

**The Radio Occultation Processing Package (ROPP)
Pre-processor Module User Guide**

Version 11.0

31 December 2021

The ROM SAF Consortium

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 2021	
Reviewed by:	O. Lewis	ROM SAF Project Team	31 December 2021	
Approved by:	K. B. Lauritsen	ROM SAF Project Manager	31 December 2021	

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	28 Feb 2017	IC	Ninth full release (v9.0)
Version 9.1	30 Jun 2019	IC	Update to ninth full release (v9.1)
Version 10.0	30 Sep 2020	IC	Tenth full release (v10.0).
Version 11.0	31 Dec 2021	IC	Eleventh full release (v11.0). Changes from ROPP-10.0 in red — see Change Log (SAF/ROM/METO/SRN/ROPP/018) for details.

ROM SAF

The Radio Occultation Meteorology Satellite Application Facility (ROM SAF) is a decentralised processing centre under EUMETSAT which is responsible for operational processing of radio occultation (RO) data from the Metop and Metop-SG satellites and radio occultation data from other missions. The ROM SAF delivers bending angle, refractivity, temperature, pressure, humidity, and other geophysical variables in near real-time for NWP users, as well as reprocessed Climate Data Records (CDRs) and Interim Climate Data Records (ICDRs) for users requiring a higher degree of homogeneity of the RO data sets. The CDRs and ICDRs are further processed into globally gridded monthly-mean data for use in climate monitoring and climate science applications.

The ROM SAF also maintains the Radio Occultation Processing Package (ROPP) which contains software modules that 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 ROM SAF 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	Applicable and reference documents	1
1.2.1	Applicable documents	1
1.2.2	Reference documents	1
1.3	Acronyms and abbreviations	2
1.4	Definitions, levels and types	4
1.5	Structure of this document	5
2	ROPP	7
2.1	ROPP introduction	7
2.2	User documentation	8
	References	9
3	GNSS-RO data processing	10
3.1	Radio occultation geometry	10
3.2	Radio occultation observations	11
3.3	Bending angle	12
3.3.1	Preprocessing amplitude and excess phase data	12
3.3.2	Geometric optics	14
3.3.3	Wave optics — CT2 algorithm	15
3.4	Ionospheric correction	17
3.5	Abel inversion	18
3.6	Refractivity	19
3.7	Dry temperature	20
3.8	Summary	20
	References	20
4	ROPP pre-processor: amplitude and phase to bending angle	22
4.1	ROPP occultation tool	24
4.1.1	Implementation	24
4.1.2	Code organisation	26
4.2	Configuration options	27
4.3	Preprocessing ropp_pp_preprocess	29
4.3.1	GRAS data preprocessing	29
4.3.2	COSMIC data preprocessing	31

4.3.3	GNOS data preprocessing	32
4.3.4	CHAMP, GRACE etc data preprocessing	32
4.3.5	Model excess phase	33
4.3.6	Transforming open loop data to excess phases	33
4.3.7	Data cutoff	35
4.3.8	Compute spectra	35
4.3.9	Correction of L2 data	35
4.4	Geometric optics ropp_pp_bending_angle_go	35
4.5	Wave optics (CT2) ropp_pp_bending_angle_wo, ropp_pp_DCT	36
4.6	Spatial spectra ropp_pp_spectra_tool, ropp_pp_spectra	38
4.7	Reading GRAS raw sampling data ropp_pp_grasrs2ropp	38
4.8	Processing the results of the ROPP wave optics propagator	38
	References	39
5	ROPP pre-processor: L1 and L2 bending angle to refractivity and dry temperature	42
5.1	ROPP inversion tool	43
5.1.1	Implementation	44
5.1.2	Code organisation	44
5.2	Configuration options	46
5.3	Observation data	46
5.3.1	Data order	47
5.3.2	Standard impact parameter levels ropp_pp_merge_profile	47
5.4	Linear combination ropp_pp_linear_combination	47
5.5	Ionospheric correction	48
5.5.1	Climatology bending angle data	48
5.5.2	Statistical optimization ropp_pp_ionospheric_correction	51
5.6	Inversion	53
5.6.1	Climatology bending angle data	54
5.6.2	Inverse Abel algorithm	54
5.6.3	Hydrostatic integration	56
	References	56
6	Wave optics propagator	60
6.1	Introduction	60
6.2	The 1D ROPP wave optics tool: ropp_pp_wopt_tool	60
6.3	Wave Optics Simulation	63
6.3.1	Initialization	63
6.3.2	Multiple Phase Screen Propagation	63
6.3.3	Propagation to the LEO	66
6.3.4	Optional Full Spectrum Inversion	69
6.4	Example results	70
6.5	The 2D ROPP wave optics tool: ropp_pp_wopt_2D_tool	71

6.5.1	Key modifications when compared to the 1D ROPP wave optics tool	72
6.5.2	An example comparing 1D and 2D retrieved bending angles	75
	References	77
A	Installing and using ROPP	78
A.1	Software requirements	78
A.2	Software release notes	78
A.3	Third-party packages	78
A.3.1	NetCDF (optional in principle)	79
A.3.2	BUFR (optional)	80
A.3.3	GRIB (optional) - either GRIB_API or ecCodes	81
A.3.4	SOFA (optional)	81
A.3.5	RoboDoc (optional)	82
A.3.6	autoconf and automake (optional)	82
A.4	BUILDPACK script	82
A.5	Building and installing ROPP manually	83
A.5.1	Unpacking	84
A.5.2	Configuring	84
A.5.3	Compiling	85
A.5.4	Installing	85
A.5.5	Cleaning up	86
A.6	Linking	86
A.7	Testing	87
A.7.1	ropp_utils	87
A.7.2	ropp_io	87
A.7.3	ropp_pp	88
A.7.4	ropp_apps	89
A.7.5	ropp_fm	89
A.7.6	ropp_1dvar	89
A.8	Troubleshooting	90
B	ropp_pp program files	91
C	ROPP extra diagnostic data	95
C.1	ropp_io_addvar	95
C.2	PPDiag	96
C.3	ropp_fm_bg2ro	96
C.4	VarDiag	96
D	ROPP user documentation	98
E	Authors	103
F	Copyrights	105

1 Introduction

1.1 Purpose of this document

This document, the ROPP_PP User Guide ([RD.2c]) describes the pre-processor module of the Radio Occultation Processing Package (ROPP). The pre-processor is designed to compute ionospheric corrected bending angle, refractivity and dry temperature profiles either from excess phase or L1 and L2 channel bending angle data measured during a radio occultation. It also includes 1- and 2-dimensional wave optics propagation tools, which simulate the radio signal received at the LEO satellite after the GNSS radio signal has been diffracted by an idealised atmospheric refractivity field.

1.2 Applicable and reference documents

1.2.1 Applicable documents

The following documents have a direct bearing on the contents of this document.

[AD.1] Proposal for the Third Continuous Development and Operations Phase (ROM SAF CDOP-3) March 2017 – February 2022, as endorsed by Council 7th December 2016

[AD.2] Product Requirements Document (PRD). SAF/GRAS/METO/MGT/PRD/001

[AD.3] ROPP User Licence. SAF/ROM/METO/LIC/ROPP/002

1.2.2 Reference documents

The following documents provide supplementary or background information and could be helpful in conjunction with this document.

[RD.1] ROPP Architectural Design Document (ADD). SAF/ROM/METO/ADD/ROPP/001

[RD.2] The ROPP User Guides:

[RD.2a] Overview. SAF/ROM/METO/UG/ROPP/001

[RD.2b] ROPP_IO. SAF/ROM/METO/UG/ROPP/002

[RD.2c] ROPP_PP. SAF/ROM/METO/UG/ROPP/004

[RD.2d] ROPP_APPS. SAF/ROM/METO/UG/ROPP/005

[RD.2e] ROPP_FM. SAF/ROM/METO/UG/ROPP/006

[RD.2f] ROPP_1DVAR. SAF/ROM/METO/UG/ROPP/007

[RD.2g] ROPP_UTILS. SAF/ROM/METO/UG/ROPP/008

1.3 Acronyms and abbreviations

AC	Analysis Correction (NWP assimilation technique)
API	Application Programming Interface
Beidou	Chinese GNSS navigation system. Beidou-2 also known as COMPASS
BG	Background
BUFR	Binary Universal Format for data Representation
CASE	Computer Aided Software Engineering
CDR	Climate Data Record
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)
CMA	Chinese Meteorological Agency
C/NOFS	Communications/Navigation Outage Forecasting System (US)
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
FY-3C/D	GNSS radio occultation receivers (CMA)
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, GALILEO and Beidou)
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)
GRACE–FO	GRACE Follow-on 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
ICDR	Intermediate Climate Data Record
IERS	International Earth Rotation Service
ITRF	International Terrestrial Reference Frame
ITRS	International Terrestrial Reference System
IGS	International GPS Service
ISRO	Indian Space Research Organisation
JPL	Jet Propulsion Laboratory (NASA)
KMA	Korean Meteorological Agency
KOMPSAT–5	GNSS radio occultation receiver (KMA)
LAM	Local Area Model (NWP concept)
LEO	Low Earth Orbit
LGPL	Lesser GPL (<i>q.v.</i>)
LOS	Line Of Sight
Megha-Tropiques	Tropical water cycle (and RO) experiment (India/France)
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. Leading entity is DMI; collaborating entities are 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 Baricentrique (Barycentric Dynamical Time)
TDT	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)

1.4 Definitions, levels and types

RO data products from the Metop, Metop-SG and Sentinel-6 satellites and RO data from other missions are grouped in *data levels* (Level 0, 1, 2, or 3) and *product types* (NRT, Offline, NTC, CDR, or ICDR). The data levels and product types are defined below¹. The lists of variables should not be considered as the complete contents of a given data level, and not all data may be contained in a given data level.

Data levels:

¹Note that the level definitions differ partly from the WMO definitions: http://www.wmo.int/pages/prog/sat/dataandproducts_en.php.

- **Level 0:** Raw sounding, tracking and ancillary data, and other GNSS data before clock correction and reconstruction;
- **Level 1A:** Reconstructed full resolution excess phases, total phases, pseudo ranges, SNRs, orbit information, I, Q values, NCO (carrier) phases, navigation bits, and quality information;
- **Level 1B:** Bending angles and impact parameters, tangent point location, and quality information;
- **Level 2:** Refractivity, geopotential height, “dry” temperature profiles (Level 2A), pressure, temperature, specific humidity profiles (Level 2B), surface pressure, tropopause height, planetary boundary layer height (Level 2C), ECMWF model level coefficients (Level 2D), quality information;
- **Level 3:** Gridded or resampled data, that are processed from Level 1 or 2 data, and that are provided as, e.g., daily, monthly, or seasonal means on a spatiotemporal grid, including metadata, uncertainties and quality information.

Product types:

- **NRT product:** Data product delivered less than: (i) 3 hours after measurement (ROM SAF Level 2 for EPS); (ii) 150 min after measurement (ROM SAF Level 2 for EPS-SG Global Mission); (iii) 125 min after measurement (ROM SAF Level 2 for EPS-SG Regional Mission); item
- **Offline and NTC products:** Data product delivered from **about** 5 days to up to 6 months after measurement, depending on the applicable requirements. The evolution of this type of product is driven by new scientific developments and subsequent product upgrades;
- **CDR:** Climate Data Record generated from a dedicated reprocessing activity using a fixed set of processing software². The data record covers an extended time period of several years (with a fixed end point) and constitutes a homogeneous data record appropriate for climate usage;
- **ICDR:** An Interim Climate Data Record (ICDR) regularly extends in time a (Fundamental or Thematic) CDR using a system having optimum consistency with and lower latency than the system used to generate the CDR³.

1.5 Structure of this document

Section 2 briefly describes ROPP and its documentation. Section 3 describes the theoretical processing of GNSS-RO data, from measured excess phase through to bending angles through to refractivity and dry temperature. Section 4 describes the methods and options within the `ropp_pp_occ` tool, which carries out this processing within ROPP. Section 5 does the same for the `ropp_pp_invert` tool, which does the same starting from bending angles. Finally, Section 6 describes the ROPP ‘wave optics propagator’, which calculates the diffraction of GNSS radio waves through an idealised refractivity field to generate a simulated radio signal at the LEO receiver.

Appendices give brief instructions on how to build ROPP, list the files in the `ropp_pp` module, list the ‘extra diagnostic data’ that is produced by the various ROPP tools (usually by means of a ‘-d’ option),

²(i) GCOS 2016 Implementation Plan; (ii) <http://climatemonitoring.info/home/terminology/>.

³<http://climatemonitoring.info/home/terminology> (the ICDR definition was endorsed at the 9th session of the joint CEOS/CGMS Working Group Climate Meeting on 29 March 2018 (<http://ceos.org/meetings/wgclimate-9>)).

record useful ROPP and other ROM SAF documentation, list the principal authors of ROPP, and state the copyright information that applies to various parts of the code.

2 ROPP

2.1 ROPP introduction

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. It was originally designed to process data from the GRAS instrument on Metop-A and B, but the software should be adaptable enough to handle data from any other GNSS-LEO radio occultation mission.

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 2.1 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.

ROPP contains support for a generic data format for radio occultation data (`ropp_io`), one- and two-dimensional forward models (`ropp_fm`), routines for the implementation of 1D-Var retrievals, including quality control routines (`ropp_1dvar`), pre-processing and wave optics propagator routines (`ropp_pp`), and various standalone applications (`ropp_apps`). Utility routines used by some or all of the ROPP modules are provided in an additional module (`ropp_utils`). This structure (Figure 2.1) 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.

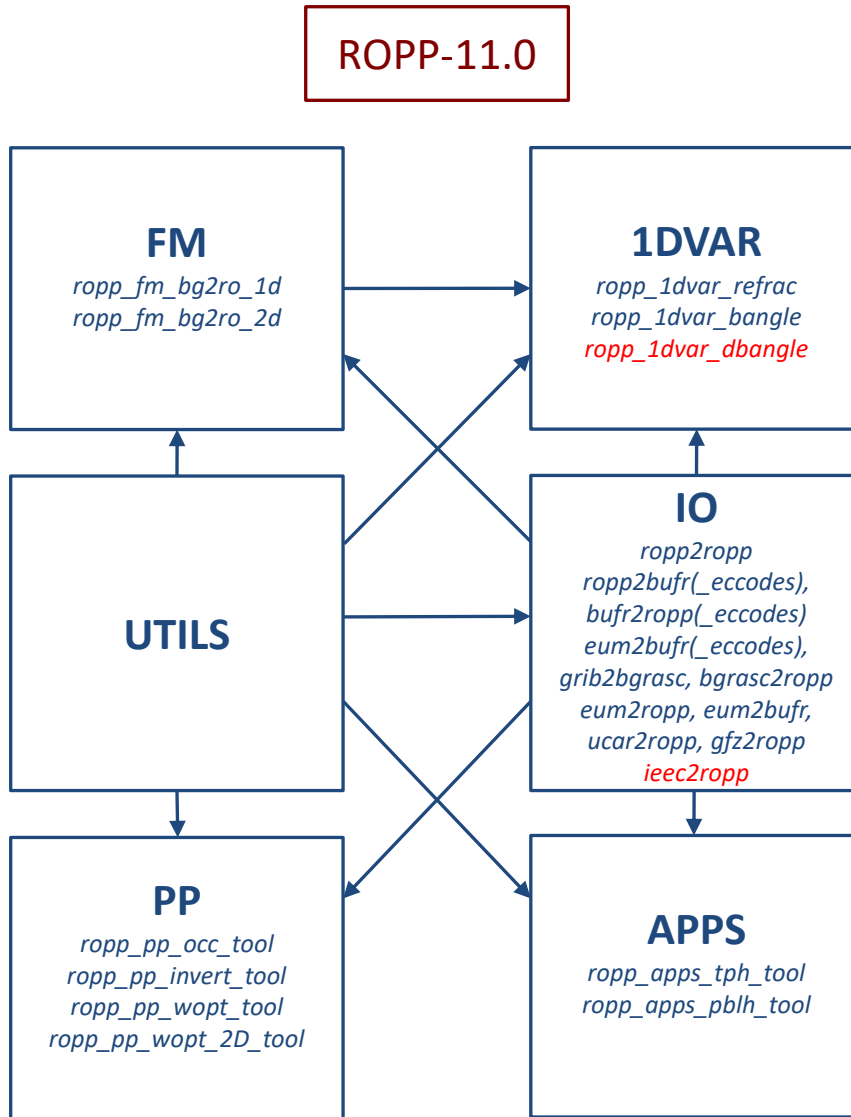


Figure 2.1: The **modules** and **tools** within **ROPP-11.0**. The module at the head of an arrow depends directly on the module at its tail.

2.2 User documentation

A full list of user documentation is provided in Tables D.1, D.2 and D.4. 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.

Module-specific user guides for the utilities (ROM SAF, 2021f), input/output (ROM SAF, 2021d), pre-processor (ROM SAF, 2021e), forward model (ROM SAF, 2021c), 1D–Var (ROM SAF, 2021a) and applications (ROM SAF, 2021b) 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 radio

occultation data from excess phase to bending angle or refractivity, to forward model background fields to refractivity and bending angle profiles, to simulate the propagation of GNSS radio waves through idealised atmospheric refractivity structures, and to perform 1D–Var retrievals of radio occultation data, as well as advice on how to implement ROPP in their own applications.

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 D.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) 1D–Var module User Guide, SAF/ROM/METO/UG/ROPP/007, Version 11.0, 2021a.

ROM SAF, The Radio Occultation Processing Package (ROPP) Applications module User Guide, SAF/ROM/METO/UG/ROPP/005, Version 11.0, 2021b.

ROM SAF, The Radio Occultation Processing Package (ROPP) Forward model module User Guide, SAF/ROM/METO/UG/ROPP/006, Version 11.0, 2021c.

ROM SAF, The Radio Occultation Processing Package (ROPP) Input/Output module User Guide, SAF/ROM/METO/UG/ROPP/002, Version 11.0, 2021d.

ROM SAF, The Radio Occultation Processing Package (ROPP) Pre-processor module User Guide, SAF/ROM/METO/UG/ROPP/004, Version 11.0, 2021e.

ROM SAF, The Radio Occultation Processing Package (ROPP) Utilities module User Guide, SAF/ROM/METO/UG/ROPP/008, Version 11.0, 2021f.

3 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.

3.1 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 3.1. 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)

$$p = n(r)r \sin \phi = \text{constant} \quad (3.1)$$

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 3.1). 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

$$p = n(r_t)r_t \quad (3.2)$$

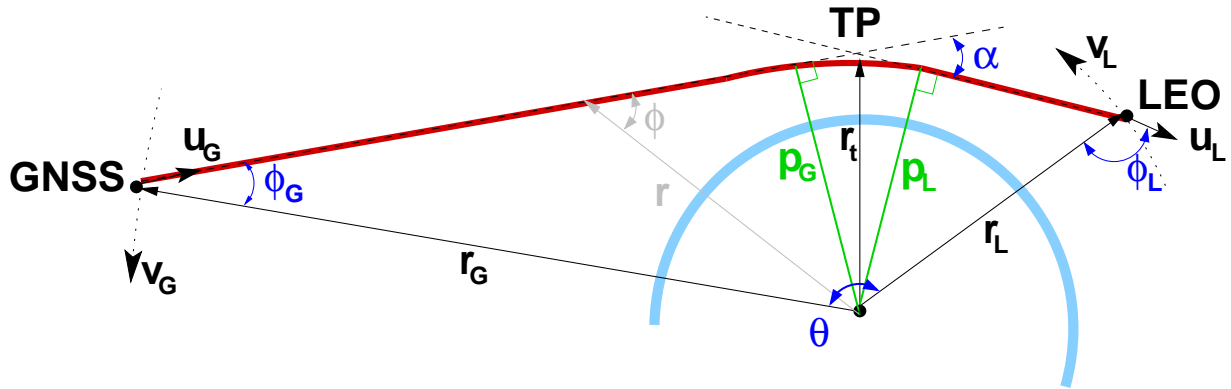


Figure 3.1: Radio occultation geometry. Shown are the bending angle α , the GNSS and LEO side impact parameters (p_G and p_L), the GNSS and LEO coordinate vectors (\mathbf{r}_G , \mathbf{r}_L), the ray path (solid red line) and the satellite side asymptotes of the ray path (dashed). Radius r_t shows 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

$$p = (h + R_c) \cdot n(h + R_c) \quad (3.3)$$

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 (3.2) at some hypothetical tangent point is not necessarily related to the satellite side impact parameters in the non-symmetrical case.

3.2 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) $\Psi(t)$

$$u(t) = A(t) \exp(ik\Psi(t)) \quad (3.4)$$

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 3.4).

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.

3.3 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.

$$u(t) = \sum_j u_j(t) = \sum_j A_j(t) \exp(ik\Psi_j(t)) \quad (3.5)$$

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.

3.3.1 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.

- A significant proportion ($\sim 15\%$) of the occultations detected by the GNOS instrument on the FY-3C satellite suffer from poor tracking of the L2 signal below about 20 km. The resulting extrapolation leads to severe errors in the derived bending angles. A revised extrapolation, directly on the bending angles, is therefore applied to GNOS data. Very rarely this instrument also reports spurious jumps in the LEO position during an occultation; such occultations undergo no further processing (Liao, 2017).
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 radiographic 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 (radio-optic analysis).
 - Compute the mean value of the impact parameter

$$\bar{p}_{1,2}(t) = \frac{\int |v_{1,2}(t, \eta)|^2 p(t, \eta) d\eta}{\int |v_{1,2}(t, \eta)|^2 d\eta} \quad (3.6)$$

- Compute the spectral width

$$\delta p_{1,2}(t) = \frac{\int |v_{1,2}(t, \eta)|^2 (p(t, \eta) - \bar{p}_{1,2}(t))^2 d\eta}{\int |v_{1,2}(t, \eta)|^2 d\eta} \quad (3.7)$$

- Compute a penalty function for L2 data as

$$W(t) = 1 - \exp \left[- \left(\frac{|\bar{p}_1(t) - \bar{p}_2(t)|}{\Delta p_A} + \frac{\delta p_2(t)}{\Delta p_D} \right)^2 \right] \quad (3.8)$$

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

$$\Phi_2^{\text{cor}}(t_i) = \sum_{j=1}^{i-1} D_j \psi_2^{\text{cor}} \quad (3.9)$$

$$D_i \Phi_2^{\text{cor}} = D_i \Phi_2 (1 - W(t_i)) + (D_i \Phi_1 + D_i \Delta\bar{\Phi}) W(t_i) \quad (3.10)$$

$$D_i F = F(t_{i+1} - t_i) \quad (3.11)$$

- Correct L2 amplitude data as

$$A_2^{\text{cor}}(t) = A_2(t)(1 - W(t)) \quad (3.12)$$

3.3.2 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.

$$\Delta s(t) = \frac{c}{f_G} \int_{t_0}^t \left(f_L^{(0)}(t') - f_L(t') \right) dt' \quad (3.13)$$

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

$$f_L^{(0)} = f_G \frac{c - \mathbf{v}_L \cdot \mathbf{u}^{(0)}}{c - \mathbf{v}_G \cdot \mathbf{u}^{(0)}} \sqrt{\frac{c^2 - v_G^2}{c^2 - v_L^2}} \quad (3.14)$$

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

$$f_L = f_G \frac{c - \mathbf{v}_L \cdot \mathbf{u}_L}{c - \mathbf{v}_G \cdot \mathbf{u}_G} \sqrt{\frac{c^2 - v_G^2}{c^2 - v_L^2}} \quad (3.15)$$

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 3.1):

$$\cos \alpha = \mathbf{u}_L \cdot \mathbf{u}_G \quad (3.16)$$

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

$$d = \frac{f_L - f_G}{f_G} = \frac{f_L}{f_G} - 1 = \frac{c - \mathbf{v}_L \cdot \mathbf{u}_L}{c - \mathbf{v}_G \cdot \mathbf{u}_G} - 1 \quad (3.17)$$

The relative Doppler shift is also related to Δs as

$$d = \frac{f_L - f_G}{f_G} = \frac{f_L}{f_G} - 1 = \frac{f_L^{(0)}}{f_G} - \frac{1}{c} \frac{d\Delta s}{dt} - 1 \quad (3.18)$$

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:

$$\frac{c - \mathbf{v}_L \cdot \mathbf{u}_L}{c - \mathbf{v}_G \cdot \mathbf{u}_G} - 1 = d \quad (3.19)$$

$$\mathbf{r}_L \times \mathbf{u}_L - \mathbf{r}_G \times \mathbf{u}_G = \mathbf{0} \quad (3.20)$$

$$\mathbf{u}_L \cdot \mathbf{u}_L = 1 \quad (3.21)$$

$$\mathbf{u}_G \cdot \mathbf{u}_G = 1 \quad (3.22)$$

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 Bouguer'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 $n_L = 1$).

3.3.3 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.

$$\hat{\Phi}_2 u(z) = \sqrt{\frac{-ik}{2\pi}} \int a_2(z,t) \exp(ikS_2(z,t)) u(t) dt \quad (3.23)$$

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

$$dS_2 = \xi dz - \eta dt \quad (3.24)$$

The amplitude function is derived from energy conservation as

$$a_2(z,t) = \sqrt{\left| \frac{\partial^2 S_2(z,t)}{\partial z \partial t} \right|} \quad (3.25)$$

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 .

$$\eta(p,t) = \mathbf{v}_L \cdot \mathbf{u}_L - \mathbf{v}_G \cdot \mathbf{u}_G = p\dot{\theta} + \frac{v_G}{r_G} \sqrt{r_G^2 - p^2} + \frac{v_L}{r_L} \sqrt{r_L^2 - p^2} \quad (3.26)$$

$$S_2(p,t) = -p\theta - \sqrt{r_G^2 - p^2} + a \cos^{-1} \left(\frac{p}{r_G} \right) - \sqrt{r_L^2 - p^2} + p \cos^{-1} \left(\frac{p}{r_L} \right) \quad (3.27)$$

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

$$a_2(p,t) = \sqrt{\left| \mu(p,t) \frac{\partial^2 S_2(p,t)}{\partial p \partial t} \right|} \quad (3.28)$$

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

$$\mu = \sqrt{r_L^2 - p^2} \sqrt{r_G^2 - p^2} \frac{r_L r_G}{p} \sin \theta \left(\dot{\theta} - \frac{v_G}{r_G} \frac{p}{\sqrt{r_G^2 - p^2}} - \frac{v_L}{r_L} \frac{p}{\sqrt{r_L^2 - p^2}} \right) \quad (3.29)$$

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 .

$$\tilde{p}(t, \sigma) = p_0(t) + \frac{\partial p_0}{\partial \sigma}(\sigma - \sigma_0(t)) = f(t) + \frac{\partial p_0}{\partial \sigma} \sigma \quad (3.30)$$

$$f(t) = p_0(t) - \left(\dot{\theta} - \frac{v_G}{r_G} \frac{p_0}{\sqrt{r_G^2 - p_0^2}} - \frac{v_L}{r_L} \frac{p_0}{\sqrt{r_L^2 - p_0^2}} \right)^{-1} \sigma_0(t) \quad (3.31)$$

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

$$dY = \left(\frac{\partial p_0}{\partial \sigma} \right)^{-1} dt \quad (3.32)$$

$$\eta = \frac{\partial p_0}{\partial \sigma} \sigma \quad (3.33)$$

$$\tilde{p} = f(Y) + \eta \quad (3.34)$$

$$\xi = -Y \quad (3.35)$$

Then,

$$S_2(\tilde{p}, Y) = \xi d\tilde{p} - \eta dY = -\tilde{p}Y + \int_0^Y f(Y') dY' \quad (3.36)$$

and

$$a_2(\tilde{p}, Y) = \left(\sqrt{r_L^2 - \tilde{p}^2} \sqrt{r_G^2 - \tilde{p}^2} \frac{r_L r_G \sin \theta}{\tilde{p}} \right)^{1/2} \quad (3.37)$$

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

$$\hat{\Psi}_2 u(\tilde{p}) = \sqrt{\frac{-ik}{2\pi}} a_2(\tilde{p}, Y_s(\tilde{p})) \int \exp(-ik\tilde{p}Y) \exp(ik\tilde{Y}_0 f(Y') dY') u(Y) dY \quad (3.38)$$

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).

3.4 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 possible because, in the L-band, ionospheric refractivity is proportional to the inverse square of the frequency whereas the neutral atmospheric refractivity is largely independent of frequency. The simplest way to eliminate ionospheric effects is by linear combination of bending angles at a common impact parameter (Vorob'ev and Krasil'nikova, 1994).

$$\alpha(a) = \frac{f_1^2 \alpha_1(a) - f_2^2 \alpha_2(a)}{f_1^2 - f_2^2} \quad (3.39)$$

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 (by default) 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 ,

$$\begin{pmatrix} \alpha_N(a) \\ \alpha_I(a) \end{pmatrix} = \begin{pmatrix} \alpha_{CLIM}(a) \\ \bar{\alpha}_I(a) \end{pmatrix} + \bar{K}^{-1} \begin{pmatrix} \Delta\alpha_1(a) - \bar{\alpha}_I(a) \\ \Delta\alpha_2(a) - (f_1/f_2)^2 \bar{\alpha}_I(a) \end{pmatrix} \quad (3.40)$$

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

$$\bar{K}^{-1} = (K^T C_N^{-1} K + C_S^{-1})^{-1} K^T C_N^{-1} \quad (3.41)$$

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

$$\bar{\alpha}_I(a) = \frac{f_1^2}{f_2^2 - f_1^2} (\Delta\bar{\alpha}_1(a) - \Delta\bar{\alpha}_2(a)) \quad (3.42)$$

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

$$\Delta\alpha_i(a) = \alpha_i(a) - \alpha_{CLIM}(a). \quad (3.43)$$

Neither the linear combination nor the statistical optimization methods of ionospheric correction extrapolate missing data. (Wave-optics-processed GNOS L2 data *are* extrapolated to low levels, as described in Sec 4.5.)

3.5 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).

$$\begin{aligned} \alpha(a) &= -2 \int_{r_t}^{\infty} d\alpha = -2a \int_{r_t}^{\infty} \frac{1}{\sqrt{r^2 n^2 - a^2}} \frac{d \ln(n)}{dr} dr \\ &= -2a \int_a^{\infty} \frac{1}{\sqrt{x^2 - a^2}} \frac{d \ln(n)}{dx} dx . \end{aligned} \quad (3.44)$$

The second expression is obtained by substituting $x = nr$ (and $a = n(r_t)r_t$). The negative sign in Equation (3.44) follows the convention that bending towards the Earth's surface is positive.

Remarkably, Equation (3.44) can be inverted, so that the refractive index profile $n(r)$ for a given corrected bending angle profile $\alpha(a)$ is given by

$$n(r) = \exp \left[\frac{1}{\pi} \int_x^\infty \frac{\alpha(a)}{\sqrt{a^2 - x^2}} da \right], \quad (3.45)$$

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 3.4.

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

3.6 Refractivity

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

$$N = (n - 1) \times 10^6. \quad (3.46)$$

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.

$$N = \kappa_1 \frac{P}{T} + \kappa_3 \frac{e}{T^2} + \kappa_4 \frac{n_e}{f^2} + \kappa_5 W. \quad (3.47)$$

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 the ROPP_1DVAR User Guide (ROM SAF, 2021).

3.7 Dry temperature

The so-called dry temperature is obtained by ignoring the water vapour in Equation (3.47) such that

$$N = \kappa_1 \frac{P}{T} . \quad (3.48)$$

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 (3.47) can be neglected, but it is significantly smaller than real temperature in moist regions of the troposphere.

Section 5.6.3 provides further details.

3.8 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:10.1029/2002RS002763, 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.
- Liao, M., Visiting Scientist Report 32: Preliminary investigation with GNOS bending angle data: Monitoring, quality control and possible implications for ROPP, SAF/ROM/DMI/REP/VS32/001, Version 1.0, 2017.
- 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) 1D–Var module User Guide, SAF/ROM/METO/UG/ROPP/007, Version 11.0, 2021.
- 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.

4 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 5. 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 3 and Gorbunov and Lauritsen (2004), Gorbunov et al. (2005).

Figure 4.1 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.

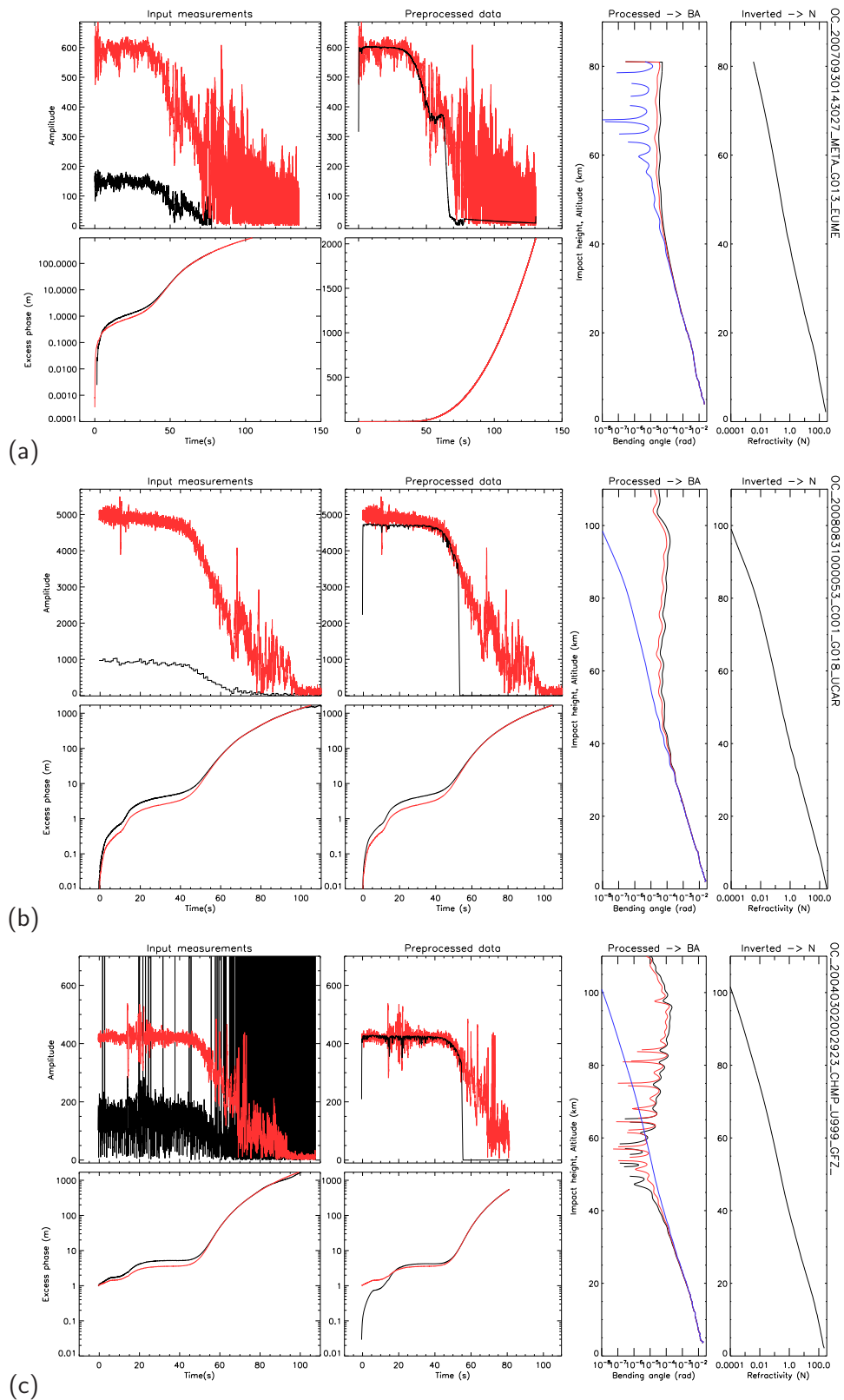


Figure 4.1: (a) A GRAS profile, (b) a COSMIC profile and (c) a CHAMP profile. Example L1 (red) and L2 (black) amplitude and phase measured data, pre-processed data and the resulting bending angle profiles computed using `ropp_pp` routines. The ionospheric corrected bending angle as a function of impact parameter is plotted in blue, from which the refractivity profile as a function of altitude is computed.

4.1 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 4.2 shows the general outline of `ropp_pp_occ_tool` processing.

4.1.1 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, 2021) 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: a local file search for MSIS_coeff.nc)
<code>-bfile <bfile></code>	background atmospheric profile file (if using 'BG' method)
<code>-navfile <nfile></code>	auxiliary file holding external navigation bits (COSMIC only) (Default: use derived (COSMIC) or ingested (GRAS) navbits)
<code>-occ <occ_method></code>	processing method, W0 or G0 (default: W0)
<code>-filter <f_method></code>	filtering method, slpoly or optest (default: slpoly)
<code>-fit</code>	use two parameter fitting method (default: .TRUE., but can be switched off in configuration file)
<code>-ellipsoid</code>	output height with respect to WGS84 ellipsoid (default: output height wrt EGM96 geoid)
<code>-full</code>	output whole profile (default: only output the data within the observed range)
<code>-w</code>	process idealised output from <code>ropp_pp_wopt_tool</code> or <code>ropp_pp_wopt_2D_tool</code>
<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. By default (or using `-occ W0`), `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

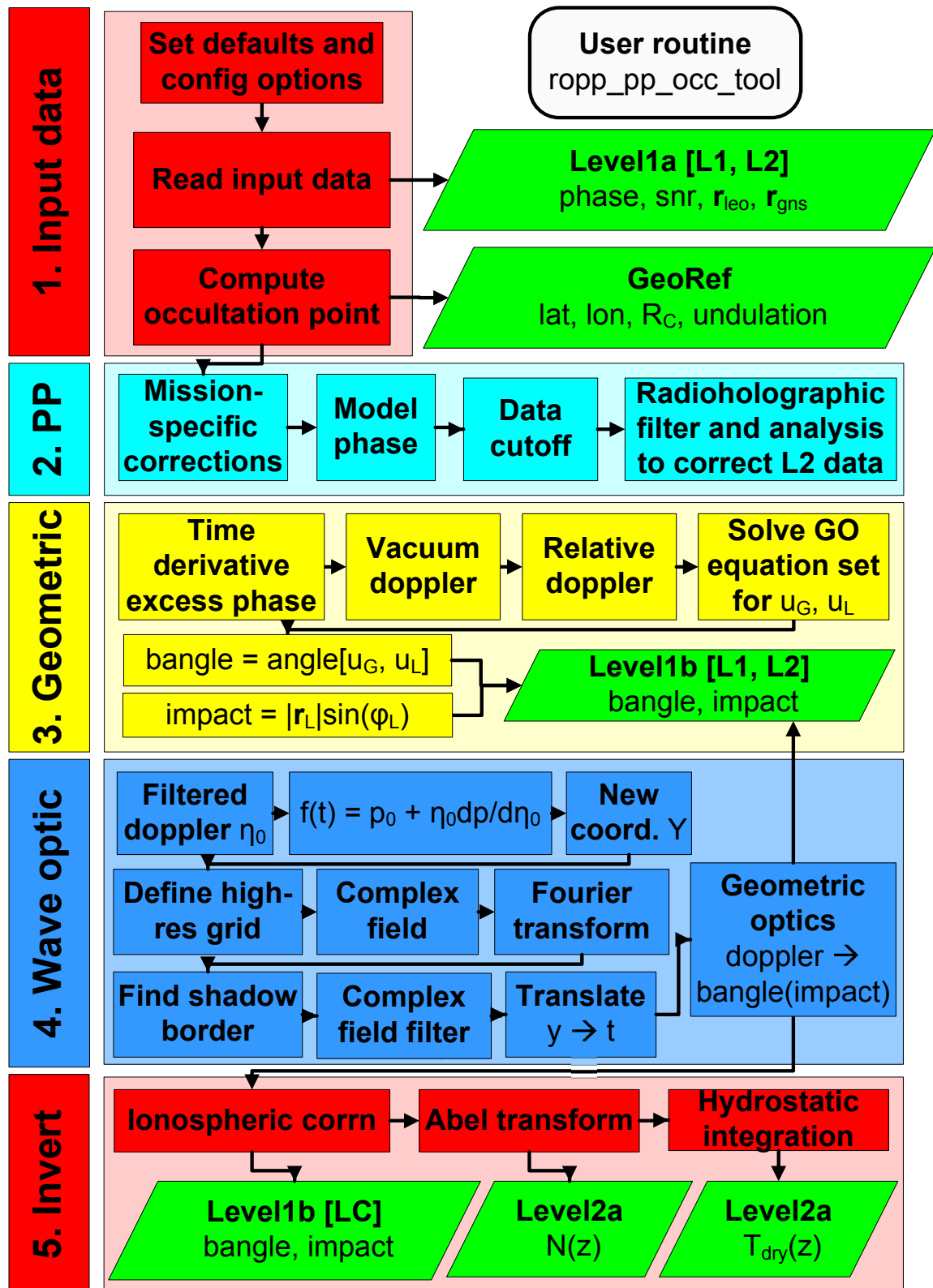


Figure 4.2: Flow chart illustrating calling tree of the ROPP pre-processor occ tool to compute ionospheric corrected bending angle and refractivity profiles from input L1 and L2 channel amplitude and phase measurements. See also Figure 5.2.

the `-m` flag. By default, or by using `-m GMSIS` or `-m GBAR0`, `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 BAR0` 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 (3.39)) and no climatological or background model data are used for the Abel inversion.

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. By default (or using `-filter slpoly`) a sliding polynomial filter is used. Specifying `-filter optest` causes an optimal estimation filter to be used.

4.1.2 Code organisation

Figure 4.2 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_IO User Guide (ROM SAF, 2021).

- **Compute occultation point and undulation**

The `ropp_utils` routine `occ_point` calculates the occultation point, which, in ROPP, is defined to be the location of the lowest occultation perigee point. Thus, for every available pair of satellite positions, the smallest distance from the centre of the Earth (strictly, from the centre of curvature of the Earth) to the straight line between the satellites is calculated. This is the perigee, and the one which is nearest the centre (of curvature) of the Earth defines the occultation point. Cartesian coordinates of the centre of curvature and the radius of curvature at the occultation point are also computed by this routine.

The undulation — the difference between the WGS84 ellipsoid (NGA/WGS84) 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 Geospatial-Intelligence Agency website (NGA/EGM96). 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 4.1), 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 3.3.1 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 (3.8)) 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 3.3.2 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 3.3.3 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 (3.39). 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 5 for further details.

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

Routine `ropp_pp_invert` performs the Abel inversion (Equation (3.45)) to derive a refractivity profile from the corrected bending angles. See Section 5 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 5.6.3 for further details.

4.2 Configuration options

A number of configuration options can be defined by the user in order to tune the preprocessing and wave optics algorithms. Table 4.1 lists these configuration options and their default values held in a structure of derived type `PPConfig` (see also Table 5.1). 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`. Example configuration files, tailored to particular satellite missions, are provided in the `config` directory of the `ropp_pp` distribution. The `wopt_pp.cf` configuration file is suitable for processing the (synthetic) output of `ropp_pp_wopt_tool` or `ropp_pp_wopt_2D_tool`, as described in Sec 6.

PPConfig		
Structure element	Default	Description
...%occ_method	W0	Phase to bending angle method (G0 = geometric, W0 = 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	Refractivity 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 radiographic filtering)
...%dsh	200 m	Shadow border width
...%opt_DL2	.TRUE.	Flag to indicate degraded L2 data
...%opt_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
...%tp_bending	.FALSE.	Account for bending in calculation of TP lat-lons

Table 4.1: Elements of the PPConfig 'configuration' structure that are used by `ropp_pp` preprocessing and wave optics routines. The default values are used unless overwritten by configuration options that are read from an input file. In the code, the above default values are held in `ropp_pp/common/ropp_pp_types.f90`.

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 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 4.1. Note that this routine requires that the `ropp_io` module is installed (ROM SAF, 2021).

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

The L1 raw sampling or open loop phase measurements include *navigation bits*, which are pseudo-random phase changes of 0 or π that are applied to the excess phase. These are removed by `ropp_pp_openloop`.

Different preprocessing applies to different instruments. A key aspect is the calculation of the *lost carrier flag*, LCF, which is an integer variable that holds information about the navigation bits of the occultation (among other things) in a bit-wise fashion, as follows:

- bit 0: 0 \Rightarrow closed loop mode, 1 \Rightarrow open loop/raw sampling mode
- bit 1: external navigation bits
- bit 2: external navigation bits quality
- bit 3: missing data indicated
- bit 4: data provider's internal navigation bits
- bit 5: data provider's internal navigation bits quality
- bit 6: closed loop and raw sampling overlap
- bit 7: ROPP-generated internal navigation bits
- bit 8: ROPP-generated internal navigation bits quality

4.3.1 GRAS data preprocessing

GRAS data from the Metop satellites are a combination of open loop/raw sampling measurements for a ray height below 5-10 km and closed loop tracking measurements elsewhere. GRAS data from EUMETSAT are converted into standard ROPP format by means of the `ropp_io` data readers `eum2ropp` (if they are in the now-standard data format detailed by (Marquardt, 2016)), or `ropp_pp_grasrs2ropp` (if they are in the deprecated data format described in connection with the data reader in Sec 4.7). Both input data formats contain 'internal' navigation bits, which are always read. The input files may also contain 'external' navigation bits. If these are present, they are read; otherwise they are set equal to the internal navbits.

LCF variables are defined for closed loop and open loop/raw sampling parts of the data. In both old and new data formats, bit 3 of the closed loop LCF is set (i.e. the data are considered missing) if the data sample separation is more than 5% greater than the expected data gap of 0.02 sec (i.e. $1/(50 \text{ Hz})$). Bit 6 is set (later) wherever there is significant overlap between the closed loop and open loop/raw sampling data.

Bit 3 of the open loop/raw sampling LCF is set if the data gap is more than 5% greater than 0.02 sec for open loop data or 0.001 sec (i.e. $1/(1000 \text{ Hz})$) for raw sampling data. Bit 0 (open loop) of this LCF is always set, whether it be open loop or raw sampling data. Bit 1 is set when the ingested external navbit equals 1, and bit 4 is set when the ingested internal navbit equals 1. The 'quality' navbits, 2 and 5, are always set. Navbits 7 and 8 are not set here. Thus, the internal navbit information actually appears in the 'alternative' navbit elements of LCF.

Recent (later than Jan 2018) GRAS data files from EUMETSAT contain L1 excess phase I and Q components that have already been corrected for navigation bits. For these files, therefore, the internal and external navbits are nullified, to prevent any miscorrections being applied in later processing.

The closed loop and open loop/raw sampling LCFs are merged into a single LCF for the entire profile, which is held in the first `vlist%VlistD1d` substructure of the `R0prof` data structure, and written out to the ROPP format netCDF file as the variable `open_loop_lcf`.

The resulting ROPP format files can then be used by `ropp_pp_occ_tool`. The first step of the preprocessing is to cut off corrupted data as described in Sec 4.3.7. GRAS-specific preprocessing is carried out by subroutine `ropp_pp_preprocess_GRASRS`, which does the following.

- It reads `ro_data%vlist%VlistD1d` and sets it equal to LCF. It then uses this to determine which parts of the profile are closed loop and which are open loop/raw sampling.
- It generates a merged time grid, anchored at the highest point of the raw sampling section.
- It interpolates closed loop and open loop/raw sampling data to the merged time grid, and then merges them.
- It generates a model/reference excess phase, using `ropp_pp_modelphase` (see Sec 4.3.5). This is subtracted from the L1 excess phase before re-accumulating it, removing phase discontinuities, filling in data gaps (as identified by bit 3 of the merged LCF) by interpolation, and finally adding back in the model phase.
- It (re-)calculates the occultation point longitude and latitude, for later use.

After this, subroutine `ropp_pp_cutoff_amplitude` is called, which removes missing data points and those whose `snr_L1ca` values are too large. See Sec 4.3.7. An excess phase model (or reference) is then calculated by calling `ropp_pp_modelphase`, as described in Sec 4.3.5. This is used in the processing of open loop data to excess phase data, which is carried out by subroutine `ropp_pp_openloop`, as described in Sec 4.3.6. (Missing and open loop/raw sampling L2 phase data are first set equal to the model phase data.) Some data points are removed, based on their amplitude, impact parameters and smoothed geometrical-optics-derived bending angles, by `ropp_pp_cutoff`. (See Sec 4.3.7 again.) Spectra (i.e. signal amplitudes as a function of

Doppler shift and time) are calculated if `config%opt_spectra` is set. See Sec 4.3.8. If `config%opt_DL2` is set, then the L2 channel data are assumed to be degraded. They are therefore corrected by the measures described in Sec 4.3.9. Finally, bit 2 of `ro_data%PCD` is set if the time ordering of the impact parameters indicates that this is a rising occultation (ROM SAF, 2021).

4.3.2 COSMIC data preprocessing

COSMIC data are a combination of open loop measurements for a ray height below 8-11 km and closed loop measurements elsewhere. `atmPhs` files from the CDAAC website (UCAR/CDAAC, 2014), which contain these data, must first be converted into ROPP format files by means of the `ropp_io` tool `ucar2ropp`. See (ROM SAF, 2021) for details. The relevant point here is that the open loop fragment of the input excess phase dataset is identified by the open loop phase model variable `xmdl` in the UCAR `atmPhs` datafile. Open loop data points have `xmdl` \neq -999. If present, `xmdl` is held in the first `vlist%VlistD1d` substructure of the `R0prof` data structure, and written out to the ROPP format netCDF file as the variable `open_loop_lcf`.

`xmdl` is absent from `atmPhs` files containing Metop and Kompsat-5 data. For these files, `xmdl`, the 'differenced' open loop phase model, is used instead.

The resulting ROPP format files can then be used by `ropp_pp_occ_tool`. The first step of the preprocessing is to cut off corrupted data — see Sec 4.3.7. COSMIC-specific preprocessing¹ is carried out by subroutine `ropp_pp_preprocess_COSMIC`, which does the following.

- It removes the worst phase discontinuity.
- It retrieves lost carrier flag information from input data file. Where `ro_data%vlist%VlistD1d` (which is `xmdl` or `xmdl` in the original `atmPhs` file) is not equal to -999.0, bit 0 of a null-initialised LCF variable is set, indicating open loop/raw sampling data.
- It reads external navigation bits from an auxiliary file, if the `-navbit` option to `ropp_pp_occ_tool` is invoked. These files, which have names like `gpsBit_C001.2015.001.00.08.G19_2014.2860.txt`, can be obtained from the CDAAC website (UCAR/CDAAC, 2014). Their structure (exemplified by the above file) is:

```
1104106080.00 1104106230.00 7500
11
01
...
11
11
```

The first and second numbers are the GNSS start time (`tg0`) and end time (`tgN`) of the navigation bits (in seconds); the third (`ngb`) is the number of two-digit navigation bits (`gb`) in the file, which complete the file. (Note that $ngb/(tgN-tg0) = 50$ Hz.) The bits are assumed to be uniformly spaced

¹And that of Metop data provided by CDAAC.

between tg_0 and tg_N , and the nearest one to each occultation measurement time modifies that time's LCF value according to the rule:

- If $gb \geq 10$, set external navigation bit (number 1) of LCF;
 - If $gb = 1 \pmod{10}$, set external navigation bit quality (number 2) of LCF.
- It multiplies `snr_L1ca`, `snr_L1p` and `snr_L2p` by 0.1.

Subsequent preprocessing, from the call to `ropp_pp_cutoff_amplitude` to the setting of the `PCD_rising` bit of `ro_data%PDC`, is the same as that for GRAS data, save that missing and open loop/raw sampling L2 excess phases are not set equal to the model phases before calling `ropp_pp_openloop`.

4.3.3 GNOS data preprocessing

GNOS data from the FY-3C and FY-3D satellites occasionally display unphysical ‘jumps’ in the LEO position between the beginning and end of an occultation. `ropp_pp_preprocess_gnos` calculates the angle θ between the radii to the LEO at these events according to $\cos \theta = \mathbf{r}_{LEO}(1) \cdot \mathbf{r}_{LEO}(n) / |\mathbf{r}_{LEO}(1)| |\mathbf{r}_{LEO}(n)|$. If this angle exceeds 10 degrees the occultation is excluded from further processing. This is enough to catch occasional rogue profiles. This routine also calculates the minimum valid L2 SLTA from the L2 excess phase data, and, for future use, stores it in the new `L2_min_SLTA` element of the `PPDiag` data structure. Finally, some QC diagnostics are set according to the average L1 and L2 excess phases between 60 and 80 km SLTA. Specifically, if these are both less than 150 m for a rising profile, or if the minimum valid L2 SLTA is higher than 50 km, then `ro_data%overall_qual` is set to 20. (The default is 100.)

Otherwise, the preprocessing of GNOS data follows the same pattern as for GRAS and COSMIC data, except that `ropp_pp_openloop` is not called, as it is assumed that navigation bits have already been corrected in level 1a data in these files.

GNOS data processing is new at ROPP-9.1, and details are subject to change in later releases. However, this is the processing used operationally at CMA to generate the bending angles that they distribute to global NWP centres via the GTS.

4.3.4 CHAMP, GRACE etc data preprocessing

CHAMP, GRACE, TerraSAR-X etc data are provided by GFZ in pairs of ascii files, which can be processed by the `ropp_io` tool `gfz2ropp`. Level 1a files provide LCF directly, which is copied directly to the output netCDF file as variable `lcf`. (It is still read in as the first element of `ro_data%vlist%VlistD1d`.)

When passed through `ropp_pp_occ_tool`, the only mission-specific processing of these data is to set `snr_L2p` equal to `snr_L1ca` if any of the former are zero. Otherwise, the preprocessing of CHAMP, GRACE etc data follows the same pattern as for GRAS and COSMIC data, except that `ropp_pp_openloop` is not called, as it is assumed that navigation bits have already been corrected in level 1a data in these files.

4.3.5 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.

$$\alpha_{MSIS}(p) = \theta(t) - \cos^{-1} \frac{p}{r_G(t(p))} - \cos^{-1} \frac{p}{r_L(t(p))} \quad (4.1)$$

The relative Doppler shift, $d = d(p(t))$, is computed using Equation (3.17) 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 $\Delta\Psi_{MSIS}$ is given by

$$\Delta\Psi_{MSIS} = \Psi_{MSIS}(t) - \Psi_0(t) = c \int_0^t d(p(t)) dt - |\mathbf{r}_G - \mathbf{r}_L| \quad (4.2)$$

4.3.6 Transforming open loop data to excess phases

Subroutine `ropp_pp_openloop` removes the navbits necessary to transform open loop data into excess phase data suitable for further processing by `ropp_pp_occ_tool`. This is only done for data from GRAS and COSMIC missions. It follows the methodology of (Sokolovskiy et al., 2006) and (Sokolovskiy et al., 2009) in doing the following.

- The model or reference excess phase $\Delta\Psi_{MSIS}$ is removed from the measured L1 and L2 excess phases $\Delta\Psi_1$ and $\Delta\Psi_2$ (which are actually distances, having been divided by their respective k_s) to form 'model-denuded' L1 and L2 excess phases $\Delta\phi_1$ and $\Delta\phi_2$ (in radians):

$$\Delta\phi_J = k_J(\Delta\Psi_J - \Delta\Psi_{MSIS}), \quad \text{for } J = 1, 2. \quad (4.3)$$

- The quality of the navbits are assessed by computing the following two functions at each data point:

$$F_i^{\Delta\phi} = \left| \frac{(\Delta\phi_1(i) - \Delta\phi_1(i-1) + \pi) \bmod 2\pi - \pi}{\pi} \right| \quad (4.4)$$

$$F_i^B = |B(i) - B(i-1)| \quad (4.5)$$

where $B(i)$ is the external navbit at point i . $F_i^{\Delta\phi}$ is a sawtooth function, peaking at unity where $|\Delta\phi_1(i) - \Delta\phi_1(i-1)| = \pi$. It should therefore be correlated with F_i^B , which has the same property, provided that the navbits have been calculated correctly, a supposition that is checked by calculating

the correlation between the two functions:

$$C^B(\Delta i) = \frac{\langle F_i^{\Delta\phi} F_{i+\Delta i}^B \rangle}{\sqrt{\langle (F_i^{\Delta\phi})^2 \rangle \langle (F_{i+\Delta i}^B)^2 \rangle}}, \quad (4.6)$$

in which the averaging over the index i , indicated by $\langle \rangle$, spans a time interval of 4 seconds below the maximum height of the open loop region. The position of maximum correlation defines the shift that is applied to the navigation bits sequence. This shifting has been found necessary for some COSMIC profiles. If the maximum correlation is less than 0.6, then internal navigation bits are derived instead. These are defined using the scheme of (Sokolovskiy et al., 2009), which is encoded in subroutine `ropp_pp_internal_navbit`, which is itself contained within `ropp_pp_openloop`. This routine provisionally identifies internal navigation bits from changes in the external navbits, if available. (For COSMIC data these will need to have been provided in an auxiliary file.) The first value is initially used throughout. If, however, $\Delta\phi_1$ changes by more than $\pm\pi/2$ across one of the data ‘frame boundaries’, which occur every 0.02 sec (= 1/50 Hz), then it is swapped (0 to 1 or 1 to 0), and the new value is used until another phase shift occurs at a later frame boundary. (If working with 50 Hz data, the next frame boundary will of course occur at the next data point.) Thus we arrive at a piecewise-constant series of internal navbits, either 0 or 1 throughout each 0.02 sec frame, and these define bit 7 (internal navbit) of the LCF. Bit 8 (internal navbit quality) is also set for all open loop points.

- The external navbits (or internal navbits, if the external ones are missing or unusable) are used to correct the open loop L1 excess phase data by adding π wherever they are set:

$$\Delta\phi_1 \mapsto \Delta\phi_1 + \pi \begin{cases} 1, & \text{when LCF(bit 1)} = 1 \text{ (or when LCF(bit 7)} = 1, \text{ if using internal navbits)} \\ 0, & \text{otherwise.} \end{cases} \quad (4.7)$$

- L1 excess phases are re-accumulated according to the rules:

$$\Delta\phi_1(t_i) = \Delta\phi_1(t_{i-1}) + \begin{cases} (\Delta\phi_1(t_i) - \Delta\phi_1(t_{i-1}) + \pi) \bmod 2\pi - \pi, & \text{if open loop and navbits OK for } i \text{ and } i-1 \\ (\Delta\phi_1(t_i) - \Delta\phi_1(t_{i-1}) + \pi/2) \bmod \pi - \pi/2, & \text{if open loop and otherwise, or if closed loop.} \end{cases}$$

- The open and closed loop model-denuded L1 and L2 excess phases are further re-accumulated by means of

$$\Delta\phi_J(t_i) = \Delta\phi_J(t_{i-1}) + (\Delta\phi_J(t_i) - \Delta\phi_J(t_{i-1}) + \pi) \bmod 2\pi - \pi, \quad \text{for } J = 1, 2. \quad (4.8)$$

- Any discontinuity between the closed loop and open loop segments of the data is removed, after which the reference phase model is added back to both phases, by inverting Eqn (4.3), thus:

$$\Delta\Psi_J = \Delta\Psi_{MSIS} + (\Delta\phi_J/k_J), \quad \text{for } J = 1, 2. \quad (4.9)$$

4.3.7 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`.

4.3.8 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 4.6.

4.3.9 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 (3.6)–(3.12)). 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

$$Q^{L2}(t) = \left(\frac{|\bar{p}_1(t) - \bar{p}_2(t)|}{\Delta p_A} + \frac{\delta p_2(t)}{\Delta p_D} \right)^2 \quad (4.10)$$

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 $\bar{\alpha}_1 + \Delta\alpha_f$ where $\Delta\alpha_f$ 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 $\Psi_{1,2}$ (using `ropp_pp_bangle2phase`), which are used to correct L2 excess phase and amplitude data according to Equations (3.8)–(3.12).

4.4 Geometric optics `ropp_pp_bending_angle_go`

Computation of L1 and L2 bending angles and impact parameters using the geometric optics approximation (Section 3.3.2) 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 (3.22) are then solved for each time sample by calling `ropp_pp_geometric_optics`.

4.5 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 3.3.3) 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 (3.38)). The transformation from measured data to the new coordinate system (Equations (3.31)–(3.35)) 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 $|\hat{\Phi}_2 u_{1,2}(p)|$ 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,

$$w(p, \xi) = \int_{p-\Delta p/2}^{p+\Delta p/2} \cos\left(\frac{\pi(p' - p)}{\Delta p}\right) \frac{\hat{\Phi}_2 u_1(p')}{\exp(ik\bar{\Psi}'(p))} \exp(-ik\xi p') dp' \quad (4.11)$$

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

$$\delta\alpha_T(p) = \left(\frac{\int |w(p, \xi)|^2 \xi^2 d\xi}{\int |w(p, \xi)|^2 d\xi} \right)^{1/2} \quad (4.12)$$

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 W0
                                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)
```

Additional GNOS data processing

Excess phases from the GNOS instrument on FY-3C and FY-3D are susceptible to poor L2 signals at low altitudes. These data therefore undergo further processing in subroutine `ropp_pp_bending_angle_gnos`. This handles the $\sim 15\%$ of poor GNOS L2 excess phase signals by fitting the *geometrical optics*-processed L2–L1 bending angles to a simple function of impact parameter a , namely

$$\alpha_{L2}(a) - \alpha_{L1}(a) = Kr_0/(r_0^2 - a^2)^{3/2} \quad (4.13)$$

where $r_0 = R_C + 300$ km. (This functional form results from assuming an infinitely thin, spherically symmetric ‘delta-function’ ionosphere sited at r_0 — see ROM SAF (2015) for details.) K is calculated from a least squares fit to $\alpha_{L2} - \alpha_{L1}$ in the 20 km above the minimum valid L2 SLTA. Eqn (4.13) is then used

to extrapolate the *wave optics*-processed L2 signal below this height. If this fit is particularly noisy, as determined by its mean square deviation from the data, then `ro_data%overall_qual` is again set to 20.

GNOS data processing is new at ROPP-9.1, and details are subject to change in later releases. However, this is the processing used operationally at CMA to generate the bending angles that they distribute to global NWP centres via the GTS.

4.6 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 shows how users could plot the output. Examples for the occultations shown in Figure 4.1 are plotted in Figure 4.3.

4.7 Reading GRAS raw sampling data `ropp_pp_grasrs2ropp`

`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 data, define the lost carrier flag (LCF) bits, and output the data in a standard ROPP format netCDF file (ROM SAF, 2021). See Sec 4.3.1 for details of the generation of LCFs from the internal and external navigation bits. The output file is then suitable for processing using `ropp_pp_occ_tool`, thus:

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

Note that `ropp_pp_grasrs2ropp` is purely a data format conversion code. Apart from the merging of the raw sampling and closed loop data, and some phase accumulation calculations, it undertakes no processing of the data. That work is done by `ropp_pp_occ_tool`.

It should also be noted that the format for the input files that are read by this tool has long been superseded by the newer EUMETSAT file format detailed in (Marquardt, 2016). These files can be converted to ROPP format by means of the `eum2ropp` tool described in the ROPP IO user guide (ROM SAF, 2021). It is therefore likely that `ropp_pp_grasrs2ropp` will be removed from ROPP in the near future.

4.8 Processing the results of the ROPP wave optics propagator

Users wishing to calculate bending angles from the excess phases generated by the ROPP wave optics propagators `ropp_pp_wopt_tool` or `ropp_pp_wopt_2D_tool`, described in Sec 6, need to do two things:

- Apply the '-w' option to `ropp_pp_occ_tool` or `ropp_pp_wopt_2D_tool`. This switches off some mission-specific pre-processing and input data cut-offs, which are not appropriate for such artificial data.
- Use (something like) the configuration file `ropp_pp/config/wopt_pp.cf`. This contains what are currently considered the best processing parameters for WOPT-derived data. Users can of course experiment with any of the elements of the configuration file, but this is a good place to start.

Such CT2-derived bending angles could be profitably compared to the (optional) FSI-derived bending angles derived directly by `ropp_pp_wopt_tool` or `ropp_pp_wopt_2D_tool`, as described in Sec 6.3.4 of this User Guide.

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.
- Marquardt, C., Radio Occultation Level 1 Product Format Specification, EUM/TSS/SPE/16/817861, Issue v1A, 2016.
- NASA/NIMA, NASA and NIMA joint geopotential model EGM96 website, <http://cddis.nasa.gov/926/egm96/egm96.html>.
- NGA/EGM96, National Geospatial-Intelligence Agency EGM96 website, <http://earth-info.nga.mil/GandG/wgs84/gravitymod/egm96/egm96.html>.
- NGA/WGS84, National Geospatial-Intelligence Agency WGS84 website, <https://earth-info.nga.mil/GandG/update/index.php>.
- ROM SAF, Simulation of L1 and L2 bending angles with a model ionosphere, SAF/ROM/METO/REP/RSR/017, 2015.

ROM SAF, The Radio Occultation Processing Package (ROPP) Input/Output module User Guide, SAF/ROM/METO/UG/ROPP/002, Version 11.0, 2021.

Sokolovskiy, S., Kuo, Y.-H., Rocken, C., Schreiner, W. S., Hunt, D., and Anthes, R. A., Monitoring the atmospheric boundary layer by GPS radio occultation signals recorded in the open-loop mode, *Geophys. Res. Lett.*, 33, L12 813, doi:10.1029/2006GL025 955, 2006.

Sokolovskiy, S., Rocken, C., Schreiner, W., Hunt, D., and Johnson, J., Postprocessing of L1 GPS radio occultation signals recorded in open-loop mode, *Radio Sci.*, 44, RS2002, doi:10.1029/2008RS003, 2009.

UCAR/CDAAC, COSMIC Data Analysis and Archive Center, <http://cdaac-www.cosmic.ucar.edu/cdaac>, 2014.

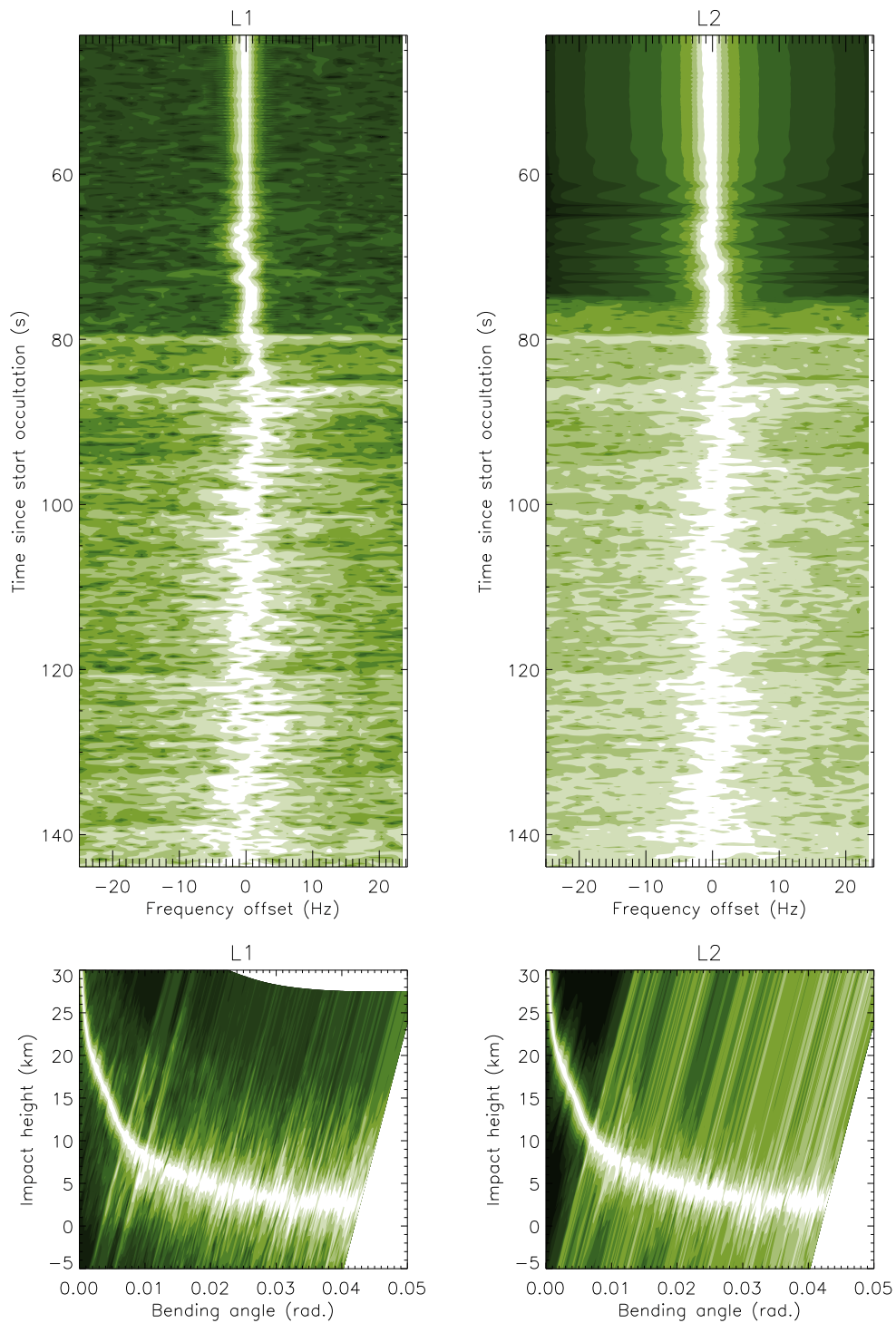


Figure 4.3: Local spatial spectra for channel L1 and L2 data for the COSMIC profile in Figure 4.1. Spectra are computed in the (time-frequency) and (bending angle-impact parameter) domains.

5 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 5.1 shows example refractivity and bending angle profiles computed from measured L1 and L2 bending angle data. The results of the simple linear combination (Equation (3.39)) 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 (3.45)).

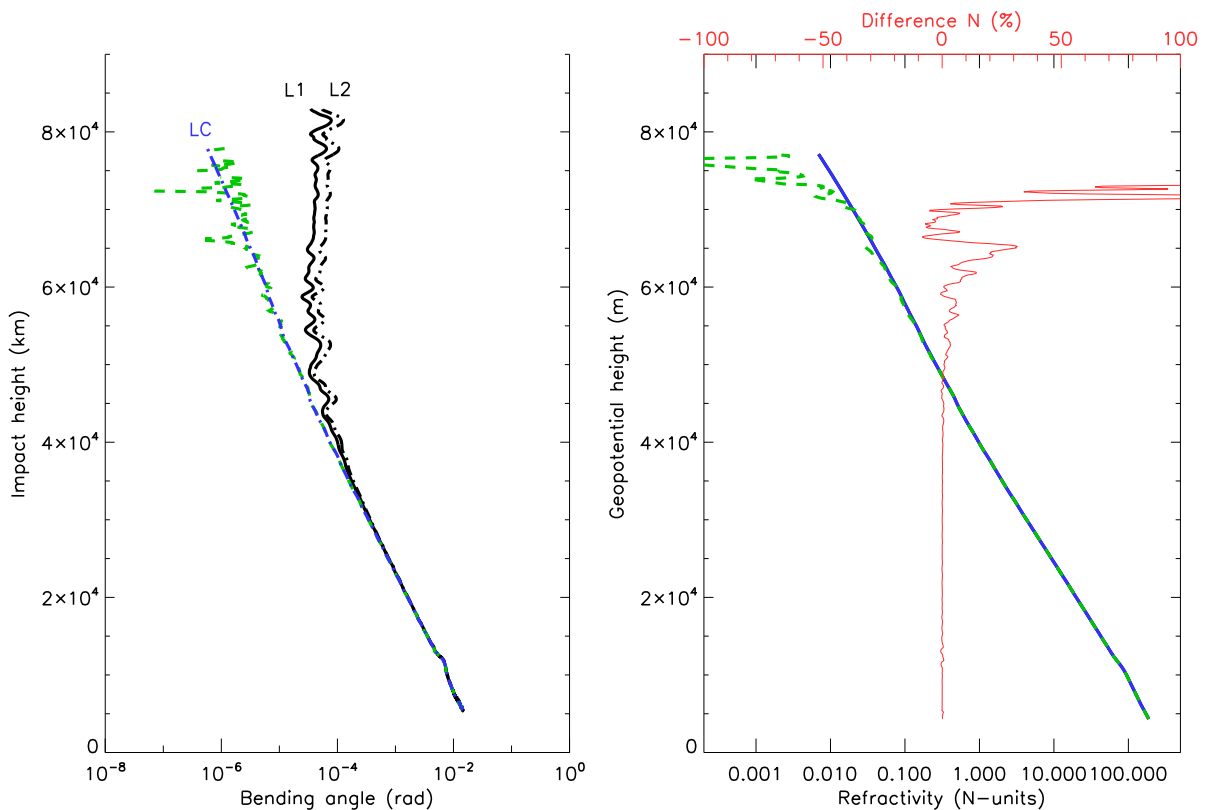


Figure 5.1: Example refractivity and bending angle profiles computed using the linear combination (green) and statistical optimization (blue) of measured bending angle data at L1 and L2 channel frequencies during a radio occultation. The difference between refractivity profiles computed using linear combination with inversion and statistical optimization with MSIS-90 climatology is plotted in red.

5.1 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 5.2 shows how the `ropp_pp` routines are integrated in the `ropp_pp_invert_tool` code.

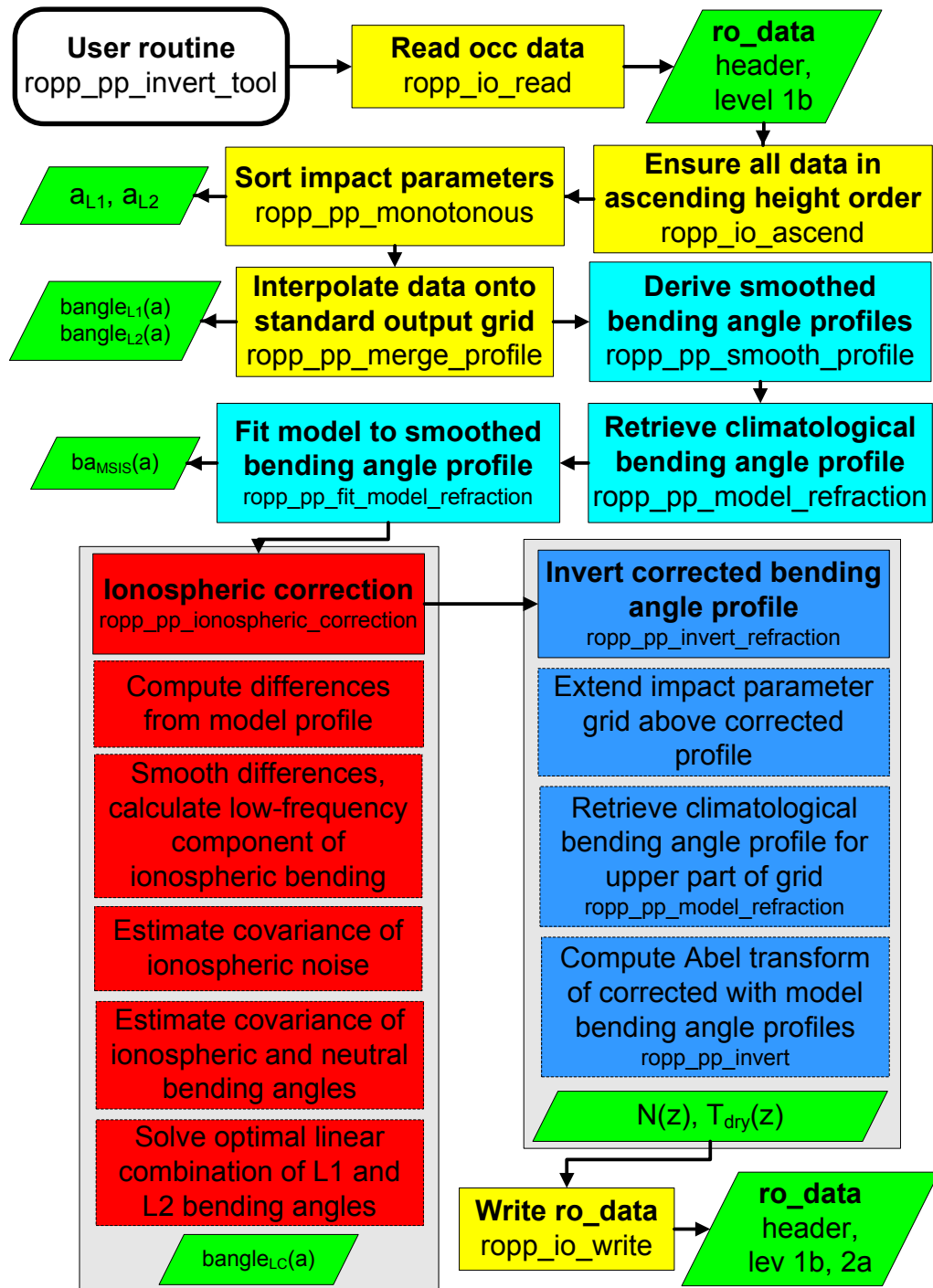


Figure 5.2: Flow chart illustrating calling tree of the ROPP pre-processor invert tool to compute ionospheric corrected bending angle and refractivity profiles from input L1 and L2 channel bending angles.

5.1.1 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, 2021b) 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 [<code>NONE,MSIS,GMSIS,BG,BARO,GBARO</code>] (default: <code>GMSIS</code>)
<code>-mfile <mfile></code>	climatology coefficients filename (default: a local file 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 (<code>VerboseMode</code>)
<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* MSISE-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 (3.39)) and no climatological or background model data are used for the Abel inversion.

5.1.2 Code organisation

Figure 5.2 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_IO User Guide (ROM SAF, 2021b).

- **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 (3.39) 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 best fitting 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 = serial` (the default being `sf_method = parallel`), 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 = serial`, 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 (3.40) 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 (3.45)) 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 (3.45) 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**

5.2 Configuration options

A number of configuration options can be defined by the user in order to tune the ionospheric correction algorithm. Table 5.1 lists the 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`. 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)
```

5.3 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, 2021b). The Level 1b bending angle data as a function of impact parameter within `R0prof` are held in a structure of type `L1btype`.

5.3.1 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
```

5.3.2 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

$$n_{\text{grid}} = 1 + \frac{\max(\text{bangle}\% \text{impact_L1}) - \min(\text{bangle}\% \text{impact_L1})}{\text{config}\% \text{dpi}} \quad (5.1)$$

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 5.1).

5.4 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 (3.39). 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 bangle_opt.)

Level-by-level longitudes, latitudes and azimuths (`lon_tp`, `lat_tp` and `azimuth_tp` respectively) are computed by calling

```
USE ropp_pp
CALL tangent_point(ro_data%lev1a%r_leo, ro_data%lev1a%r_gns, &
                  bangle%lat_tp, bangle%lon_tp, bangle%azimuth_tp)
```

This assumes that the ray follows a straight line path between the GNSS and the LEO. Alternatively, if the PPConfig configuration element `config%tp_bending` is `.TRUE.`, then the tangent point calculation takes some account of the bending of the ray, using the theory of Foelsche et al. (2011).

5.5 Ionospheric correction

By 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 (3.40).

5.5.1 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 'serial' global search method, which differs from the 'parallel' one effected by `ropp_pp_search_model_refraction`, is implemented in `ropp_pp_search_model_refraction_new`. The essential difference is that the 'serial' method, which is activated by setting `sf_method = serial` in the PPConfig configuration structure, does the search and then the fitting (see final bullet point below), whereas the older, 'parallel' method, which is activated by setting `config%sf_method = parallel`, does the fitting within the search. `sf_method = parallel` remains the default, for backwards compatibility, but `sf_method = serial` 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 = serial`, 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 logical structure 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 ('serial')
    CALL ropp_pp_search_model_refraction_new(config%mfile, impact, smooth_ba, &
                                              out_impact, model_bangle, config)
  CASE ('parallel')
    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 ('serial')
  CALL ropp_pp_fit_model_refraction_new(impact, smooth_ba, &
                                         out_impact, model_bangle, config)
CASE ('parallel')
  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 (3.44)) 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 5.1). 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.

$$rf = \frac{\frac{\sum_{i=hmin}^{i=hmax} \alpha_{smooth}(i) \alpha_{CLIM}(i)}{\sum_{i=hmin}^{i=hmax} \alpha_{CLIM}(i)^2} + (\sigma/\omega_fit)^2}{1 + (\sigma/\omega_fit)^2} \quad (5.2)$$

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 = serial`, 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 for the 2-parameter fit no longer being 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:

$$s(h) = s(h \leq h_{min}) \cos^2 \delta + s(h \geq h_{max}) \sin^2 \delta \quad (5.3)$$

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 climatological bending angles $\alpha_{CLIM}(h)$ are multiplied by $s(h)$.

5.5.2 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 `bangle_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 (5.1) 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`.

$$whi = \frac{config\%f_width * (config\%npoints - 1) * (nh - 1)}{(config\%pmax - config\%pmin) * (n_{obs} - 1)} \quad (5.4)$$

$$wei = \frac{config\%s_smooth * (nh - 1)}{config\%pmax - config\%pmin} \quad (5.5)$$

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 5.1).

The outline stages of `ropp_pp_ionospheric_correction` are as follows.

- Compute difference between observed and fitted CLIM bending angles

$$\Delta\alpha_{L1} = \alpha_{L1} - \alpha_{CLIM} \quad (5.6)$$

$$\Delta\alpha_{L2} = \alpha_{L2} - \alpha_{CLIM} \quad (5.7)$$

- 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.

$$\bar{\alpha}_I = \frac{f_2^2}{f_2^2 - f_1^2} (\Delta\bar{\alpha}_{L1} - \Delta\bar{\alpha}_{L2}) \quad (5.8)$$

- 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`

$$C_N = \frac{1}{2N_{ion}} \begin{pmatrix} \sum (\Delta\tilde{\alpha}_{L1} - \Delta\tilde{\alpha}_{L2}(f_2/f_1)^2)^2 & 0 \\ 0 & (f_1/f_2)^4 \sum (\Delta\tilde{\alpha}_{L1} - \Delta\tilde{\alpha}_{L2}(f_2/f_1)^2)^2 \end{pmatrix} \quad (5.9)$$

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

$$\sigma^{IS} = \frac{1}{N_{ion}} \sum \left(\frac{1}{2} (\Delta\tilde{\alpha}_{L1} + \Delta\tilde{\alpha}_{L2}(f_2/f_1)^2) - \bar{\alpha}_I \right)^2 - C_N(1,1) \quad (5.10)$$

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

$$\sigma^{NS} = \frac{1}{N_{neut}} \sum \left(\frac{\Delta\alpha_{L1}}{\alpha_{CLIM}} \right)^2 \quad (5.11)$$

If `config%model_err` is positive (by default it equals -0.5) and equal to σ_{static} , say, then use the fixed estimate

$$\sigma^{NS} = \sigma_{static} \quad (5.12)$$

instead.

- Estimate ionospheric signal covariance matrix

$$C_S = \begin{pmatrix} \sigma^{NS} \alpha_{CLIM}^2 & 0 \\ 0 & \sigma^{IS} \end{pmatrix} \quad (5.13)$$

- Compute quasi-inverse matrix

$$\bar{K}^{-1} = (K^T C_N^{-1} K + C_S^{-1})^{-1} K^T C_N^{-1} \quad (5.14)$$

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 (3.40).

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 (3.40)); 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, 2021b). The full composition of the `PPdiag` structure is given in Table 5.2.

5.6 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 (3.45)). 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 ionosphericly 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 (3.45) on data quality in the region of interest. This is achieved by calling `ropp_pp_invert_refraction`.

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

5.6.1 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 5.5.1 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.

5.6.2 Inverse Abel algorithm

Two inverse Abel algorithms are provided in `ropp_pp`. In both cases the algorithms solve the inverse Abel integral (Equation (3.45)) 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 (3.45) 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:

$$n_i = \exp \left[\frac{1}{\pi} \sum_{j=i}^n \Delta(\ln n)_j \right] \quad (5.15)$$

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

$$\Delta(\ln n)_j = \frac{1}{\pi \Delta a_j} \times \left[(\alpha_j a_{j+1} - \alpha_{j+1} a_j) \ln \left(\frac{a_{j+1} + \sqrt{a_{j+1}^2 - a^2}}{a_j + \sqrt{a_j^2 - a^2}} \right) + (\alpha_{j+1} - \alpha_j) \left(\sqrt{a_{j+1}^2 - a^2} - \sqrt{a_j^2 - a^2} \right) \right] \quad (5.16)$$

The infinite upper boundary condition may be accounted for by adding a further correction to the

summation of refractive index contributions, thus:

$$\Delta(\ln n)_{\text{top}} = \frac{\alpha_{\text{top}}}{\sqrt{\pi h'}} \exp\left(\frac{a_{\text{top}} - a_{\text{bot}}}{h}\right) \left[1 - \operatorname{erf}\left(\sqrt{\frac{a_{\text{top}} - a_{\text{bot}}}{h}}\right)\right] \quad (5.17)$$

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

$$\Delta(\ln n)_j = \frac{1}{\sqrt{2\pi x k_j}} \alpha_j \exp(k_j(a_j - x)) \left[\operatorname{erf}\left(\sqrt{k_j(a_{j+1} - x)}\right) - \operatorname{erf}\left(\sqrt{k_j(a_j - x)}\right) \right] \quad (5.18)$$

where

$$k_j = \frac{\ln \alpha_j / \alpha_{j+1}}{(a_{j+1} - a_j)} \quad (5.19)$$

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

$$\Delta(\ln n)_j = \frac{1}{\sqrt{2\pi x k_j}} \alpha_j \exp(k_j(a_j - x)) \left[1 - \operatorname{erf}\left(\sqrt{k_j(a_j - x)}\right)\right] \quad (5.20)$$

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

```

5.6.3 Hydrostatic integration

The ideal gas¹ 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

$$\frac{d \ln P_{\text{dry}}}{dz} = f(z, \ln P_{\text{dry}}(z)) = \frac{-g(z)N(z)}{R\kappa_1 \exp(\ln P_{\text{dry}}(z))} \quad (5.21)$$

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 (ROM SAF, 2021c) for details. The ideal gas refractivity constant $\kappa_1 = 77.60$ N-unit K hPa⁻¹ in ROPP — see Chapter 3 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

$$\ln P_{\text{dry}}(z_{\text{top}}) = \ln \left(\frac{-g(z_{\text{top}})N(z_{\text{top}})}{R\kappa_1(d \ln N/dz)} \right) \quad (5.22)$$

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 (3.48))

$$T_{\text{dry}} = \kappa_1 P_{\text{dry}}/N. \quad (5.23)$$

References

Foelsche, U., Syndergaard, S., Fritzer, J., and Kirchengast, G., Errors in GNSS radio occultation data: relevance of the measurement geometry and obliquity of profiles, *Atmospheric Measurement Techniques*, 4, 189–199, 2011.

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.

¹'Ideal gas' because the inclusion of non-ideal gas compressibility factors (see Sec 6 of the ROPP FM User Guide (2021a)) would considerably complicate Eqns (5.21) – (5.23). It is therefore necessary to use the ideal gas refractivity coefficient $\kappa_1 = 77.60$ N-unit K hPa⁻¹ in these equations.

Gorbunov, M. E., Ionospheric correction and statistical optimization of radio occultation data, *Radio Sci.*, 37, 10.1029/2000RS002370, 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) Forward model module User Guide, SAF/ROM/METO/UG/ROPP/006, Version 11.0, 2021a.

ROM SAF, The Radio Occultation Processing Package (ROPP) Input/Output module User Guide, SAF/ROM/METO/UG/ROPP/002, Version 11.0, 2021b.

ROM SAF, The Radio Occultation Processing Package (ROPP) Utilities module User Guide, SAF/ROM/METO/UG/ROPP/008, Version 11.0, 2021c.

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, 1cso = 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	(from input file)	Number of input data points
...%r_curve	(from input file)	Local radius of curvature
...%pmin	(computed)	Minimum impact parameter (IP)
...%pmax	(computed)	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	parallel	Search and fit method (parallel or serial)
...%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
...%dzt_invert	20 km	Profile depth for inversion regression

Table 5.1: Elements of the PPConfig 'configuration' structure that are used by ropp_pp ionospheric correction and Abel transform routines. The default values are used unless overwritten by configuration options that are read from an input file. In the code, the above default values are held in ropp_pp/common/ropp_pp-types.f90.

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
...%L2_min_SLTA	Lowest valid L2 SLTA (m)

Table 5.2: Elements of PPDiag structure

6 Wave optics propagator

6.1 Introduction

The aim has been to develop a relatively simple, understandable, and adaptable wave optics simulation tool, similar to that used by Sokolovskiy for various science studies (e.g., Sokolovskiy (2001); Sokolovskiy *et. al.* (2014)). It is hoped that the ROPP code will enable users to investigate specific science questions, such as assessing retrieval accuracy. In addition, the code may also become useful in the development of new forward models, for example as a baseline for the investigation of current ROPP forward model errors. The ROPP code is not yet designed for full, end-to-end simulations of real measurements, and the more sophisticated and general “WOP code”, developed by Michael Gorbunov, is recommended by the ROM SAF for those activities. However, this ROPP code will be developed further in future ROPP releases, and it should be noted that many of the component subroutines are quite general, and can be modified by the user and used individually for the use in more specific and sophisticated problems. We also note that much of the complexity of more sophisticated codes is related to effectively mapping the general problem into the idealised geometry assumed here.

6.2 The 1D ROPP wave optics tool: `ropp_pp_wopt_tool`

The main driver program, `ropp_pp_wopt_tool`, is designed to demonstrate the use of wave optics routines. It has the following main components:

1. Read in an input, output and configuration filenames from command line.
2. Set up idealised, two-dimensional geometry which assumes a spherically symmetric refractivity field.
3. Run the wave optics simulation.
4. Output the phase/amplitude and orbital parameters so that they can be read by the `ropp_pp_occ_tool` retrieval code.

The tool is called in the following way:

```
prompt> ropp_pp_wopt_tool 1d_refractivity_file -o output_file [-c config_file] [-f]
```

where `1d_refractivity_file` is an ROPP netCDF file, containing level 2a refractivity values on geopotential levels (eg from the ECMWF model) from the surface to ~ 80 km. For example, one of the 55 cases produced by ECMWF for an earlier study designed to find challenging profiles for instrument simulation

activities (Healy, 2012). (All 55 1D profiles are available from the ROM SAF web pages (ROM SAF, 2019). The corresponding 55 2D refractivity slices will also be made available on these web-pages.)

The tool currently assumes an idealised two-dimensional (2D), planar measurement geometry, with a stationary transmitter and a circular orbit for the LEO. We define a Cartesian (x,y) grid in the 2D plane, where x is essentially the direction of propagation (horizontal), and y is perpendicular to this (vertical). The 2D simulation domain is centred at $(x,y) = (0,0)$. Fig 6.1 defines the geometry.

Schematic Wave Optics Propagator Geometry

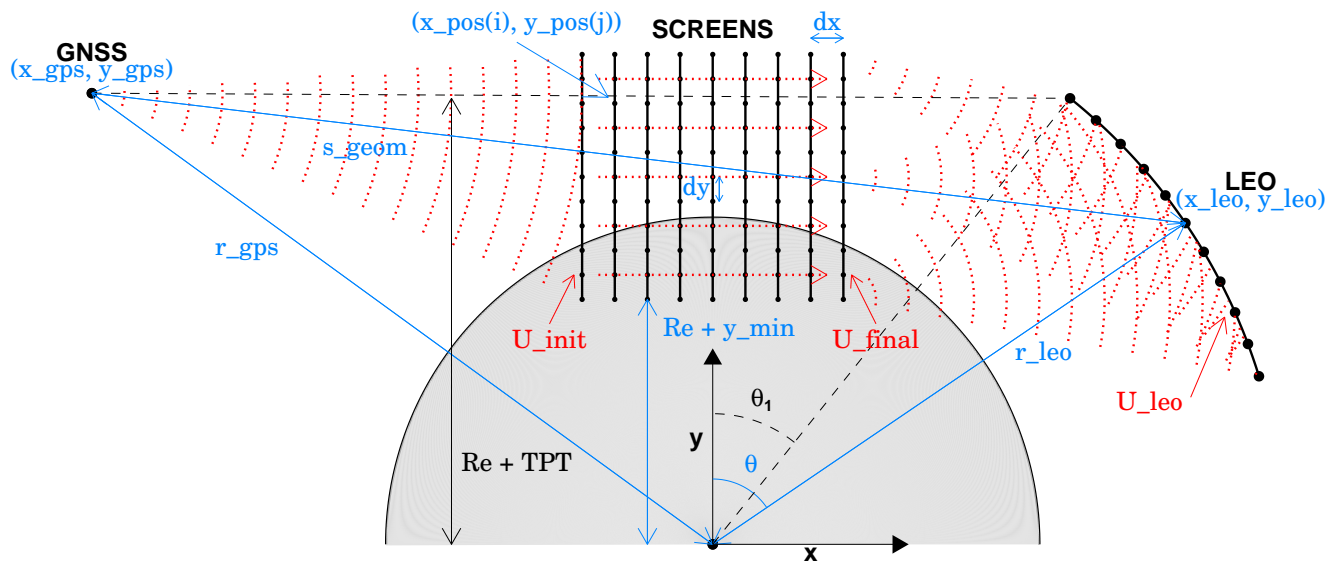


Figure 6.1: Schematic diagram of the ROPP wave optics propagator geometry. Coordinates in blue; complex RO signals in red.

The optional configuration file is a Fortran 95 namelist which contains the tunable parameters for the simulation. Two examples are included in the distribution: `ropp_pp/config/default_wopt.nml`, which defines the standard configuration (i.e. the default parameters that are used if no configuration file is supplied); and `ropp_pp/config/quick_wopt.nml`, which is a coarser resolution version of the above. The latter has little scientific value but is used in the automatic 'make test' step which is part of a routine build. `ropp_pp/config/default_wopt.nml` is reproduced here:

```

! =====
! Namelist of ROPP wave optics propagator configuration parameters
! =====
&config_wopt
! -----
! Grid specifications
! -----
nx = 401,           ! No. of horizontal screens (must be odd)
log2ny = 19,       ! log2 of number of vertical points in each screen
n_leo = 20000,     ! Number of points (and times) for which phases will be calculated
nsample = 32,      ! Number of points in each 'sample' of the final screen,
                   ! which are used to calculate the signal at the LEO
dx = 5.0e3,        ! Separation of screens (m)
dy = 1.0,          ! Separation of vertical points (m)
ymin = -3.0e5,     ! Lower limit of screens (m)
y_apodize = 1.2e5, ! Height where apodization starts (m)
leo_altitude = 8.0e5, ! LEO altitude (m)
gps_altitude = 2.02e7, ! GNSS altitude (m)
tpt_altitude = 8.0e4, ! Nominal TPT altitude (m)
delta_t = 0.005,   ! Time between phase/amp observations at LEO (s)
! -----
! Mandatory end of namelist delimiter
! -----
/

```

The simulation is for a setting occultation. The stationary GNSS satellite is at an altitude `gps_altitude` above the earth. It is convenient to consider an (r, θ) polar co-ordinate system for the circular LEO orbit, where by definition $\theta = 0$ when $x = 0$. The constant angular velocity $\dot{\theta}$ of the LEO is given by v_{LEO}/r_{LEO} , where $v_{LEO} = 7.4 \text{ kms}^{-1}$ and $r_{LEO} = R_E + \text{leo_altitude} = R_E + 800 \text{ km}$ by default. The LEO orbit starts where the straight line between the two satellites is closest to the earth at `tpt_altitude` = 80 km by default. The LEO signal is calculated every `delta_t` = 0.005 s by default.

In order that the wave optics simulator output can be subsequently processed by the ROPP "OCC" retrieval tool, (`ropp_pp_occ_tool`), which expects real, three-dimensional (3D) orbits, we assume that the measurement occurs in the equatorial plane and set all satellite velocity values perpendicular to this plane to zero. The centre of the active 2D simulation domain, ($x = 0$), is located in equatorial plane at latitude = 0, longitude = 0. The appropriate radius of curvature and undulation values are calculated for this location to be consistent with the ROPP OCC tool. Currently, a one dimensional refractivity profile is used in the simulation, so the impact of horizontal refractivity gradients is not considered. This simplification enables forward/inverse testing without the additional complication of horizontal gradient errors. A version of the tool that can handle two-dimensional refractivity fields is described in Sec 6.5.

6.3 Wave Optics Simulation

Once the simplifications of the measurement geometry described above have been introduced, the wave optics simulation problem can be split into three parts:

1. Propagate the signal from the stationary transmitter to the first “phase screen”, assuming a vacuum.
2. Model the propagation of the signal between the first and last phase screens, using the standard Multiple Phase Screen (MPS) approach (Karayel and Hinson, 1997; Sokolovskiy, 2001)
3. Propagate the signal from the final phase screen to the LEO trajectory and compute accumulated excess phase and amplitude of the signal.

6.3.1 Initialization

Computing the complex amplitude on the first screen at position (x,y) is straightforward and it is given by,

$$\mathbf{U}(x,y) = \frac{A \exp[ikd(x_1,y)]}{d(x_1,y)} \quad (6.1)$$

where k is the wavenumber, $d(x_1,y)$ is distance between the GNSS transmitter and a point on the first screen, and A is a factor that ensures the amplitude is unity at the point on the first screen that is closest to the (stationary) GNSS (i.e. the point where the straight line to the GNSS is perpendicular to the screen). $d(x_1,y)$ is given by

$$d(x_1,y)^2 = (x_{\text{GNSS}} - x_1)^2 + (y_{\text{GNSS}} - \tilde{y})^2 \quad (6.2)$$

where $\tilde{y} = R_c + u + y$. The undulation u and radius of curvature R_c are needed because the refractivities are input on refractivity altitudes, whose datum is the geoid, i.e. the surface of zero geopotential. The distance of *this* from the centre of the earth (strictly, its centre of curvature) is given by $R_c + u$. Both R_c and u are calculated by a call to the `ropp_utils` routine `occ_point`. See Sec 4.1.2 for details. As explained there, the calculation of the undulation requires some ancillary data, which is held in files whose full pathnames are given by the environment variables `GEOPOT_COEF` and `GEOPOT_CORR`. Users should therefore set these variables accordingly, or run ROPP from a place where it can find the files, such as `$ROPP_SRC/ropp_pp/tests`.

Note that Sokolovskiy (2001) often assumes a plane wave at the first screen, which is only strictly valid for a transmitter at infinity, but which nevertheless produces physically reasonable results.

6.3.2 Multiple Phase Screen Propagation

Multiple phase screen (MPS) propagation is a standard wave optics propagation approach, and it is described by, for example, Karayel and Hinson and Sokolovskiy (2001). It is straightforward to implement, as it is essentially a sequence of fast Fourier transform (FFT) calculations.

Consider the Cartesian (x,y) grid, where the signal is primarily propagating in the positive x direction (horizontal). A series of n_x screens parallel to the y axis are defined, with a fixed spacing between screens

of $x_{i+1} - x_i = \Delta x$. In the ROPP code, n_x should be an odd number, and the location of the central screen is defined to be ($x = 0$). In `ropp_pp_wopt_tool`, we currently use $n_x = 401$ and $\Delta x = 5$ km by default, so the horizontal domain spans 2000 km. These parameters might need to be adjusted by the user for specific applications. For example, Sokolovskiy *et al.* (2014) use $\Delta x = 100$ m when horizontal gradients are included. (Despite this, we have for now used the same default $\Delta x = 5$ km in the 2D version of the code, which is described in Sec 6.5.)

Each vertical screen is split into n_y vertical samples with a fixed separation, Δy . For convenience, n_y is set to a power of 2, to enable simple FFT implementations to be used in the code. It is currently defaulted to $n_y = 2^{19} = 524288$ with $\Delta y = 1$ m, so the vertical domain spans 524.288 km. This offers the future possibility of using the wave optics propagator to study ionospheric effects, possibly of very small scale if Δx is small enough. This would require the implementation of a loop over radio wave frequencies, and the encoding of frequency-dependent refractivities.

The *complex signal*, \mathbf{U} , is stored at each of the n_y positions on a screen. The MPS approach enables the computation of the wave field at x_{i+1} given the wave field at x_i . Firstly, we compute the Fourier transform of $\mathbf{U}(y, x_i)$ in the y direction, to represent the signal at x_i as a spectrum of plane waves,

$$u(k_y, x_i) = \int_{-\infty}^{\infty} \mathbf{U}(y, x_i) \exp(-ik_y y) dy. \quad (6.3)$$

For example, the m th spatial frequency is $k_y^m = \sqrt{k^2 - (2\pi m / (n_y \Delta y))^2}$, with a range ($-n_y/2 \leq m \leq n_y/2$). Each spatial frequency in this spectrum is then propagated individually, because the distance travelled between the screens will depend on the direction of the signal,

$$u(k_y^m, x_{i+1}) = u(k_y^m, x_i) \exp(ik_y^m \Delta x) \quad (6.4)$$

We then Fourier transform the signal back to phase and amplitude,

$$\mathbf{U}(y, x_{i+1}) = \int_{-\infty}^{\infty} u(k_y, x_{i+1}) \exp(ik_y y) dk_y \quad (6.5)$$

We then have to account for the differential delays associated with variation of the refractive index in the y direction. This is done by introducing a phase delay that depends on y , using the model refractivity,

$$\mathbf{U}(y, x_{i+1}) \rightarrow \mathbf{U}(y, x_{i+1}) \exp(ik\phi(y)) \quad (6.6)$$

where,

$$\phi(y) = (n(x_{i+1}, y) - 1) \Delta x = 10^{-6} N(x_{i+1}, y) \Delta x \quad (6.7)$$

The ROPP MPS code cannot produce reflected signals and it is necessary to prevent these by using a “window function” (see Eqn (5), Sokolovskiy (2001), and the related discussion). This step seems somewhat arbitrary, but it is designed to remove unphysical diffraction effects associated with an artificially sharp boundary between the atmosphere and the earth’s surface used in the simulation. It also ensures that the FFT does not alias signals near the top of the domain into erroneous signals near the bottom. The windowing is achieved by a smooth reduction of the computed signal at each screen, \mathbf{U} , to zero towards the

top and bottom of the vertical domain, using a Gaussian function. The signal reduction is applied above $y_{apodize} = 120$ km by default (Sokolovskiy, 2001) and below the surface of the earth, and uses a default width of 500 m.

A section of the ROPP MPS code is given below. The subroutine `ropp_pp_wopt_propagate` computes the Fourier transform steps, given by Eqns (6.3), (6.4) and (6.5), using an FFT. The subroutine `ropp_pp_wopt_propagate` is code developed by Michael Gorbunov, but we have implemented a simplified version. The subroutine `ropp_pp_wopt_ref_1d` computes the refractivity values on the screen, based on a cubic spline fit to the natural log of the refractivity values (see section 6.3.2). The phase delay is computed using the refractivity values on the screen, and then the window function is called `ropp_pp_wopt_window`. This code loops through all of the `nx` phase screens.

```
! loop through the screens
U(:) = U0(:)
dy = y(2) - y(1) ! this should be fixed for the entire screen
DO i = 1, nx-1
  U_tmp(:) = U(:)
! step in horizontal
  dx = x(i+1) - x(i)
  CALL ropp_pp_wopt_propagate(U_tmp, kval, dy, dx, U)
! compute the refractivity on the i+1 plane
  CALL ropp_pp_wopt_ref_1d(x(i+1), y, radius, zed, log_refrac, d2logndy2, refrac)
! now update the phases
  DO j=1, ny
    U(j) = U(j) * EXP(Ci*MODULO(kval*dx*refrac(j), 2*Pi))
  ENDDO
! multiply by the window function
  CALL ropp_pp_wopt_window(x(i+1), y, (radius+zed(1)), y_apodize, U)
ENDDO
```

The main code in the subroutine `ropp_pp_wopt_window`, which reduces the signal, U , towards the top and bottom of the vertical domain, is given by,

```
DO i = 1, ny
  IF (y(i) <= y_bot) THEN
    U(i) = U(i)*EXP(-(1.0_wp*(y(i)-y_bot)/width)**2)
  ELSE IF (y(i) >= y_top) THEN
    U(i) = U(i)*EXP(-(1.0_wp*(y(i)-y_top)/width)**2)
  ENDIF
ENDDO
```

where y_{bot} is the y -coordinate of the point where the screen meets the surface of the earth, y_{top} equals $y_{apodize}$ (in the namelist) minus `width`, and `width` is set to 500 m.

Note that if the LEO trajectory is assumed to be a straight line, parallel to the y axis, then MPS routine can be used to estimate the complex field at the LEO satellite. This approximation is often acceptable in science studies (e.g. Karayel and Hinson (1987); Solokovskiy (2001); Sokolovskiy *et al.* (2014)). However, if the LEO trajectory is not a straight line, a diffraction integral must be solved for the LEO positions (see section 6.3.3).

Cubic spline interpolation of refractivity profiles

The computation of amplitudes requires refractivity gradients that are continuous across the model level boundaries. To achieve this, we fit a cubic spline to the log of the refractivity profile, $\ln(N(z))$. The cubic spline routines used in the wave optics code are based on the section 3.3 of Press *et al.* (1992). More specifically, the existing subroutine `ropp_pp_init_spline` is based on the *natural cubic spline* solution given in Numerical Recipes subroutine `spline`.

The 1D ROPP spline initialisation routine is called in the main program as follows,

```
CALL ropp_pp_init_spline(zed, log_refrac, d2logndy2)
```

where `zed` is the geometric height of the model levels and `log_refrac` is the natural log of the refractivity values on the model levels. The output is the vector, `d2logndy2`, which contains the spline coefficients. These are subsequently used when computing the refractivity values on the MPS screens in the subroutine `ropp_pp_wopt_ref_1d`.

6.3.3 Propagation to the LEO

The propagation to the LEO satellite is probably the only unique feature in the ROPP wave optics implementation. A mathematical derivation of the approach used here can be found in ROM SAF (2018).

We need to compute the complex signal at the receiver location, $\mathbf{U}(x_r, y_r)$, given the complex signal at the final phase screen, $\mathbf{U}_f(x_f, y)$, produced by the MPS calculation. The signal at the receiver can be evaluated by integrating,

$$\mathbf{U}(x_r, y_r) = \frac{1}{\sqrt{\lambda}} \int_y \mathbf{U}_f(y) \frac{\cos \phi}{\sqrt{r}} \exp(ikr - i\pi/4) dy \quad (6.8)$$

where the integration is performed over the entire MPS final screen in the y direction. The distance between a point on the final phase screen and the receiver is $r = (x_r^2 + (y - y_r)^2)^{1/2}$, and $\cos \phi = x_r/r$ is the obliquity (or inclination) factor. (For this calculation we assume that the final screen is sited at $x = 0$.) The $-i\pi/4$ factor in the exponent arises as a result of mapping the 2D diffraction integral to 1D, and effectively integrating out the direction perpendicular to both x and y at the final screen. However, it is of no physical significance in this problem, because it is not a function of receiver position.

We show in ROM SAF (2018) that Eqn (6.8) can be approximated with,

$$\mathbf{U}(x_r, y_r) = \frac{\exp(ikx_r - i\pi/4)}{\sqrt{\lambda x_r}} \int_y \mathbf{U}_f(y) \exp\left(i \frac{k}{2x_r} (y - y_r)^2\right) dy. \quad (6.9)$$

Furthermore, if the amplitude and accumulated phase at the final screen can both be approximated with a piecewise linear functions, this integral can be computed efficiently using Fresnel integrals (see Section 7.3, Abramowitz and Stegun, 1972).

In `ropp_pp_wopt_propagate_to_leo` we compute the accumulated phase and amplitude at the ($n_y = n_y$) points in the final screen, and then restrict attention to its central points, where the signal amplitude exceed 10^{-6} times the maximum amplitude. The final screen is then subdivided into (n_y/n_{sample}) vertical intervals, and a separate integration of Eqn (6.9) is carried out on each of these so-called *mini-screens*. By default, $\Delta y = 100$ m and $n_{\text{sample}} = 32$, although larger values of n_{sample} may be sufficient for profiles with weak refractivity gradients. In each vertical interval, we fit a straightline to both the accumulated phase and amplitude values. We then write the signal in a sample as,

$$\mathbf{U}_f(y) = (a_1 + \dot{a}(y - y_1)) \exp(i\psi_1 + i\dot{\psi}(y - y_1)) \quad (6.10)$$

where $y = y_1$ at the lower boundary of the interval, (a_1, ψ_1) are the fitted amplitude and phase values at this boundary, respectively, and $(\dot{a}, \dot{\psi})$ are the amplitude and phase gradient estimates, respectively.

We show in ROM SAF (2018) that the contribution of each vertical interval to the phase and amplitude at the LEO, $\mathbf{U}(x_r, y_r)$, requires the computation of two integrals of the form,

$$I_1 = \int_0^x \exp\left(\frac{i\pi}{2}t^2\right) dt \quad (6.11)$$

which is a Fresnel integral (see Section 7.3, Abramowitz and Stegun, 1972), and computed easily (Heald (1985)), and,

$$I_2 = \int_0^x 2t \exp\left(\frac{i\pi}{2}t^2\right) dt \quad (6.12)$$

which is easy to solve after the substitution $v = t^2$. For each LEO position, the routine simply loops through the vertical intervals evaluating these integrals, and adding the contribution to $\mathbf{U}(x_r, y_r)$.

ROM SAF (2018) shows that in extreme cases it is beneficial to include the third term in the expansion of $r = (x_r^2 + (y - y_r)^2)^{1/2}$, the distance between a point on the final screen and the LEO, namely $-(y - y_r)^4/8x_r^3$. (The second term, $(y - y_r)^2/2x_r$, has been used in the derivation of Eqn (6.9).) The correction can be approximately accommodated within the Fresnel integral framework by multiplying the contribution of each mini-screen to Eqn (6.9) by $\exp\left(-ik(y_j^m - y_r)^4/8x_r^3\right)$, where y_j^m is the y -coordinate of the centre of the j th mini-screen. This approximation assumes that the additional phase factor does not vary significantly over a given vertical mini-screen, but that it can vary between them. (In the presence of very strong vertical refractivity gradients/bending angles, the variation of the phase factor $k(y_j^m - y_r)^4/8x_r^3$ across a mini-screen could amount to 0.17 rad, but it would usually be much less than this. (ROM SAF, 2018).)

Having computed $\mathbf{U}(x_r, y_r)$, we then remove the phase delay associated with the straightline path between the GNSS and LEO satellite in order to obtain the excess phase,

$$\mathbf{U}(x_r, y_r) \rightarrow \mathbf{U}(x_r, y_r) \exp(-iks_r) \quad (6.13)$$

where s_r is the length of the straight line between the GNSS satellite and the LEO at (x_r, y_r) .

We then calculate the amplitude, $A(x_r, y_r)$, and accumulated excess phase, $\phi_{ex}(x_r, y_r)$, at the LEO positions in the subroutine `ropp_pp_wopt_phase_and_amplitude_leo`. Hence, the outputs from `ropp_pp_wopt_propagate_to_leo` are:

1. the complex number, $\mathbf{U}(x_r, y_r)$
2. the amplitude, $A(x_r, y_r)$
3. the excess phase, $\phi_{ex}(x_r, y_r)$

at each LEO position.

Computing the Fresnel Integrals

The Fresnel integrals, $C(x)$ and $S(x)$, are defined as,

$$\begin{aligned} \int_0^x \exp\left(\frac{i\pi}{2}t^2\right) dt &= C(x) + iS(x) \\ S(x) &= \int_0^x \sin\left(\frac{\pi}{2}t^2\right) dt \\ C(x) &= \int_0^x \cos\left(\frac{\pi}{2}t^2\right) dt. \end{aligned} \quad (6.14)$$

See section 7.3, Abramowitz and Stegun, 1972. They can be computed efficiently using the rational polynomial approximations given by Heald (1985). Heald defines (noting an error in Heald Eqn (3),

$$C(x) = \frac{1}{2} - R_{lm}(x) \sin\left[\frac{\pi}{2}(A_{jk}(x) - x^2)\right] \quad (6.15)$$

$$S(x) = \frac{1}{2} - R_{lm}(x) \cos\left[\frac{\pi}{2}(A_{jk}(x) - x^2)\right] \quad (6.16)$$

where

$$R_{lm} = \frac{\sum_{i=0}^l c_i x^i}{\sum_{i=0}^m d_i x^i} \quad (6.17)$$

$$A_{jk} = \frac{\sum_{i=0}^j a_i x^i}{\sum_{i=0}^k b_i x^i} \quad (6.18)$$

We use the coefficients given in row 5 of Table 1, in Heald 1985, i.e. those for which $(l, m) = (5, 6)$ and $(j, k) = (4, 6)$. These are reproduced in Table 6.1 for completeness. They incur a maximum error of 4×10^{-8} .

The rational polynomial approximations are computed in the subroutine `ropp_pp_wopt_fresnel`, which is called as follows,

```
CALL ropp_pp_wopt_fresnel( (/y_upper, y_lower/), s_fresnel, c_fresnel).
```

The routine computes both the $S(x)$ (`=s_fresnel`) and $C(x)$ (`=c_fresnel`) integrals at the integration limits `y_upper` and `y_lower`.

Table 6.1: Row 5 from Table 1, Heald (1985).

R_{56}		A_{46}	
$c_0 = 1$	$d_0 = \sqrt{2}$	$a_0 = 1$	$b_0 = 2$
$c_1 = 0.7769507$	$d_1 = 2.5129806$	$a_1 = 0.1945161$	$b_1 = 2.9355041$
$c_2 = 0.6460117$	$d_2 = 2.7196741$	$a_2 = 0.2363641$	$b_2 = 2.7570246$
$c_3 = 0.3460509$	$d_3 = 1.9840524$	$a_3 = 0.068324$	$b_3 = 1.875721$
$c_4 = 0.1339259$	$d_4 = 1.0917325$	$a_4 = 0.0241212$	$b_4 = 0.978113$
$c_5 = 0.0433995$	$d_5 = 0.4205217$		$b_5 = 0.356681$
	$d_6 = 0.13634704$		$b_6 = 0.118247$

Accumulated phase at the LEO

`ropp_pp_wopt_phase_and_amplitude_leo` computes the accumulated excess phase and amplitude at the LEO using the signal using Eqn (6.13). The accumulated phase relies a phase model based on the previous accumulated phase estimates. More specifically, we use a least-squares approach to fit a straight line to the previous `nsmooth=4` accumulated excess phase values. Consider the j th complex amplitude value at the LEO, U_j . We multiply this by $\exp(-i\phi_m)$ to obtain the complex number $U_j \exp(-i\phi_m) = (x + iy)$, where ϕ_m is the phase model. The j th accumulated excess phase is then,

$$\phi_{ex}^j = \arg(x + iy) + \phi_m. \quad (6.19)$$

This excess phase is the same for both L1 and L2 components, as ionospheric effects are currently not included in the ROPP wave optics propagators. The ROPP output variables `phase_L1` and `phase_L2` therefore both contain ϕ_{ex}^j .

Amplitude at the LEO

The amplitude at the LEO is defined as the modulus of the complex signal $\mathbf{U}(x_r, y_r)$ in Eqn (6.13). This amplitude is copied directly, without any consideration of C/A, P1 or P2 code modulations, to the ROPP output variables `snr_L1ca`, `snr_L1p` and `snr_L2p`, even though it, like all fields in the ROPP wave optics propagator, is only calculated for a monochromatic L1 frequency signal. (It is not easy to suppress the output of all but one of these fields within the current ROPP infrastructure.)

6.3.4 Optional Full Spectrum Inversion

If the `-f` option to `ropp_pp_wopt_tool` is invoked, `ropp_pp_wopt_fsi_quick` carries out a Full Spectrum Inversion (FSI, Jensen et al (2003)) on the excess phase data at the LEO, to generate the corresponding bending angles. These, their impact heights and their spectral amplitudes are included as 'extra data' in the `R0prof` data structure and thence the output file. Only those bending angles with impact heights greater than `min_impact_height` (= 2 km by default), and spectral amplitudes more than `amp_threshold` (= 0.2 by default) times the maximum, are included.

The FSI is basically a Fast Fourier Transform (FFT) of the interpolated phase $\phi(t)$ and amplitude $A(t)$ at the LEO satellite. Before computing the FFT, the interpolated signal at the LEO is multiplied by a *phase model*, thus:

$$A(t) \exp(i\phi(t)) \mapsto A(t) \exp(i(\phi(t) - D_m t)) \quad (6.20)$$

where, in this simple FSI implementation, D_m is given by the minimum value of the excess Doppler at the LEO. In ROPP, the search for the minimum excess Doppler stops as soon as the signal amplitude falls below 1% of its maximum value. This prevents the inclusion of erroneous Doppler values when the amplitude is very low, which can prevent the FSI from producing any bending angles in some cases.

The FSI option allows users to compare bending angles derived from the wave optics propagator with those generated from the original refractivity profiles by, for example, the ROPP 1D forward model `ropp_fm_bg2ro_1d`, which is described in ROM SAF (2021). Bending angles can also be derived from the WOPT-derived excess phases by means of the ‘CT2’ algorithm, which lies at the heart of `ropp_pp_occ_tool`, as described in Sec 4 of this User Guide. Users wishing to exercise this option should consult Sec 4.8.

6.4 Example results

If the excess phase data produced by `ropp_pp_wopt_tool` are passed through the ROPP excess phase to refractivity (and bending angle) processing tool `ropp_pp_occ_tool` (see Sec 4), the resulting refractivity profile can be compared to the one that was input to `ropp_pp_wopt_tool`. Fig 6.2 shows two examples of this, both drawn from the 55 profile dataset of ‘challenging’ refractivity profiles. Case 7 is relatively smooth, and the refractivity retrieval is pretty good (within 1%) over nearly all the profile. Encouragingly, the same is largely true of Case 12, which has much steeper refractivity gradients at the top of the boundary layer.

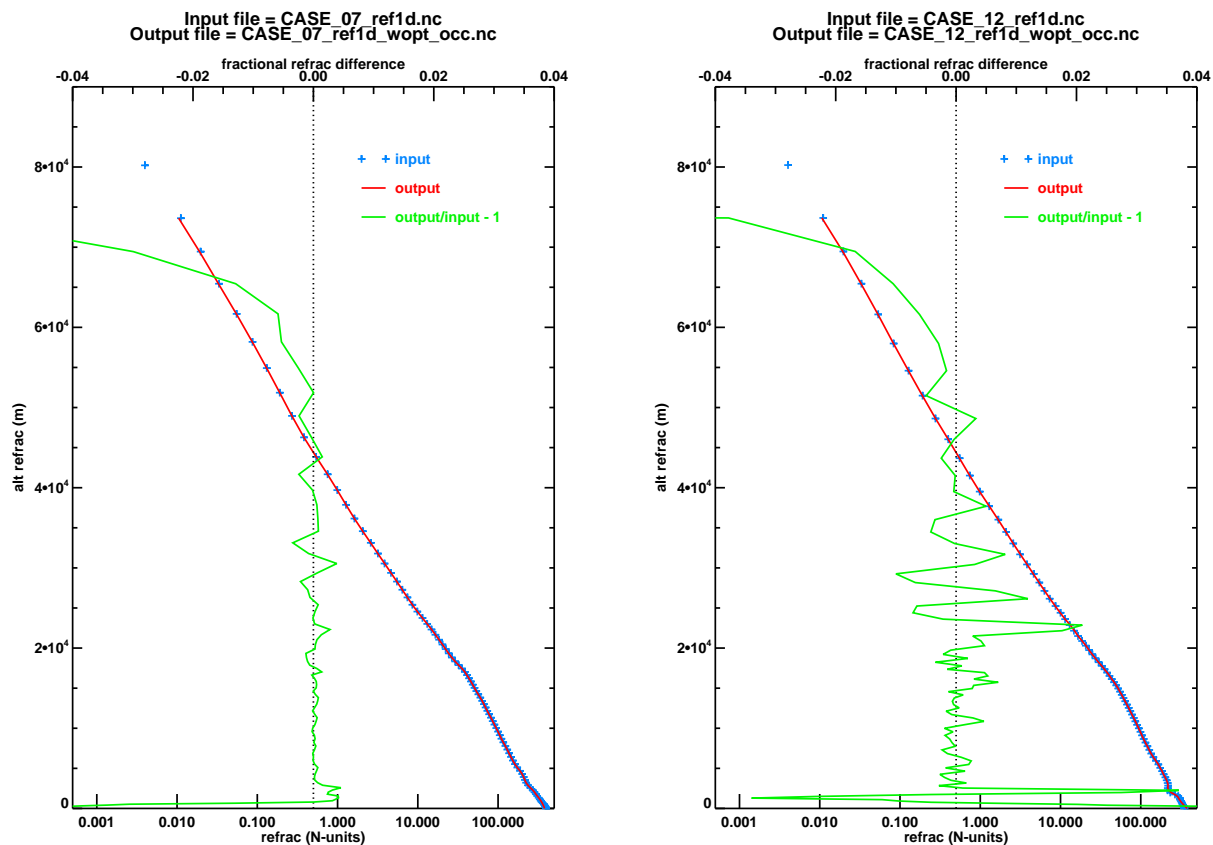


Figure 6.2: Comparison of input and derived refractivity profiles, for cases 07 and 12.

6.5 The 2D ROPP wave optics tool: ropp_pp_wopt_2D_tool

Similarly to the 1D wave optics tool, the 2D tool is called with:

```
ropp_pp_wopt_2D_tool 2d_refractivity_slice -o output_file [-c config_file] [-f]
```

where now `2d_refractivity_slice` contains refractivity information for a 2D refractivity slice. More specifically, `2d_refractivity_slice` is an ROPP netCDF file containing profiles of level 2a refractivity values on altitude levels (e.g. from the ECMWF model). The profiles should go from the surface to ~ 80 km, and they should be given at a set of horizontal locations with a uniform angular separation, thus forming a 2D slice or plane. A test dataset of 55 2D refractivity slices will be made available from the ROM SAF web pages (ROM SAF, 2019). Each 2D slice in this dataset is composed of 121 refractivity profiles, separated by 25 km in the horizontal, thereby spanning a horizontal distance of $(121-1) \times 25$ km = 3000 km. Each profile in the slice has 91 vertical levels. The central profile in a given slice – profile number = 61 – corresponds to the refractivity profile given by the corresponding 1D profile dataset, already available on the ROM SAF website. So, for example, we can now investigate the impact of horizontal gradients in, say, “case 16”, by comparing the output of the 1D computation,

```
ropp_pp_wopt_tool refrac_case16_1d.nc -o output_file_1D [-c config_file] [-f]
```

with that of the 2D wave optics computation,

```
ropp_pp_wopt_2D_tool refrac_case16.nc -o output_file_2D [-c config_file] [-f]
```

noting that, as before, the `-f` option uses `ropp_pp_wopt_fsi_quick` to perform a simplified FSI inversion suitable for noise-free simulated data.

6.5.1 Key modifications when compared to the 1D ROPP wave optics tool

The 1D and 2D wave optics tools are based on the same physics, so they share the same basic code structure and use many common routines. For example, the subroutine that computes the phase and amplitude at the LEO, `ropp_pp_wopt_phase_and_amplitude_leo`, is unchanged in the 2D tool. Therefore, this section focuses on how the 1D tool has been modified to enable a 2D simulation.

As illustrated in Fig 6.1, the geometry of the MPS calculation is Cartesian (x, y) , with wavefront propagating in the primarily x direction. During the MPS computation the refractivity has to be interpolated onto a set of screens at fixed horizontal locations x_k , separated by a constant distance Δx . The interpolated refractivity values at screen $i + 1$, $N(x_{i+1}, y)$, are used in Eqn (6.7) to compute a phase delay caused by the atmosphere,

$$\phi(y) = (n(x_{i+1}, y) - 1) \Delta x = 10^{-6} N(x_{i+1}, y) \Delta x. \quad (6.21)$$

The key difference between the 1D and 2D wave optics simulations is in the computation of the interpolated refractivity values, $N(x_{i+1}, y)$, used in Eqn (6.21) (and Eqn (6.7)). In the 1D case, the interpolated refractivity values computed in `ropp_pp_wopt_ref_1d`, $N(r)$, only depend on radius $r = \sqrt{x_{i+1}^2 + y^2}$. However, in the 2D simulation the interpolated refractivity computed in `ropp_pp_wopt_ref_2d`, $N(r, \theta)$, depends on both r and θ , where θ is the angular position within the 2D plane, given by $\theta = \tan^{-1}(x_{i+1}/y)$.

Some changes in the main the routine are required to accommodate the 2D refractivity interpolation in the MPS calculation. Firstly, two dimensional refractivity data (`TYPE(R0prof2d1) :: ro_data1`) is read by the `ropp_pp_wopt_2D_tool`, and then two new variables are defined,

```
nhoriz = ro_data1 % lev2a % nhoriz
dtheta = ro_data1 % lev2c % dtheta
```

where `nhoriz` is the number of refractivity profiles in the 2D slice, and `dtheta` is the (constant) angular separation between these profiles in radians. (`dtheta` is read in as the scalar netCDF variable of the same name; `nhoriz` is the length of the 'second' dimension of the 2D netCDF variable `alt_refrac`, called `dim_horiz` in the example netCDF file `ropp_pp/data/CASE_12_ref2d.nc`, to which the user is referred for further explanation.) The spline initialisation in the main routine now loops through the `nhoriz` profiles in the slice, as follows.

```
!-----
! 10. Initialise spline fitting of log_refrac. d2logndy2 are the spline coefficients.
!-----
```

¹The `lev2a` substructure of the `R0prof2d` structure has been extended from one to two dimensions at ROPP10.0, precisely to allow the easy ingestion of 2D refractivity data in this way.


```
CALL message( msg_info, 'Fitting splines to refractivity profiles in slice' )
ALLOCATE (d2logndy2(nlev,nhoriz))
DO i = 1, nhoriz ! each position in slice
  CALL ropp_pp_init_spline(zed(:,i), log_refrac(:,i), d2logndy2(:,i))
ENDDO
```

The 2D new MPS routine `ropp_pp_wopt_mps_2d` expects 2D arrays for height (`zed`), `log(refractivity)` (`log_refrac`) and the spline parameters (`d2logndy2`). The new variable, `dtheta`, is also included in the argument list and the subroutine call is,

```
CALL ropp_pp_wopt_mps_2d(xpos,      &
                        ypos,      &
                        roc_plus_und, & ! radius of curvature + plus undulation
                        y_apodize,  &
                        kval,       &
                        dtheta,     & ! angular separation
                        zed,        & ! 2D refrac heights
                        log_refrac,  & ! 2D log(refrac)
                        d2logndy2,  & ! for spline fit
                        U_init,     & ! phase/amp at first screen
                        U_final)    & ! phase/amp at final screen 2D
```

The computation in `ropp_pp_wopt_mps_2d` loops through the phase screens and calls `ropp_pp_wopt_ref_2d` to estimate the atmospheric delay at each screen using the 2D refractivity information,

```
DO i = 1, nx-1
  U_tmp(:) = U(:)
! step in horizontal
  dx = x(i+1) - x(i)
  CALL ropp_pp_wopt_propagate(U_tmp, kval, dy, dx, U)
! compute the refractivity on the i+1 plane
! 2D
  CALL ropp_pp_wopt_ref_2d&
  &(dtheta, x(i+1), y, radius, zed(:,i), log_refrac(:,i), d2logndy2(:,i), refrac_2d)
! now update the phases
  DO j = 1, ny
    U(j) = U(j) * EXP( ci * MODULO(kval*dx*refrac_2d(j), twopi) )
  ENDDO
```

! Windowing more complicated for 2D than 1D. Now requires horizontal interpolation,
! because the input refractivities are on model levels, whose heights depend on
! surface pressure (~orography) and therefore vary from profile to profile in
! the defining 2D refractivity profile.
! Compute radius value appropriate for `x(i+1)` in 2D windowing case

```

...
CALL ropp_pp_wopt_window(x(i+1), y, radius_window, y_apodize, U(:))
ENDDO

```

noting that some print statements and lines relating to the windowing have been removed from this section of code for clarity.

As noted above, the horizontal refractivity interpolation used in subroutine `ropp_pp_wopt_ref_2d` is the main difference between the 1D and 2D tools. In the 2D case, we have to compute θ , the angular location in the plane for each y value on the phase screen using,

```
theta = ATAN(x/(radius + y(i)))
```

noting that x is a fixed value for each phase screen. The indices of the two profiles in the 2D slice that surround θ , (kk and $kk+1$), are identified,

```
kk = INT((theta+theta_mid)/dtheta) + 1
kk = MAX(MIN(kk,nhoriz-1),1)
```

Horizontal weights, based on linear interpolation in θ between the kk and $kk+1$ locations, are then determined,

```

IF ( theta < theta_max .AND. theta > theta_min) THEN
  wt(1) = ((REAL(kk)*dtheta-theta_mid)-theta)/dtheta
  wt(2) = 1.0_wp - wt(1)
ELSE IF (theta <= theta_min) THEN
  wt(1) = 1.0_wp
  wt(2) = 0.0_wp
ELSE IF (theta >= theta_max) THEN
  wt(1) = 0.0_wp
  wt(2) = 1.0_wp
ENDIF

```

These are then used to weight the refractivity at radius r computed at the kk and $kk+1$ locations,

```

refrac(i) = 0.0_wp ! initialise
DO k = 1,2
! the theta value will change with y
  IF (hval < zed(1,kk)) THEN ! constant extrapolation below the surface,
    ! to mitigate the effect of a sharp boundary
    ! at the surface. Subsequent Gaussian
    ! windowing/damping should make the results
    ! insensitive to the way this is done.
    refrac(i) = refrac(i)+wt(k)*1.0E-6_wp*EXP(logn(1,kk))

```

```

ELSE IF (hval >= zed(nlev, kk)) THEN ! linear extrapolation of log N above model top
  grad_top = (logn(nlev, kk) - logn(nlev-1, kk)) / (zed(nlev, kk) - zed(nlev-1, kk))
  log_refrac = logn(nlev, kk) + grad_top * (hval - zed(nlev, kk))
  refrac(i) = refrac(i) + wt(k) * MAX(1.0E-6_wp * EXP(log_refrac), 1.0E-18_wp)
ELSE ! spline interpolation in between
  ipos = 1
  DO
    IF (zed(ipos+1, kk) > hval .OR. (ipos+1) == nlev) EXIT
    ipos = ipos + 1
  ENDDO
  ipos = MIN(MAX(1, ipos), nlev-1)
  h = zed(ipos+1, kk) - zed(ipos, kk)
  a = (zed(ipos+1, kk) - hval) / h
  b = 1.0_wp - a
! compute LOG(refrac) from the spline coefficients
  log_refrac = a * logn(ipos, kk) + b * logn(ipos+1, kk) + &
    ((a**3 - a) * d2logndy2(ipos, kk) + (b**3 - b) * d2logndy2(ipos+1, kk)) * (h**2) / 6.0_wp
  refrac(i) = refrac(i) + wt(k) * 1.0E-6_wp * EXP(log_refrac)
ENDIF
kk = kk + 1
ENDDO ! k

```

Note that the vertical interpolation of the kk and $kk+1$ profiles is identical to that used in the 1D code for a single profile, which is held in the `ropp_pp_wopt_ref_1d` routine.

6.5.2 An example comparing 1D and 2D retrieved bending angles

Fig 6.3 shows the tropospheric refractivity contours for “case 16”. It is a challenging case because it combines sharp vertical gradients near 1.5 km above the surface with strong horizontal gradients. In the simulations, the wave front moves in the positive x direction, from left to right. The refractivity profile at $x = 0$ in Fig 6.3 is used in the 1D wave optics simulation

```
ropp_pp_wopt_tool refrac_case16_1d.nc -o output_file_1D -c default_wopt.nml -f
```

and the 2D simulation uses the full slice,

```
ropp_pp_wopt_2D_tool refrac_case16.nc -o output_file_2D -c default_wopt.nml -f
```

We can also generate the bending angles produced when the 2D forward model is given the same refractivity field (see ROM SAF (2021)).

Fig 6.4 shows the FSI bending angles produced in the 1D (green) and 2D simulations (black), and (red) the bending angles produced by the 2D forward model (i.e. the geometrical optics limit) when it is given

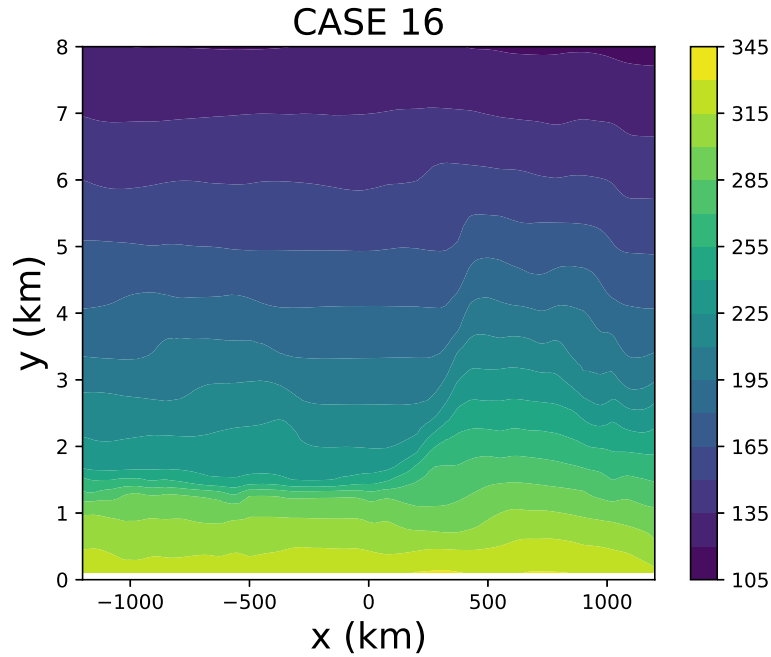


Figure 6.3: The 2D refractivity contours in the troposphere for CASE 16 used to test the 2D wave optics tool.

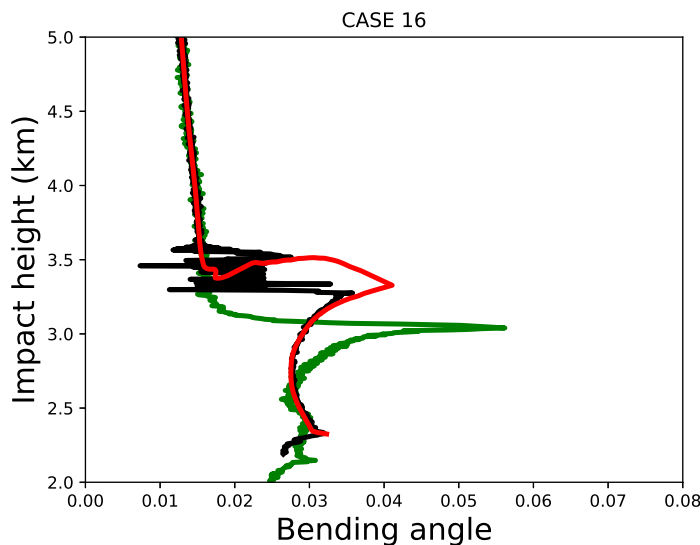


Figure 6.4: The bending angles retrieved with the FSI “quick” inversion for CASE 16, for the 1D simulation (green) and 2D simulation (black). The bending angles produced by the ROPP 2D forward model `ropp_fm_bg2ro_2d` are also shown (red). The radio waves pass from left to right in Fig 6.3.

the same 2D refractivity field. The FSI results in Fig 6.4 are not smoothed. In this case introducing realistic horizontal gradients has a significant impact on the FSI bending angles below impact heights of ~ 4 km. The noise in the 2D computation in the impact height interval between ~ 3.3 km and 3.6 km is probably

a result of “impact multipath”, caused by horizontal refractivity gradients changing the impact parameter along the ray-path. The FSI inversion approach assumes that an impact parameter cannot have multiple bending angles, but this is not the case for strong horizontal gradients. The 2D forward model bending angles are in excellent agreement with the 2D wave optics calculations, except in the impact multipath region. The horizontal gradients where $x \geq 0$ km also probably account for the upward shift of the peak bending in the 2D simulation. However, it must be recognised that validating the output of a 2D wave optics code is challenging.

References

- Abramowitz, M. and Stegun, I. A., eds., *Mathematical functions with formulas, graphs, and mathematical tables*, Dover, New York, 1972.
- Heald, M. A., Rational Approximations for the Fresnel Integrals, *Mathematics of Computation*, 44, 459–461, 1985.
- Healy, S., Optimising Tracking Strategies for Radio Occultation. Task 1 - the profile dataset, EUMETSAT/ECMWF Project Report, ECMWF, Shinfield Park, Reading, RG2 9AX, England, 2012.
- 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.
- Karayel, E. T. and Hinson, D. P., Sub-Fresnel-scale vertical resolution in atmospheric profiles from radio occultation, *Radio Sci.*, 32, 411–423, 1997.
- Press, W., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., *Numerical recipes in Fortran – The Art of Scientific Computing*, Cambridge University Press, Cambridge, New York, 2nd edn., 1992.
- ROM SAF, The Radio Occultation Meteorology Satellite Application Facility, <http://www.romsaf.org>, 2019.
- ROM SAF, Description of wave optics modelling in ROPP-9 and suggested improvements for ROPP-9.1, SAF/ROM/METO/REP/RSR/028, 2018.
- ROM SAF, The Radio Occultation Processing Package (ROPP) Forward model module User Guide, SAF/ROM/METO/UG/ROPP/006, Version 11.0, 2021.
- Sokolovskiy, S. V., Modeling and inverting radio occultation signals in the moist troposphere, *Radio Sci.*, 36, 441–458, 2001.
- Sokolovskiy, S. V., Schreiner, W., Zeng, Z., Hunt, D., Lin, Y.-C., and Kuo, Y.-H., Observation, analysis, and modeling of deep radio occultation signals: Effects of tropospheric ducts and interfering signals, *Radio Sci.*, 49, 954–970, doi:10.1002/2014RS005 436, 2014.

A Installing and using ROPP

A.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.

A.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 is 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.

A.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_utils`, `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 A.4 for further details.

A.3.1 NetCDF (optional in principle)

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/>

With effect from ROPP9.0, ROPP netCDF build support for 'classic' netCDF-4 has been dropped, which implies a need for HDF5 and, optionally, ZLIB libraries. These last two can be found at

<https://support.hdfgroup.org/HDF5/>

and

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

respectively.

In addition, the supported versions of the netCDF library are now split into two parts: a netCDF-Core library, written in C, and a netCDF-Fortran interface. The ROPP buildpack script (see Sec A.4 for more details) allows installation of these libraries as follows:

```
> buildpack zlib <compiler>
> buildpack hdf5 <compiler>
> buildpack netcdf <compiler>      (the netCDF-Core library)
> buildpack netcdfff <compiler>    (the netCDF-Fortran library)
```

These packages need to be installed in this order, since each depends on the previous one. Note, however, that the `zlib` and the `HDF5` libraries may already be installed as part of a standard Linux distribution, in which case, of course, the user need not build a local version.

Note that the `tests` subdirectory of the `ropp_io` distribution contains a simple test to check if the netCDF installation works; see Section A.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_apps`, `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_apps.a`, `libropp_fm.a` and `libropp_1dvar.a` can be compiled and installed without `ropp_io`

¹See <http://nco.sourceforge.net/>.

and therefore without 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).

A.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. Alternatively, the BUFR encoder/decoder tools `ropp2bufr_eccodes` and `bufr2ropp_eccodes` can be used if the ECMWF ecCodes library is pre-installed. If no BUFR library is detected by the installation configure script, then these tools will not be built.

The tools to BUFR-encode EUMETSAT-format grouped netCDF data, `eum2bufr` and `eum2bufr_eccodes` in `ropp_io`, require the ECMWF BUFR library or ECMWF ecCodes library to be pre-installed, respectively.

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

<https://software.ecmwf.int/wiki/display/BUFR>

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

<https://confluence.ecmwf.int/display/ECC/ecCodes+Home>

Note that a small change has been made to the ecCodes tarball supplied with ROPP to suppress the warning message that is produced each time a missing data indicator is set. This change can be made to a user's own copy of the ecCodes library by using the patch provided at `ropp_io/tools/eccodes_patch`.

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

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

or

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

while the ECMWF BUFR library would be built with

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

and the ECMWF ecCodes library would be built with

```
> buildpack eccodes <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 BUFR library or ecCodes library:

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

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

A.3.3 GRIB (optional) - either GRIB_API or ecCodes

The GRIB background reading tool `grib2bgrasc` in `ropp_io` requires either the ECMWF GRIB_API library or the ECMWF ecCodes library to be pre-installed. If neither is detected by the installation configure script, then this tool will not be built.

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

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

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

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

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

<https://confluence.ecmwf.int/display/ECC/ecCodes+Home>

The ROPP buildpack script allows installation of ecCodes by typing:

```
> buildpack eccodes <compiler>
```

A.3.4 SOFA (optional)

The routines in `ropp_utils` that transform coordinates between reference frames have the option of using the IAU Standards of Fundamental Astronomy (SOFA) library to convert between some frames. If this library is unavailable, less sophisticated formula-based versions of the routines will be used instead.

The SOFA libraries are freely available for use, provided the routines are not modified in any way. They can be downloaded from

<http://www.iausofa.org/>

The ROPP buildpack script allows installation of the SOFA library by typing:

```
> buildpack sofa <compiler>
```

A.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.

A.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.

A.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]
```

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

The shell scripts `build*_ropp`, `build_deps` and `build_ropp` have also been provided to help automate the build process by calling `buildpack` with a pre-determined sequence of packages or compilers, and to save a copy of all screen output to a disk log file. Users should review and edit these to suit their requirements. Using these tools, a complete check out of ROPP from scratch can be effected by running (in order):

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

```
> buildzlib_ropp <compiler>
> buildhdf5_ropp <compiler>
> buildnetcdf_ropp <compiler> (note that this builds the core and Fortran libs)
> buildmobufr_ropp <compiler> or buildecbufr_ropp <compiler> or buildeccodes_ropp <compiler>
> buildgrib_ropp <compiler> or buildeccodes_ropp <compiler>
> buildsofa_ropp <compiler>
> build_ropp <compiler>
```

Or, even more quickly:

```
> build_deps <compiler> zlib hdf5 netcdf netcdf mo/ecbufr/ecbufr/eccodes grib/eccodes sofa
> build_ropp <compiler>
```

A.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: `zlib`, `hdf5`, `netcdf`, `netcdf`, `mo/ecbufr`, `grib` (as required)
- 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.

A.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-11.0.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. 11.0), as in ropp_utils-11.0.tar.gz. If GNU tar is available (as on Linux systems), gzipped tar files can be unzipped with

```
> tar -xvzf ropp-11.0.tar.gz
```

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

```
> gunzip -c ropp-11.0.tar.gz | tar -xv
```

In all cases, a new subdirectory named (in the above example) ropp-11.0 will be created which contains the source code of the complete package.

A.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, nagfor, pgf95, ...) and <os> is the operating system ID, as output by the uname(1) command but entirely in lower case (linux, cygwin, ...). 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 and individual ROPP modules are all installed under \$ROPP_ROOT, i.e. the libraries can be found in the directory \$ROPP_ROOT/lib and/or \$ROPP_ROOT/lib64, and header 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 sh, ksh and bash users)
> setenv ROPP_ROOT $HOME (for csh and tcsh users)
```

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 will proceed without building the parts related to the missing library. Should the build process fail to find usable BUFR libraries, for example, and therefore fail to build the BUFR tools, `config.log` should again provide evidence on what went wrong.

A.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 A.7.

Currently supported Fortran compilers include (on Linux unless otherwise stated): Intel's `ifort` (v16 and v17); NAG's `nagfor` (v6.1); Portland Group's `pgf95` (v16); GNU `gfortran` (v4.8.5); Cray's `ftn` (v8.3.4). For the authoritative list please refer to the ROPP Release Notes and README files in each sub-package.

A.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. By default, this is `$ROPP_ROOT`. If no `--prefix` is given, the installation root directory defaults to `/usr/local` which would normally require root (`sudo`) privileges.

A.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.

A.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. Note that the netCDF libraries recommended for use with ROPP are now split in this way.)

A.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 generate output which can be compared against existing reference plots. Generally the user would only do this if one of the tests failed. If IDL is unavailable the tests will bypass this step.

A.7.1 ropp_utils

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

A.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 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 respectively 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`.

A.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. It also tests a low resolution of the wave optics propagator code, which resides in the `ropp-pp` module. 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.

A.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.

A.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 (demanding) 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.

A.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).

Further tests are run of retrievals based on COSMIC observations (refractivities and bending angles) and co-located Met Office background profiles, and of retrievals based on GRAS observations (refractivities and bending angles) and co-located ECMWF background profiles. A simple test of a retrieval using L1 and L2 bending angles is also included.

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.

A.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), even though 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 the compilation is successful, 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 local 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_ldvar`, `ropp_apps` and `ropp_fm` also allow the user to add sets of pre-defined variables to the `ROpprof` structure, which are written out in `netCDF` format with the usual variables. The first two modules do this by means of an 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 `ROpprof` 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.

B 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`)
`ropp_pp_wopt_tool.f90` (requires `ropp_io`)
`ropp_pp_wopt_2D_tool.f90` (requires `ropp_io`)

tests/

t_ropp_pp.sh (requires ropp_io)
test_pp_occ.sh (requires ropp_io)
test_pp_occ_gnos.sh (requires ropp_io)
test_pp_invert.sh (requires ropp_io)
test_pp_abel.sh (requires ropp_io)
test_pp_rs.sh (requires ropp_io)
test_pp_spectra.sh (requires ropp_io)
test_pp_wopt.sh (requires ropp_io)
test_pp_wopt_2d.sh (requires ropp_io)
ropp_test_abel.f90 (requires ropp_io)
ropp_pp_summary.f90
ropp_pp_compare.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
ropp_pp_bending_angle_gnos.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 (requires ropp_io)
ropp_pp_preprocess_grasrs.f90 (requires ropp_io)
ropp_pp_preprocess_champ.f90 (requires ropp_io)
ropp_pp_preprocess_gnos.f90 (requires ropp_io)
ropp_pp_radioholographic_filter.f90
ropp_pp_radioptic_analysis.f90
ropp_pp_set_coordinates.f90 (requires ropp_io)
ropp_pp_spectra.f90
```

wopt/

```
ropp_pp_wopt.f90
ropp_pp_wopt_fresnel.f90
ropp_pp_wopt_fsi_quick.f90
ropp_pp_wopt_mps_1d.f90
ropp_pp_wopt_mps_2d.f90
ropp_pp_wopt_phase_and_amplitude.f90
ropp_pp_wopt_phase_and_amplitude_leo.f90
ropp_pp_wopt_propagate.f90
ropp_pp_wopt_propagate_to_leo.f90
ropp_pp_wopt_ref_1d.f90
ropp_pp_wopt_ref_2d.f90
ropp_pp_wopt_window.f90
```

C ROPP extra diagnostic data

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

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

C.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 C.2.

PPDiag (config%output_diag = TRUE in ropp_pp)	
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
...%L2_min_SLTA	Lowest valid L2 SLTA (m)

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

C.3 ropp_fm_bg2ro

The extra data which are appended to the R0prof structure if the `ropp_fm` tool `ropp_fm_bg2ro_1d` is called without the `-f` option, are described in Table C.3.

R0prof (Absence of <code>-f</code> option in call to <code>ropp_fm_bg2ro_1d</code> , in <code>ropp_fm</code>)	
Structure element	Description
...%gradient_refrac	$\partial N_i / \partial x_j$ matrix
...%gradient_bangle	$\partial \alpha_i / \partial x_j$ matrix

Table C.3: Additional elements of R0prof structure, available from `ropp_fm`. See Table C.1 for the detailed structure.

C.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 C.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
...%bg_ne	Background electron density
...%bg_ne_sigma	Error in background electron density
...%res_ne	Analysis electron density
...%res_ne_sigma	Error in analysis electron density
...%VTEC_bg	VTEC of background electron density
...%VTEC_an	VTEC of analysis electron density

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

D 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_IO User Guide	SAF/ROM/METO/UG/ROPP/002	Description of ropp_io module content and functionality
ROPP_PP User Guide.	SAF/ROM/METO/UG/ROPP/004	Description of ropp_pp module content and functionality
ROPP_APPS User Guide.	SAF/ROM/METO/UG/ROPP/005	Description of ropp_apps module content and functionality
ROPP_FM User Guide.	SAF/ROM/METO/UG/ROPP/006	Description of ropp_fm module content and functionality
ROPP_1DVAR User Guide.	SAF/ROM/METO/UG/ROPP/007	Description of ropp_1dvar module content and functionality
ROPP UTILS Reference Manual	SAF/ROM/METO/RM/ROPP/001	Reference manual for the ropp_utils module
ROPP IO Reference Manual	SAF/ROM/METO/RM/ROPP/002	Reference manual for the ropp_io module
ROPP FM Reference Manual	SAF/ROM/METO/RM/ROPP/003	Reference manual for the ropp_fm module
ROPP 1D-Var Reference Manual	SAF/ROM/METO/RM/ROPP/004	Reference manual for the ropp_1dvar module
ROPP PP Reference Manual	SAF/ROM/METO/RM/ROPP/005	Reference manual for the ropp_pp module
ROPP APPS Reference Manual	SAF/ROM/METO/RM/ROPP/006	Reference manual for the ropp_apps module
WMO FM94 (BUFR) Specification for Radio Occultation Data	SAF/ROM/METO/FMT/BUFR/001	Description of BUFR template for RO data

Table D.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 D.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
The calculation of planetary boundary layer heights in ROPP	SAF/ROM/METO/REP/RSR/024	Description of ROPP PBLH diagnostics
Survey on user requirements for potential ionospheric products from EPS-SG radio occultation measurements	SAF/ROM/METO/REP/RSR/025	Results of a ROM SAF survey of the interest in possible EPS-SG ionospheric products
Estimates of GNSS radio occultation bending angle and refractivity error statistics	SAF/ROM/METO/REP/RSR/026	RO error statistics as derived by forward modelling ECMWF model errors
Recent forecast impact experiments with GPS radio occultation measurements	SAF/ROM/METO/REP/RSR/027	Impacts in NWP of 2014–2015 RO data
Description of wave optics modelling in ROPP-9 and suggested improvements for ROPP-9.1	SAF/ROM/METO/REP/RSR/028	Wave optics propagator in ROPP-9.0 and 9.1

Table D.3: ROM SAF Reports

Title	Reference	Description
Testing reprocessed GPS radio occultation datasets in a reanalysis system	SAF/ROM/METO/REP/RSR/029	Impact of reprocessed RO data on reanalyses
A first look at the feasibility of assimilating single and dual frequency bending angles	SAF/ROM/METO/REP/RSR/030	Single and dual frequency assimilation
Sensitivity of some RO measurements to the shape of the ionospheric electron density profile	SAF/ROM/METO/REP/RSR/031	Ionospheric shape sensitivity
An initial assessment of the quality of RO data from KOMPSAT-5	SAF/ROM/METO/REP/RSR/032	KOMPSAT-5 quality assessment
Some science changes in ROPP-9.1	SAF/ROM/METO/REP/RSR/033	ROPP-9.1 science
An initial assessment of the quality of RO data from Metop-C	SAF/ROM/METO/REP/RSR/034	Metop-C quality assessment
An initial assessment of the quality of RO data from FY-3D	SAF/ROM/METO/REP/RSR/035	FY-3D quality assessment
An initial assessment of the quality of RO data from PAZ	SAF/ROM/METO/REP/RSR/036	PAZ quality assessment
6 th ROM SAF User Workshop	SAF/ROM/METO/REP/RSR/037	ROM SAF-IROWG 2019 report
An initial assessment of the quality of RO data from COSMIC-2	SAF/ROM/METO/REP/RSR/038	COSMIC-2 quality assessment
Impacts of RO mission differences on trends in multi-mission data records	SAF/ROM/METO/REP/RSR/039	RO mission CDR differences
Anomalous GRAS radio occultations	SAF/ROM/METO/REP/RSR/040	Anomalous occultations
Assessment of sensitivity of the ROM SAF 1D-Var solutions to various error covariance choices	SAF/ROM/METO/REP/RSR/041	Sensitivity to error covariances
A one-dimensional variational ionospheric retrieval for truncated GNSS Radio Occultation measurements	SAF/ROM/METO/REP/RSR/042	Ionospheric 1dvar

Table D.4: ROM SAF Reports (continued)

Title	Reference	Description
CDOP-3 Proposal	SAF/ROM/DMI/MGT/CDOP3/001	Proposal for the Third Continuous Development and Operations Phase (CDOP-3) March 2017 – February 2022
Co-operation Agreement	EUM/C/85/16/DOC/19	C/A between EUMETSAT and DMI, Lead Entity for the CDOP-3 of the ROM SAF, signed at the 86th Council meeting on 7th December 2016
Product Requirements Document (PRD)	SAF/ROM/DMI/MGT/PRD/001	Detailed specification of the products of the ROM SAF
System Requirements Document (SRD)	SAF/ROM/DMI/RQ/SRD/001	Detailed specification of the system and software requirements of the ROM SAF

Table D.5: Applicable documents

E Authors

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

ROPP Authors

Name	Current institute	Contribution
Carlo Buontempo	Met Office	Savitzky-Golay thinner code.
Chris Burrows	ECMWF	2nd ROPP Test Manager. Test folder developments, improved FM vertical interpolation scheme.
Ian Culverwell	Met Office	2nd ROPP Development Manager. Documentation, testing, consolidation, IO development, GRIB2 reader, implementation of tropopause height diagnostics and planetary boundary layer height diagnostics, forward modelling of L1 and L2 bending angles, implementation of VaryChap f2–f2 FM and 1DVAR code
Axel von Englén	EUMETSAT	Author of original Test Folder system and of EUMETSAT-formatted RO data reader.
Hans Gleisner	DMI	Elements of ropp_pp, prototype GRIB2 reader, ec _{2ec} code.
Michael Gorbunov	Russian Academy of Sciences	Original pre-processor code.
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, 1D and 2D wave optics propagators, prototype VaryChap f2–f2 FM and 1DVAR code.
Helge Jønch-Sørensen	DMI	BAROCLIM code.
Kjartan Kinch	DMI	Elements of ropp_pp.
Kent Bækgaard Lauritsen	DMI	Code reviews; liaison with EUMETSAT (licences, beta tester contracts).
Huw Lewis	Met Office	1st ROPP Development Manager, FM and 1D–VAR extensions. PP module.
Owen Lewis	Met Office	BUFR developments.
Christian Marquardt	EUMETSAT	Author of majority of ROPP-1 code in UTILS, IO, FM and 1DVAR modules, and much personal, pre-existing software.
Dave Offiler	Met Office	ROPP Project Manager, IO application code and IO extensions, BUFR format/template.
Michael Rennie	ECMWF	1st ROPP Test Manager. Test folder developments.
Barbara Scherllin-Pirscher	Wegener Center	BAROCLIM (3) dataset for statistical optimisation.
Torsten Schmidt	GFZ	Guidance on tropopause height diagnostics.
Stig Syndergaard	DMI	Original spectral version of MSIS model (expansion in spherical harmonics and Chebychev polynomials), PP module developments.
Francis Warrick	Met Office	Implementation of ecCodes lib; ROPP devt and testing.
Feiqin Xie	Texas A & M	Suggested boundary layer height diagnostic algorithms.

Table E.1: Contributors to ROPP

F Copyrights

The majority of ROPP code is

© Copyright 2009-2021, 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 2018, 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.

This ROPP package may also contain a dataset available through its author, Barbara Scherllin-Pirscher, and is

© Copyright 2013-2014 Barbara Scherllin-Pirscher

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of the latest BAROCLIM (V3) dataset (the "Dataset") to use, copy, publish and merge copies of the Dataset for scientific and non-commercial purposes only and to permit persons to whom the Dataset is furnished to do so also for scientific but non-commercial purposes only, subject to the following conditions:

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

THE DATASET 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 NON-INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDER 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 DATASET OR THE USE OR OTHER DEALINGS OF THE DATASET.