a supported BUFR library is found. The existence of such additional libraries is also checked during `configure`. If these libraries are missing, however, the installation can nevertheless proceed – the parts related to the missing library are simply not built. Should the build process fail in correctly finding the, e.g., BUFR libraries and therefore not build the BUFR tools, `config.log` should again provide evidence on what went wrong.

B.5.3 Compiling

If configuration was successful, the software can be built with the command

```
> make
```

This will compile all relevant source code, but may take several minutes. The resulting object library archive will be located in the `build` subdirectory. It will be named similar to the package following usual Unix conventions; for example, the `ropp_utils` library is named `libropp_utils.a`. Sample applications and test programs or scripts will also have been built in the relevant subdirectories. Sample and test runs can be performed without installing the software; for details on available test programs, see B.7.

Currently supported Fortran compilers include (on Linux unless otherwise stated): Intel's `ifort` (v9.x, v10.x and v11.x, XE (v12.x), v13.1.3); NAG's `f95` (v5.1), `nagfor` (v5.2); Portland Group's `pgf95` (v7, v11); SUN's `sunf95` (v8 with SunStudio 12); GNU's `g95` and `gfortran` on Linux and under Cygwin on MS Windows; IBM's (`xlf95`) on AIX. This list may be contracted or extended in the future. For a full list please refer to the ROPP Release Notes and README files in each sub-package.

B.5.4 Installing

After building the software successfully, the command

```
> make install
```

will install libraries in `{prefix}/lib`, Fortran modules in `{prefix}/include`, and any application programs in `{prefix}/bin`. Here, `{prefix}` is the prefix directory given as argument to the `--prefix` option of the `configure` command. In the above example, this would be the home directory. If no `--prefix` is given, the installation root directory defaults to `/usr/local` which would normally require root (sudo) privileges.

B.5.5 Cleaning up

The temporary files created during the compilation of any ROPP package can be removed from the package directory tree with

```
> make clean
```

Note that this will keep the information gathered during configuration as well as the build libraries and executables intact. Thus, a new build can be attempted using `make` without the need for another `configure`. To remove all data related to the build and install process, run

```
> make distclean
```

which will restore the original state of the unpacked package, but with all potential user modifications to the source code still in place.

If the software has been installed previously, but shall be removed from the user's computer, this can be accomplished with the command

```
> make uninstall
```

performed in the source code distribution directory. Note that this requires a configuration which is identical to the one used for the original installation of the software. It is not necessary to rebuild the software again before uninstalling it.

B.6 Linking

If one (or more) ROPP packages have been installed successfully, linking your application's code against the ROPP libraries requires the specification of all ROPP and all external libraries. For example, to create an executable from your own `application.f90` and the `ropp_io` libraries, something like

```
> ifort -o application application.f90 -L/usr/local/lib -L$ROPP_ROOT/lib \
-L$ROPP_ROOT/lib64 -lropp_io -lropp_utils -lnetcdf (-lnetcdf)
```


will be required. (Since netCDF-4.1.1, the netCDF C and Fortran routines have been split, with the latter held in `libnetcdf.a`. Hence, if compiling Fortran routines against a recent version of netCDF, `-lnetcdf` must be included in the list of libraries to be linked.)

B.7 Testing

The ROPP software has undergone formal testing before distribution, as will all future modifications and improvements. A subset of the test procedures and some reference files are provided with the source code in order to facilitate quick tests whether the compilation was completed successfully. Users can run these tests to ensure that there are no major problems. It should be kept in mind, though, that not all of the functionality of the corresponding package is fully tested. Note also that several of the test scripts attempt to run IDL to verify the output and display test results using standard viewer utilities (e.g. `xv`). (Future releases may use Python instead.) Tests that generate graphical output automatically display a corresponding reference figure (part of the distribution), against which the test result can be compared. By setting the environment variable `$ROPP_PAUSE` to `TRUE`, the user can examine the two figures at leisure, before allowing the test script to move on to the next figure by hitting any key.

B.7.1 `ropp_utils`

Tested as part of the other modules, mainly with `ropp_io`.

B.7.2 `ropp_io`

The subdirectory `tests` of the `ropp_io` distribution contains several test programs and scripts to test various aspects of the software. A test is provided to check the user's installation of the netCDF library. They can be run after a successful compilation of the `ropp_io` package with

```
> make test_netcdf
```

from within the `tests` subdirectory. The program executed for this test does not use `ropp_io`, but is exclusively based on the native Fortran 90 interfaces for netCDF. Failure of this test strongly indicates that there is a problem with the installation or setup of the external library, which needs to be fixed before `ropp_io` can be used.

A second test can be run with

```
> make test_ropp
```

which runs a script performing several conversions between ROPP data files. Running this test through `make` has the advantage that the results of the conversions are interpreted properly and result in 'success' or 'failure' messages.

If a supported BUFR library is available, the `tests` subdirectory will also contain a test script for the two programs `ropp2bufr` and `bufr2ropp` which convert ROPP data files to and from BUFR format data files. Issuing the command

```
> make test_bufr
```

will run a number of conversions and provide some verbose information on the content of the BUFR files and the encoding and decoding process. The script finally also compares the results. Its output should be fairly self-explanatory. Note that due to limitations of the BUFR format, non-significant loss of precision may be detected and flagged as differences from the reference file; this is normal.

The `gfz2ropp` and `ucar2ropp` tools to convert GFZ native text files or UCAR netCDF files to ropp-standard netcdf are tested with the commands

```
> make test_gfz
> make test_ucar
```

The `grib2bgrasc` and `bgrasc2ropp` tools which extract background profiles from GRIB-format gridded data and convert to ascii format, and then convert this to a ROPP-format netCDF file, are tested with the commands

```
> make test_grib
> make test_bgrasc
```

The `eum2ropp` and `eum2bufr` tools to convert 'EUMETSAT-format' RO data into standard ROPP netCDF or BUFR files, are tested with the commands

```
> make test_eum
> make test_eumbufr
```

Finally, the command

```
> make test
```

will run all of the above described tests.

The test of the `ropp_io` library and tools can also be tested manually by running, for example,

```
> t_ropp2ropp -t -n
```

which will create a series of different files. These should be compared (e.g., using `diff`) according to the advice given through the program's execution. Users can safely ignore numerical differences in the order of the cutoff in the text representation of the ROPP data files. Also note that different file names will show up in the first line of the text representation of netCDF data files (files created by the test script with the extension `.cd1`) and can be ignored. The `test_ropp` target actually does the same, but interprets the differences between the files with the above issues in mind. Note that the output of `t_ropp2ropp` can be found in the file `t_ropp2ropp.log` when run through `make`.

B.7.3 ropp_pp

The subdirectory `tests` of the `ropp_pp` distribution contains testing software, to compare the geometric optic and wave optic processing with known output, check the consistency of the Abel integral routines and their inverses, and compare the ionospheric correction processing with known output. Run

```
> make test
```

to check if solutions agree with precalculated solutions to within expected small tolerances. If IDL is available on the user's machine, plots of the results are made and can be compared against reference plots. A table summarising the results of the tests is written to `stdout` after they have all run.

B.7.4 ropp_apps

The subdirectory `tests` of the `ropp_apps` distribution contains testing software, to calculate tropopause height, and planetary boundary layer height, from a variety of profile data: bending angles, refractivities, background temperatures etc. Run

```
> make test
```

to check if solutions agree with precalculated solutions to within expected small tolerances. A table summarising the results of the tests is written to `stdout` after they have all run.

B.7.5 ropp_fm

The subdirectory `tests` of the `ropp_fm` distribution contains testing software. Run

```
> make test
```

to check if everything is working correctly. A series of tests are run to run the 1D and 2D operator applications to generate simulated refractivity and bending angle profiles, which are compared with precalculated data. Also included are tests of the consistency of the 1D and 2D tangent linear and adjoint routines. Warning messages are written to `stdout` if the operator, tangent linear and adjoint routines do not meet the expected consistency checks. If IDL is available on the user's machine, plots of the results are made and can be compared against reference plots. A table summarising the results of the tests is written to `stdout` after they have all run.

B.7.6 ropp_1dvar

A simple test is provided to check the correct running of the 1D-Var stand-alone application. This inputs a file of 'observations' (refractivity profiles) simulated from a set of ECMWF model background profiles. The same backgrounds are used in the 1D-Var retrieval. Hence the expected retrieved output profiles should be identical to the background (within rounding errors).

The subdirectory `tests` of the `ropp_1dvar` distribution contains the testing software. Run

```
> make test
```

to check if everything is working correctly. The results of each test are numerically compared to reference results, and a PASS/FAIL message issued to stdout if the differences are smaller/greater than some small tolerance. If IDL is available on the user's machine, plots of the results are made and can be compared against reference plots. A table summarising the results of the tests is written to stdout after they have all run.

B.8 Troubleshooting

If something goes wrong during the configuration step, carefully check the full output of the last unsuccessful `configure` run to get an idea why the software could not be built; this can be found in the file `config.log`. This also applies if parts of ROPP are not built (e.g. the BUFR tools), although the required additional libraries are available.

During compilation, warnings that indicate unused variables (e.g. with the NAG compiler) or the potential trimming of character variables (with Intel compilers) can safely be ignored. If compiling is fine, but installation fails, make sure you have write permissions on the installation directories.

If linking against ROPP libraries fails because of unresolved externals, make sure that *all* relevant libraries – *including all external ones* – are specified in the correct order (some linkers are not able to recursively browse through several libraries in order to resolve externals) with lower-level libraries following higher-level (ROPP) ones.

If the BUFR encoding or decoding fail with messages about missing run-time BUFR tables, check that the appropriate environment variable `BUFR_LIBRARY` (for the MetDB library) or `BUFR_TABLES` (for the ECMWF library) have been correctly set to the path of the installed BUFR tables, and that the path ends with a `'/'` character.

Forward modelling of, and retrievals using, L1 and L2 bending angles impose heavier memory requirements than the more standard use of neutral bending angles. Users should therefore be prepared to increase the memory available on their machines if using this feature.

If an ROPP module compiles and runs satisfactorily, but produces unexpected results, an easy first step in tracking down the problem is to print out extra diagnostic information. Most of the ROPP tools provide the facility to do this by means of the `'-d'` option. `ropp_pp`, `ropp_1dvar`, `ropp_apps` and `ropp_fm` also allow the user to add sets of pre-defined variables to the `R0prof` structure, which are written out in `netCDF` format with the usual variables. The first two modules do this by means of a option in a configuration file; the last two by means of a command line option in (some of) the tools. In fact, all ROPP modules allow the user to add specified variables to the `R0prof` structure in this way, by calling `ropp_io_addvar`, as described in the ROPP I/O user Guide. This obviously requires the code to be recompiled.

C ropp_pp program files

The `ropp_pp` module provides functions to pre-process amplitude and excess phase data and compute bending angles by geometric optics and wave optics (CT2) algorithms. Functions are provided to apply ionospheric correction to L1 and L2 bending angles to derive corrected bending angle refractivity profiles by combining measured data with climatological bending angle profiles. Refractivity and bending angle profiles are computed using the forward and inverse Abel transform. A tool to calculate the tropopause heights from profiles of bending angle, refractivity, dry temperature, or (wet) temperature (if available) is also included in this module.

Files listed in bold correspond to executable stand-alone tools. These call lower-level routines. In order to build this module the required packages must be first installed. Routines having additional dependencies on other packages or ROPP modules are listed with the required modules given in brackets. If the additional (optional) packages are not recognised by the configure script, only the core functions will be compiled and installed.

- Required packages: `ropp_utils`
- Optional packages: `ropp_io`, `netcdf`
- Stand-alone tools and test programs (*optional*)

tools/

ropp_pp_abel_tool.f90 (requires `ropp_io`)
ropp_pp_occ_tool.f90 (requires `ropp_io`)
ropp_pp_invert_tool.f90 (requires `ropp_io`)
ropp_pp_spectra_tool.f90 (requires `ropp_io`)
ropp_pp_grasrs2ropp.f90 (requires `ropp_io`)

tests/

t_ropp_pp.sh (requires `ropp_io`)
ropp_test_abel.f90 (requires `ropp_io`)

- Integrated code

bangle/

`ropp_pp_dct.f90`
`ropp_pp_bending_angle_go.f90`
`ropp_pp_bending_angle_wo.f90`

ropp_pp_geometric_optics.f90
ropp_pp_geometric_optics_adj.f90

icorr/

ropp_pp_fit_bg_refraction.f90
ropp_pp_fit_model_refraction.f90
ropp_pp_fit_model_refraction_new.f90
ropp_pp_invert_refraction.f90
ropp_pp_ionospheric_correction.f90
ropp_pp_linear_combination.f90
ropp_pp_merge_profile.f90
ropp_pp_model_refraction.f90
ropp_pp_search_model_refraction.f90
ropp_pp_search_model_refraction_new.f90
ropp_pp_smooth_profile.f90

abel_invert/

ropp_pp_abel_exp.f90
ropp_pp_abel_lin.f90
ropp_pp_invert_exp.f90
ropp_pp_invert_lin.f90

msis_bangle/

ropp_pp_msis.f90
ropp_pp_bangle_msis.f90
ropp_pp_refrac_msis.f90
ropp_pp_read_msis.f90 (requires ropp_io)

baro_bangle/

ropp_pp_baro.f90
ropp_pp_bangle_baro.f90
ropp_pp_read_baro.f90 (requires ropp_io)

common/

ropp_pp.f90
ropp_pp_fft.f90
ropp_pp_constants.f90
ropp_pp_copy.f90
ropp_pp_diag2roprof.f90
ropp_pp_filter.f90
ropp_pp_fourier_filter.f90
ropp_pp_interpol.f90

ropp_pp_interpol_log.f90
ropp_pp_interpolate_trajectory.f90
ropp_pp_monotonous.f90
ropp_pp_read_config.f90
ropp_pp_refrac_bg.f90
ropp_pp_satellite_velocities.f90
ropp_pp_sliding_polynomial.f90
ropp_pp_spline.f90
ropp_pp_tdry.f90
ropp_pp_types.f90
ropp_pp_utils.f90

preprocess/

ropp_pp_preproc.f90 (requires ropp_io)
ropp_pp_amplitude_go.f90
ropp_pp_bangle2phase.f90
ropp_pp_correct_L2.f90
ropp_pp_cutoff.f90 (requires ropp_io)
ropp_pp_impact2doppler.f90
ropp_pp_modelphase.f90
ropp_pp_openloop.f90
ropp_pp_preproc.f90
ropp_pp_preprocess.f90 (requires ropp_io)
ropp_pp_preprocess_cosmic.f90
ropp_pp_preprocess_gras.f90
ropp_pp_radioholographic_filter.f90
ropp_pp_radioptic_analysis.f90
ropp_pp_set_coordinates.f90 (requires ropp_io)
ropp_pp_spectra.f90

D ROPP extra data

For reference and for completeness, the listings of the all ROPP modules' extra variables are listed below.

D.1 ropp_io_addvar

The general form of the extra data, appended to the R0_prof structure by ropp_io_addvar, is described in Table D.1.

R0prof (Additional variables requested by call to ropp_io_addvar, throughout ROPP)	
Structure element	Description
...%vlist%VlistD0d%name	Name of 1 st 0D extra variable
...%vlist%VlistD0d%long_name	Long name of 1 st 0D extra variable
...%vlist%VlistD0d%units	Units of 1 st 0D extra variable
...%vlist%VlistD0d%range	Range of 1 st 0D extra variable
...%vlist%VlistD0d%DATA	Value of 1 st 0D extra variable
...%vlist%VlistD0d%next%name (etc)	Name (etc) of 2 nd 0D extra variable
...%vlist%VlistD0d%next%next%name (etc)	Name (etc) of 3 rd 0D extra variable
...%vlist%VlistD1d%name	Name of 1 st 1D extra variable
...%vlist%VlistD1d%long_name	Long name of 1 st 1D extra variable
...%vlist%VlistD1d%units	Units of 1 st 1D extra variable
...%vlist%VlistD1d%range	Range of 1 st 1D extra variable
...%vlist%VlistD1d%DATA	Value of 1 st 1D extra variable
...%vlist%VlistD1d%next%name (etc)	Name (etc) of 2 nd 1D extra variable
...%vlist%VlistD1d%next%next%name (etc)	Name (etc) of 3 rd 1D extra variable
...%vlist%VlistD2d%name	Name of 1 st 2D extra variable
...%vlist%VlistD2d%long_name	Long name of 1 st 2D extra variable
...%vlist%VlistD2d%units	Units of 1 st 2D extra variable
...%vlist%VlistD2d%range	Range of 1 st 2D extra variable
...%vlist%VlistD2d%DATA	Value of 1 st 2D extra variable
...%vlist%VlistD2d%next%name (etc)	Name (etc) of 2 nd 2D extra variable
...%vlist%VlistD2d%next%next%name (etc)	Name (etc) of 3 rd 2D extra variable

Table D.1: Additional elements of R0prof structure, available throughout ROPP

D.2 PPDiag

The extra data which are output to the netCDF file if `config%output_diag` is set to `.TRUE.` in `ropp_pp`, are described in Table D.2.

PPDiag (<code>config%output_diag = TRUE</code> in <code>ropp_pp</code>)	
Structure element	Description
<code>...%CTimpact</code>	CT processing impact parameter (m)
<code>...%CTamplitude</code>	CT processing amplitude
<code>...%CTamplitude_smt</code>	CT processing smoothed amplitude
<code>...%CTimpactL2</code>	CT processing L2 impact parameter (m)
<code>...%CTamplitudeL2</code>	CT processing L2 amplitude
<code>...%CTamplitudeL2_smt</code>	CT processing smoothed L2 amplitude
<code>...%ba_ion</code>	Ionospheric bending angle in L1 (rad)
<code>...%err_neut</code>	Error covariance of neutral bending angle (rad ²)
<code>...%err_ion</code>	Error covariance of ionospheric bending angle (rad ²)
<code>...%wt_data</code>	Weight of data (data:data+clim) in profile
<code>...%sq</code>	SO badness score: $\text{MAX}[\text{err_neut}^{1/2}/\alpha_N] \times 100\%$
<code>...%L2_badness</code>	L2 phase correction badness score

Table D.2: Elements of PPDiag structure, available from `ropp_pp`

D.3 ropp_fm_bg2ro

The extra data which are appended to the ROPprof structure if the `ropp_fm` tool `ropp_fm_bg2ro_1d` is called without the `-f` option, are described in Table D.3.

ROprof (Absence of <code>-f</code> option in call to <code>ropp_fm_bg2ro_1d</code> , in <code>ropp_fm</code>)	
Structure element	Description
<code>...%gradient_refrac</code>	$\partial N_i / \partial x_j$ matrix
<code>...%gradient_bangle</code>	$\partial \alpha_i / \partial x_j$ matrix

Table D.3: Additional elements of ROPprof structure, available from `ropp_fm`. See Table D.1 for the detailed structure.

D.4 VarDiag

The extra data which are output to the netCDF file if `config%extended_1dvar_diag` is set to `.TRUE.` in `ropp_1dvar`, are described in Table D.4.

VarDiag (config%extended_1dvar_diag = TRUE in ropp_1dvar)

Structure element	Description
...%n_data	Number of observation data
...%n_bgqc_reject	Number of data rejected by background QC
...%n_pge_reject	Number of data rejected by PGE QC
...%bg_bangle	Background bending angle
...%bg_refrac	Background refractivity
...%OmB	Observation minus background
...%OmB_sigma	OmB standard deviation
...%pge_gamma	PGE check gamma value
...%pge	Probability of Gross Error along profile
...%pge_weights	PGE weighting values
...%ok	Overall quality flag
...%J	Cost function value at convergence
...%J_scaled	Scaled cost function value ($2J/m$)
...%J_init	Initial cost function value
...%J_bgr	Background cost function profile
...%J_obs	Observation cost function profile
...%B_sigma	Forward modelled bg standard deviation
...%n_iter	Number of iterations to reach convergence
...%n_simul	Number of simulations
...%min_mode	Minimiser exit mode
...%res_bangle	Analysis bending angle
...%res_refrac	Analysis refractivity
...%OmA	Observation minus analysis
...%OmA_sigma	OmA standard deviation

Table D.4: Elements of VarDiag structure, available from ropp_1dvar

E ROPP user documentation

Title	Reference	Description
ROPP User Licence	SAF/ROM/METO/LIC/ROPP/002	Legal conditions on the use of ROPP software
ROPP Overview	SAF/ROM/METO/UG/ROPP/001	Overview of ROPP and package content and functionality
ROPP User Guide. Part I: Input/output module	SAF/ROM/METO/UG/ROPP/002	Description of <code>ropp_io</code> module content and functionality
ROPP User Guide. Part IIa: Forward model module	SAF/ROM/METO/UG/ROPP/003a	Description of <code>ropp_fm</code> module content and functionality
ROPP User Guide. Part IIb: 1D-Var module	SAF/ROM/METO/UG/ROPP/003b	Description of <code>ropp_1dvar</code> module content and functionality
ROPP User Guide. Part III: Pre-processor module	SAF/ROM/METO/UG/ROPP/004	Description of <code>ropp_pp</code> module content and functionality
ROPP User Guide. Part IV: Applications module	SAF/ROM/METO/UG/ROPP/005	Description of <code>ropp_apps</code> module content and functionality
ROPP UTILS Reference Manual	SAF/ROM/METO/RM/ROPP/001	Reference manual for the <code>ropp_utils</code> module
ROPP IO Reference Manual	SAF/ROM/METO/RM/ROPP/002	Reference manual for the <code>ropp_io</code> module
ROPP FM Reference Manual	SAF/ROM/METO/RM/ROPP/003	Reference manual for the <code>ropp_fm</code> module
ROPP 1D-Var Reference Manual	SAF/ROM/METO/RM/ROPP/004	Reference manual for the <code>ropp_1dvar</code> module
ROPP PP Reference Manual	SAF/ROM/METO/RM/ROPP/005	Reference manual for the <code>ropp_pp</code> module
ROPP APPS Reference Manual	SAF/ROM/METO/RM/ROPP/006	Reference manual for the <code>ropp_apps</code> module
WMO FM94 (BUFR) Specification for Radio Occultation Data	SAF/ROM/METO/FMT/BUFR/001	Description of BUFR template for RO data

Table E.1: ROPP user documentation

Title	Reference	Description
Mono-dimensional thinning for GPS Radio Occultations	SAF/GRAS/METO/REP/GSR/001	Technical report on profile thinning algorithm implemented in ROPP
Geodesy calculations in ROPP	SAF/GRAS/METO/REP/GSR/002	Summary of geodetic calculations to relate geometric and geopotential height scales
ROPP minimiser - minROPP	SAF/GRAS/METO/REP/GSR/003	Description of ROPP-specific minimiser, minROPP
Error function calculation in ROPP	SAF/GRAS/METO/REP/GSR/004	Discussion of impact of approximating erf in ROPP
Refractivity calculations in ROPP	SAF/GRAS/METO/REP/GSR/005	Summary of expressions for calculating refractivity profiles
Levenberg-Marquardt minimisation in ROPP	SAF/GRAS/METO/REP/GSR/006	Comparison of Levenberg-Marquardt and minROPP minimisers
Abel integral calculations in ROPP	SAF/GRAS/METO/REP/GSR/007	Comparison of 'Gorbunov' and 'ROM SAF' Abel transform algorithms
ROPP thinner algorithm	SAF/GRAS/METO/REP/GSR/008	Detailed review of the ROPP thinner algorithm
Refractivity coefficients used in the assimilation of GPS radio occultation measurements	SAF/GRAS/METO/REP/GSR/009	Investigation of sensitivity of ECMWF analyses to empirical refractivity coefficients and non-ideal gas effects
Latitudinal Binning and Area-Weighted Averaging of Irregularly Distributed RO Data	SAF/GRAS/METO/REP/GSR/010	Discussion of alternative spatial averaging method for RO climate data
ROPP 1D-Var validation	SAF/GRAS/METO/REP/GSR/011	Illustration of ROPP 1D-Var functionality and output diagnostics
Assimilation of GPSRO Data in the ECMWF ERA-Interim Re-analysis	SAF/GRAS/METO/REP/GSR/012	Assimilation of GPSRO Data in the ECMWF ERA-Interim Re-analysis
ROPP_PP validation	SAF/GRAS/METO/REP/GSR/013	Illustration of ROPP_PP functionality and output diagnostics

Table E.2: GRAS SAF Reports

Title	Reference	Description
A review of the geodesy calculations in ROPP	SAF/ROM/METO/REP/RSR/014	Comparison of various potential geodesy calculations
Improvements to the ROPP refractivity and bending angle operators	SAF/ROM/METO/REP/RSR/015	Improved interpolation in ROPP forward models
Simplifying EGM96 undulation calculations in ROPP	SAF/ROM/METO/REP/RSR/016	Simplifying ROPP undulation calculations
Simulation of L1 and L2 bending angles with a model ionosphere	SAF/ROM/METO/REP/RSR/017	Simulating L1 and L2 bending angles in ROPP
Single Frequency Radio Occultation Retrievals: Impact on Numerical Weather Prediction	SAF/ROM/METO/REP/RSR/018	Potential impact of loss of L2 bending angle on NWP
Implementation of the ROPP two-dimensional bending angle observation operator in an NWP system	SAF/ROM/METO/REP/RSR/019	Implementation of ROPP 2D forward model at ECMWF
Interpolation artefact in ECMWF monthly standard deviation plots	SAF/ROM/METO/REP/RSR/020	Investigation into plot anomaly
5th ROM SAF User Workshop on Applications of GPS radio occultation measurements	SAF/ROM/METO/REP/RSR/021	Report on 5th ROM SAF User Workshop
The use of the GPS radio occultation reflection flag for NWP applications	SAF/ROM/METO/REP/RSR/022	Impact of reflected occultations at ECMWF
Assessment of a potential reflection flag product	SAF/ROM/METO/REP/RSR/023	Assessment of flagged COSMIC occultations

Table E.3: ROM SAF Reports

Title	Reference	Description
Product Requirements Document (PRD)	SAF/ROM/DMI/MGT/PRD/001	Detailed specification of the products of the ROM SAF

Table E.4: Applicable documents

F Acronyms and abbreviations

AC	Analysis Correction (NWP assimilation technique)
API	Application Programming Interface
BG	Background
BUFR	Binary Universal Format for data Representation
CASE	Computer Aided Software Engineering
CF	Climate and Forecasts (CF) Metadata Convention
CGS	Core Ground Segment
CHAMP	Challenging Mini-Satellite Payload
CLIMAP	Climate and Environment Monitoring with GPS-based Atmospheric Profiling (EU)
CODE	Centre for Orbit Determination in Europe
COSMIC	Constellation Observing System for Meteorology, Ionosphere & Climate
DMI	Danish Meteorological Institute
DoD	US Department of Defense
EC	European Community
ECF	Earth-centred, Fixed coordinate system
ECI	Earth-centred, Inertial coordinate system
ECMWF	The European Centre for Medium-Range Weather Forecasts
EGM-96	Earth Gravity Model, 1996. (US DoD)
EOP	Earth Orientation Parameters
EPS	EUMETSAT Polar System
ESA	European Space Agency
ESTEC	European Space Research and Technology Centre (ESA)
EU	European Union
EUMETSAT	European Organisation for the Exploitation of Meteorological Satellites
EUMETCast	EUMETSAT's primary dissemination mechanism for the NRT delivery of satellite data and products
GALILEO	European GNSS constellation project (EU)
GCM	General Circulation Model
GFZ	GFZ Helmholtz Centre (Germany)
GLONASS	Global Navigation Satellite System (Russia)
GNOS	GNSS Occultation Sounder (China)
GNSS	Global Navigation Satellite Systems (generic name for GPS, GLONASS and the future GALILEO)
GPL	General Public Licence (GNU)
GPS	Global Positioning System (US)
GPS/MET	GPS Meteorology experiment, onboard Microlab-1 (US)

GPSOS	Global Positioning System Occultation Sensor (NPOESS)
GRACE-A/B	Gravity Recovery and Climate Experiment (US/Germany)
GRAS	GNSS Receiver for Atmospheric Sounding (onboard Metop)
GUI	Graphical User Interface
GTS	Global Telecommunications System
HIRLAM	High Resolution Limited Area Model
IERS	International Earth Rotation Service
ITRF	International Terrestrial Reference Frame
ITRS	International Terrestrial Reference System
IGS	International GPS Service
JPL	Jet Propulsion Laboratory (NASA)
LAM	Local Area Model (NWP concept)
LEO	Low Earth Orbiting
LGPL	Lesser GPL (<i>q.v.</i>)
LOS	Line Of Sight
METOP	Meteorological Operational polar satellites (EUMETSAT)
MKS	Meter, Kilogram, Second
MPEF	Meteorological Products Extraction Facility (EUMETSAT)
MSL	Mean Sea Level
N/A	Not Applicable or Not Available
NASA	National Aeronautics and Space Administration (US)
NMS	National Meteorological Service
NOAA	National Oceanic and Atmospheric Administration (US)
NPOESS	National Polar-orbiting Operational Environmental Satellite System (US)
NRT	Near Real Time
NWP	Numerical Weather Prediction
OI	Optimal Interpolation (NWP assimilation technique)
Operational ROM SAF	Team responsible for the handling of GRAS data and the delivery of meteorological products during the operational life of the instrument
PAZ	Spanish Earth Observation Satellite, carrying a Radio Occultation Sounder
PFS	Product Format Specifications
PMSL	Pressure at Mean Sea Level
POD	Precise Orbit Determination
Q/C	Quality Control
RO	Radio Occultation
ROC	Radius Of Curvature
ROM SAF	The EUMETSAT Satellite Application Facility responsible for operational processing of radio occultation data from the Metop satellites. Members are DMI (leader), UKMO, ECMWF and IEEC.
ROPP	Radio Occultation Processing Package
ROSA	Radio Occultation Sounder for Atmosphere (on OceanSat-2 and Megha-Tropiques)
RMDCN	Regional Meteorological Data Communication Network

SAC-C	Satelite de Aplicaciones Cientificas – C
SAF	Satellite Application Facility (EUMETSAT)
SAG	Scientific Advisory Group
SI	Système International (The MKS units system)
TAI	Temps Atomique International (International Atomic Time)
TanDEM-X	German Earth Observation Satellite, carrying a Radio Occultation Sounder
TBC	To Be Confirmed
TBD	To Be Determined
TDB	Temps Dynamique Baricéntrique (Barycentric Dynamical Time)
TDI	Temps Dynamique Terrestre (Terrestrial Dynamical Time)
TDS	True-of-date coordinate system
TerraSAR-X	German Earth Observation Satellite, carrying a Radio Occultation Sounder
TP	Tangent Point
UKMO	United Kingdom Meteorological Office
UML	Unified Modelling Language
UT1	Universal Time-1 (proportional to the rotation angle of the Earth)
UTC	Universal Time Coordinated
VAR	Variational analysis; 1D, 2D, 3D or 4D versions (NWP data assimilation technique)
VT	Valid or Verification Time
WEGC	Wegener Center for Climate and Global Change
WGS-84	World Geodetic System, 1984. (US DoD)
WMO	World Meteorological Organization
WWW	World Weather Watch (WMO)

G Definitions

RO data products from the GRAS instrument onboard Metop and RO data from other data providers are grouped in levels and are either NRT or offline products.

Data levels:

- Level 0: Raw sounding, tracking and ancillary data, and other GNSS data before clock correction and reconstruction;
- Level 1a: Reconstructed full resolution excess phases, SNRs, amplitudes, orbit information, I, Q, and NCO values, and navigation bits;
- Level 1b: Bending angles and impact parameters, Earth location, metadata and quality information;
- Level 2: Refractivity profiles (level 2a), and pressure, temperature, and specific humidity profiles (level 2b and 2c), Earth location, metadata, and quality information;
- Level 3: Gridded level 1 and 2 offline profile products in the form of, e.g., monthly and seasonal zonal means, metadata, and quality information.

Product types:

- NRT product: data product delivered less than 3 hours after measurement;
- Offline product: data product delivered less than 30 days after measurement (the timeliness for some offline level 3 products may be up to 6 months).

H Authors

Many people, inside and outside the ROM SAF, have contributed to the development of ROPP. The principal authors are listed in Table H.1. The ROM SAF extends its sincere appreciation for their efforts.

Name	Current institute	Contribution
Christian Marquardt	EUMETSAT	Author of majority of ROPP-1 code in UTILS, IO, FM and 1DVAR modules, and much personal, pre-existing software.
Huw Lewis	Met Office	1st ROPP Development Manager, FM and 1D-VAR extensions. PP module.
Dave Offiler	Met Office	ROPP Project Manager, IO application code and IO extensions, BUFR format/template.
Sean Healy	ECMWF	Original 1D FM code, 2D FM operator code, introduction of compressibility factors, improved FM vertical interpolation scheme, forward modelling of L1 and L2 bending angles.
Michael Gorbunov	Russian Academy of Sciences	Original PP code.
Axel von Engeln	EUMETSAT	Author of original Test Folder system and of EUMETSAT-formatted RO data reader.
Stig Syndergaard	DMI	Original spectral version of MSIS model (expansion in spherical harmonics and Chebychev polynomials), PP module developments.
Ian Culverwell	Met Office	2nd ROPP Development Manager. Documentation, testing, consolidation, IO development, GRIB2 reader, implementation of tropopause height diagnostics, forward modelling of L1 and L2 bending angles.
Carlo Buontempo	Met Office	Savitzky-Golay thinner code.
Michael Rennie	ECMWF	1st ROPP Test Manager. Test folder developments.
Kjartan Kinch	DMI	Elements of ropp_pp.
Hans Gleisner	DMI	Elements of ropp_pp, prototype GRIB2 reader.
Torsten Schmidt	GFZ	Guidance on tropopause height diagnostics.
Chris Burrows	Met Office	2nd ROPP Test Manager. Test folder developments, improved FM vertical interpolation scheme.
Kent Bærkgaard Lauritsen	DMI	Code reviews; liaison with EUMETSAT (licences, beta tester contracts).

Table H.1: Contributors to ROPP

I Copyrights

The majority of ROPP code is

© Copyright 2009-2020, EUMETSAT, All Rights Reserved.

This software was developed within the context of the EUMETSAT Satellite Application Facility on Radio Occultation Meteorology (ROM SAF), under the Cooperation Agreement dated 29 June 2011, between EUMETSAT and the Danish Meteorological Institute (DMI), Denmark, by one or more partners within the ROM SAF. The partners in the ROM SAF are DMI, Met Office, UK, the Institut d'Estudis Espacials de Catalunya (IEEC), Spain and the European Centre for Medium-Range Weather Forecasts (ECMWF), UK

Some parts of the source code within this distribution were developed within the Met Office outside the context of the ROM SAF and represents pre-existing software (PES); this portion is

© Crown copyright 2016, Met Office. All rights reserved.

Use, duplication or disclosure of this code is subject to the restrictions as set forth in the contract. If no contract has been raised with this copy of the code, the use, duplication or disclosure of it is strictly prohibited. Permission to do so must first be obtained in writing from the Head of Satellite Applications at the following address:

Met Office, FitzRoy Road Exeter, Devon, EX1 3PB United Kingdom

This ROPP package also contains open source code libraries available through its author, Christian Marquardt. This is also PES, and is

© Copyright 2007 Christian Marquardt <christian@marquardt.sc>

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software as well as in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.

IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This ROPP package may also contain open source code libraries available through its author, Michael Gorbunov. This is also PES, and is

© Copyright 1998-2010 Michael Gorbunov <michael.gorbunov@zmaw.de>

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software with the rights to use, copy, modify, merge copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software as well as in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. HOWEVER, ALL EFFORTS ARE BEING MADE BY THE AUTHOR IN ORDER TO FIND AND ELIMINATE ALL POSSIBLE ERRORS AND PROBLEMS. IN THIS CASE THE AUTHOR MAY PROVIDE AN UPDATE.

This ROPP package may also contain open source code libraries available through its author, Stig Syndergaard. This is also PES, and is

© Copyright 1998 Stig Syndergaard <:ssy@dmi.dk>

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sublicense the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software as well as in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.

IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.