The Radio Occultation Processing Package (ROPP) Applications 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)

Ref: SAF/ROM/METO/UG/ROPP/005

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 8.1	31 Dec 2015	IC	First version, for update to eighth release of ROPP (v8.1)
Version 9.0	28 Feb 2017	IC	Second version, for ninth full release of ROPP (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 Purpose of this document

This document, the ROPP_APPS User Guide ([RD.2d]), describes the applications module of the Radio Occultation Processing Package (ROPP). This module currently contains tools to calculate tropopause height and planetary boundary layer height from profiles of radio occultation data.

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 Orbited
LGPL	Lesser GPL (*q.v.*)
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 Applicaciones Cientificas – C
SAF	Satellite Application Facility (EUMETSAT)
SAG	Scientific Advisory Group
SI	Système International (The MKS units system)
TAI	Temps Atomique International (International Atomic Time)
TanDEM–X	German Earth Observation Satellite, carrying a Radio Occultation Sounder
TBC	To Be Confirmed
TBD	To Be Determined
TDB	Temps Dynamique Baricéntrique (Barycentric Dynamical Time)
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 below[1]. 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:

---

[1] Note 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 software[2]. 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 CDR[3].

1.5 Structure of this document

Section 2 briefly describes ROPP and its documentation. Section 3 describes the theory and the practical implementation of the various tropopause height (TPH) diagnostics in ROPP. These are available for bending angles, refractivities, dry temperatures and 'wet' temperatures. Section 4 does the same for the planetary boundary layer height (PBHL) diagnostics in ROPP. These are available for bending angles, refractivities, dry temperatures, 'wet' temperatures, specific humidities and relative humdidities.

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

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 interrelationships 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 indepdendent, 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.

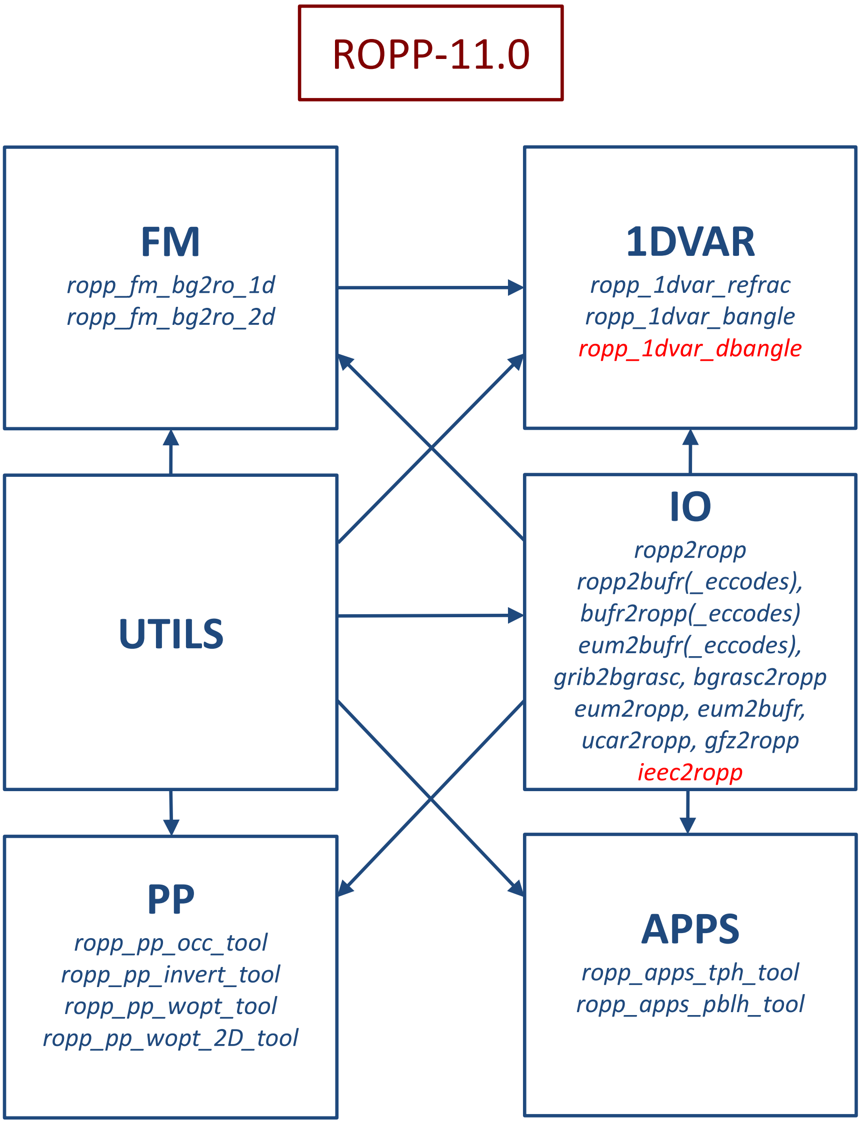

Figure 2.1: The **modules** and *tools* within ROPP-11.0. The module at the head of an arrow depends directly on the module at its tail.

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 ROPP Applications: Tropopause Height (TPH) diagnostic

The ROPP applications module (`ropp_apps`) includes the tool `ropp_apps/tools/ropp_apps_tph_tool` to diagnose the tropopause height (TPH) from profiles of bending angle, refractivity, dry temperature or (wet) temperature. These are, respectively, level 1b, 2a, 2a and 2b quantities. In each case, the TPH is diagnosed as the height of a kink at the appropriate vertical co-ordinate: impact parameter, geometric altitude, geometric altitude or geopotential height, respectively. For each of the two temperature-based tropopause heights, two TPHs are available: one based on the lapse rate and one based on the cold point.

The corresponding dependent variable at the diagnosed TPH is also recorded: the tropopause bending angle (TPA), refractivity (TPN) and temperature (TPT). The overall profile minimum temperature, PRT, and its height, PRH, are also provided for dry and 'wet' temperature profiles.

Each TPH is associated with a quality control flag, which is initialised at `ropp_MIFV` = -999 but is otherwise encoded 'bit-wise' as

$$\texttt{tph\_qc\_flag} = \sum_{r=0}^{7} l(r) 2^r \tag{3.1}$$

where the function $l(r)$ is specified in Table 3.1. If the QC flag is zero, the diagnosed TPH is therefore considered to be 'good'. Any other value indicates some question over the integrity of the derived TPH, the significance of which for the study in hand is for the user to decide. *Users are, however, recommended to use the 'good' values first, and only include those TPHs whose QC flags are non-zero if they feel confident that the overall impact of doing so is beneficial.*

TPH QC flag component definitions

Component r	Description	Value $l(r)$
0	Input data validity check	0 if OK or irrelevant; 1 if not (eg height missing).
1	Input data depth check	0 if OK or irrelevant; 1 if profile not deep enough.
2	Input data height check	0 if OK or irrelevant; 1 if profile not high enough.
3	Cov. trans. sharpness above TPH check	0 if OK or irrelevant; 1 if CT too smooth above TPH.
4	Cov. trans. sharpness below TPH check	0 if OK or irrelevant; 1 if CT too smooth below TPH.
5	Double tropopause detection	0 if OK or irrelevant; 1 if a double TP detected (in which case the lower one will be recorded).
6	TPH minimum height check	0 if OK or irrelevant; 1 if TPH below threshold.
7	TPH maximum height check	0 if OK or irrelevant; 1 if TPH above threshold.

Table 3.1: Definition of the components of `tph_qc_flag` in Eqn (3.1). Not all components are relevant to all types of TPH — for instance, the CT sharpness criteria do not apply to temperature-based TPHs. Conversely, more than one component flag might be set for any particular TPH, in which case the recorded sum will need to be decoded using Eqn (3.1).

The Lev2c substructure of the ROprof data structure has been extended to hold these QC flags, as well as the other TPH diagnostics listed in Table 3.2.

The various methods for calculating TPH are described in the following sections.

3.1 Bending angle

ROPP uses the ‘covariance transform’ method described by Lewis (2009), in which the TPH is defined as the maximum of the covariance transform of the logarithm of the bending angle, which is defined thus:

$$\tilde{f}(z) = \frac{1}{2a} \int_{\max(z_b, z-a)}^{\min(z_t, z+a)} f(z') \, [f(z') - f(z)] \, \mathrm{d}z' \tag{3.2}$$

in which $f(z) = \log(\alpha(z)/\alpha_0)$ is the natural logarithm of the bending angle α at impact parameter z, normalised by $\alpha_0 = 1$ rad, z_b (resp. z_t) is the bottom (resp. top) of the profile, and the width of the transform $2a$ is fixed at 25 km. Taking the covariance transform has the effect of sharpening the kink in $\alpha(z)$. The full algorithm is as follows.

- Ensure the impact parameters a_i are in ascending order.
- Check that some level 1b data exist. If not, return control to `ropp_apps/tools/ropp_apps_tph_tool`.
- Set QC flag = 0.
- Check the numerical robustness of the input data $\alpha(a)$:

  Are there at least two pairs (a_i, α_i) of non-missing data?

  Is a valid radius of curvature defined?

  Is a valid latitude defined?
- If any of these tests fail, set bit `TPH_QC_data_invalid` of the QC flag.
- If the undulation is missing, set it to zero, issue a warning, but carry on.
- Check the scientific robustness of the input data $\alpha(a)$:

  Do the impact heights go down to at least 15km? If not, set bit `TPH_QC_prof_depth` of the QC flag.

  Do the impact heights go up to at least 30km? If not, set bit `TPH_QC_prof_height` of the QC flag.
- If the QC flag is not zero, stop processing and return to calling program.
- Calculate the impact altitude (= impact parameter a_i − radius of curvature − undulation) and the natural logarithm of the absolute value of the normalised bending angle, $\log(|\alpha_i|/\alpha_0)$, for valid data pairs (a_i, α_i) between $2.5(3 + \cos(2\text{lat}))$ km and $2.5(7 + \cos(2\text{lat}))$ km.

- Calculate covariance transform (CT) of $f(z) = \log(\alpha/\alpha_0)$ using Eqn (3.2). See Sec 3.5 for details of the CT calculation.
- If `ropp_apps_tph_tool` is invoked with the '-d' option, add the bending angle CT, and the corresponding impact parameters, to the ROPP data structure and thence to the output file.
- Define the tropopause height (TPH) as the impact altitude of the (first) peak in the CT.
- Check that the kink in the CT of $\log\alpha$ is sharp enough to reliably define a TPH by demanding that the peak value be at least 5% greater than the average CT over the 5 km above it. If it isn't, retain the TPH but set bit `TPH_QC_CT_smooth_above` of the QC flag.
- Check that the kink in the CT of $\log\alpha$ is sharp enough to reliably define a TPH by demanding that the peak value be at least 5% greater than the average CT over the 5 km below it. If it isn't, retain the TPH but set bit `TPH_QC_CT_smooth_below` of the QC flag.
- In case of a low (< 10 km) TPH, check for the existence of a possible double tropopause by searching for a local maximum in the CT (defined to be a point with a CT at least 5% higher than the average in the 4 km range which it bisects) in the region starting 2 km above the provisional TPH. If this secondary maximum CT is at least 90% of the size of the lower maximum, interpret it as a double tropopause. Retain the lower TPH, but set bit `TPH_QC_double_trop` of the QC flag.
- Check that the TPH is greater than $2.5(3+\cos(2\text{lat}))$ km. (Should be unnecessary.) If not, set bit `TPH_QC_too_low` of the QC flag.
- Check that the TPH is lower than $2.5(7+\cos(2\text{lat}))$ km. (Should be unnecessary.) If not, set bit `TPH_QC_too_high` of the QC flag.
- Copy the QC flag to `ro_data%lev2c%tph_bangle_flag`. Set `ro_data%lev2c%tph_bangle` equal to the diagnosed TPH plus the radius of curvature plus the undulation. Set `ro_data%lev2c%tpa_bangle` equal to the bending angle at the diagnosed TPH.
- **The bending angle-derived TPH is therefore an impact parameter. The radius of curvature and undulation need to be subtracted from it to generate the impact altitude above the geoid.**

3.2 Refractivity

ROPP uses an extension of the 'covariance transform' method described by Lewis (2009). Eqn (3.2) is used again, but now $f(z) = \log(N(z)/N_0)$ is the natural logarithm of the refractivity N at refractivity altitude z, normalised by $N_0 = 1000$ N-units. $2a$ remains 25 km. The full algorithm is as follows.

- Ensure the refractivity altitudes h_i are in ascending order.
- Check that some level 2a data exist. If not, return control to `ropp_apps/tools/ropp_apps_tph_tool`.
- Set QC flag = 0.

- Check the numerical robustness of the input data $N(h)$:

  Are there at least two pairs (h_i, N_i) of non-missing data?

  Is a valid latitude defined?

- If either of these tests fail, set bit `TPH_QC_data_invalid` of the QC flag.

- Check the scientific robustness of the input data $N(h)$:

  Do the refracticity altitudes go down to at least 15km? If not, set bit `TPH_QC_prof_depth` of the QC flag.

  Do the refracticity altitudes go up to at least 30km? If not, set bit `TPH_QC_prof_height` of the QC flag.

- If the QC flag is not zero, stop processing and return to calling program.

- Calculate the the natural logarithm of the absolute value of the normalised refractivity, for valid data pairs (h_i, N_i) between $2.5(3+\cos(2\text{lat}))$ and $2.5(7+\cos(2\text{lat}))$ km.

- Calculate covariance transform (CT) of $f(z) = \log(|N|/N_0)$ using Eqn (3.2). See Sec 3.5 for details of the CT calculation.

- If `ropp_apps_tph_tool` is invoked with the '-d' option, add the refractivity CT, and the corresponding refractivity altitudes, to the ROPP data structure and thence to the output file.

- Define the tropopause height (TPH) as the refractivity altitude of the (first) peak in the CT.

- Check that the kink in the CT of $\log N$ is sharp enough to reliably define a TPH by demanding that the peak value be at least 5% greater than the average CT over the 5 km above it. If it isn't, retain the TPH but set bit `TPH_QC_CT_smooth_above` of the QC flag.

- Check that the kink in the CT of $\log N$ is sharp enough to reliably define a TPH by demanding that the peak value be at least 5% greater than the average CT over the 5 km below it. If it isn't, retain the TPH but set bit `TPH_QC_CT_smooth_below` of the QC flag.

- In case of a low ($<$ 10 km) TPH, check for the existence of a possible double tropopause by searching for a local maximum in the CT (defined to be a point with a CT at least 5% higher than the average in the 4 km range which it bisects) in the region starting 2 km above the provisional TPH. If this secondary maximum CT is at least 90% of the size of the lower maximum, interpret it as a double tropopause. Retain the lower TPH, but set bit `TPH_QC_double_trop` of the QC flag.

- Check that the TPH is greater than $2.5(3+\cos(2\text{lat}))$ km. (Should be unnecessary.) If not, set bit `TPH_QC_too_low` of the QC flag.

- Check that the TPH is lower than $2.5(7+\cos(2\text{lat}))$ km. (Should be unnecessary.) If not, set bit `TPH_QC_too_high` of the QC flag.

- Copy the QC flag to `ro_data%lev2c%tph_refrac_flag`. Set `ro_data%lev2c%tph_refrac` equal to the diagnosed TPH. Set `ro_data%lev2c%tpn_refrac` equal to the refractivity at the diagnosed TPH.

- **The refractivity-derived TPH is therefore a refractivity altitude.**

3.3 Dry temperature

ROPP follows the lapse rate method described by Reichler et al. (2003). This algorithm is expressed in terms of pressure, which is not available as a level 2a field in ROPP. However, the dry pressure can be calculated from the refractivity and dry temperature, both of which are available at this data level. We therefore use the dry pressure as a proxy for the full pressure. Throughout this Section, then, T stands for T_{dry} and p stands for p_{dry}. The algorithm in full is as follows.

- Ensure the geometric heights h_i are in ascending order.
- Check that some level 2a data exist. If not, return control to `ropp_apps/tools/ropp_apps_tph_tool`.
- Set QC flag = 0.
- Check the numerical robustness of the input data $T(h)$:

  Are there at least three pairs (T_i, h_i) of valid data?

  Is a valid latitude defined?
- If either of these tests fails, set bit `TPH_QC_data_invalid` of the QC flag.
- Check the scientific robustness of the input data $T(h)$:

  Do the geometric heights go down to at least $\mathrm{TPH}_{\mathrm{min}} = 2.5(3+\cos(2\mathrm{lat}))$ km (or 5 km if latitude is undefined)? If not, set bit `TPH_QC_prof_depth` of the QC flag.

  Do the geometric heights go up to at least $\mathrm{TPH}_{\mathrm{max}} = 2.5(7+\cos(2\mathrm{lat}))$ km (or 20 km if latitude is undefined)? If not, set bit `TPH_QC_prof_height` of the QC flag.
- If the QC flag is not zero, stop processing and return to calling program.
- Calculate the dry pressure from the refractivity and dry temperature via $p = NT/\kappa_1$, where the refractivity constant $\kappa_1 = 77.6 \times 10^{-2}$ N-unit K Pa^{-1}. If the refractivity is not available, estimate p from the dry temperature profile, by assuming T varies linearly between levels. The hydrostatic equation then implies

  $$p_{i+1}/p_i = (T_{i+1}/T_i)^{-g/R_{\mathrm{dry}}\beta_i}$$

  where

  $$\beta_i = (T_{i+1} - T_i)/(h_{i+1} - h_i) = \langle \partial T/\partial h \rangle \quad \text{over the layer.}$$

  This method requires an estimate of the pressure at the first level, p_1, which is crudely taken to be

  $$p_1 = p_{\mathrm{ref}} \exp(-gh_1/RT_1).$$

  If this estimate of p_1 is made, a warning message is issued.

- Smooth p and T according to

$$T_i \mapsto (T_{i-1} + T_i + T_{i+1})/3,$$
$$p_i \mapsto (p_{i-1} + p_i + p_{i+1})/3.$$

- Calculate the Exner pressure $\Pi_i = (p_i/p_{\text{ref}})^\kappa$, where $\kappa = R_{\text{dry}}/C_p \approx 0.285$ and $p_{\text{ref}} = 1000$ hPa.

- Calculate (-1 times) the lapse rate according to $-\Gamma_{i+1/2} = (-g/C_p)\frac{T_{i+1}-T_i}{\Pi_{i+1}-\Pi_i}\frac{\Pi_{i+1}+\Pi_i}{T_{i+1}+T_i}$.

- If `ropp_apps_tph_tool` is invoked with the '-d' option, add (-1 times) the lapse rate, and the corresponding geometric heights, to the ROPP data structure and thence to the output file.

- For each point i of the profile: if $-\Gamma_{i+1/2}$ and its average over the 2 km above are both greater than $-\Gamma_{\text{WMO}}$ = -2 K/km, *and* $-\Gamma_{i-1/2}$ is less than $-\Gamma_{\text{WMO}}$, so that $i-1/2$ and $i+1/2$ straddle the critical lapse rate, then a first estimate for the index of the lapse rate based TPH is taken to be i. The looping over i stops.

- Calculate the TPH and TPT by linear interpolation of Γ with Π, followed by linear interpolation of $\log p$ with h (since by definition the temperature is varying slowly near the tropopause):

$$\begin{aligned}
2\Pi_{\text{tph}} &= (\Pi_i + \Pi_{i-1}) + \frac{\Pi_{i+1} - \Pi_{i-1}}{\Gamma_{i+1/2} - \Gamma_{i-1/2}}(\Gamma_{\text{WMO}} - \Gamma_{i-1/2}) \\
p_{\text{tph}} &= p_{\text{ref}}\,\Pi_{\text{tph}}^{1/\kappa} \\
h_{\text{tph}} &= h_{i-1} + \frac{h_i - h_{i-1}}{\log(p_i/p_{i-1})}\log(p_{\text{tph}}/p_{i-1}) \\
T_{\text{tph}} &= T_{i-1} + \frac{T_i - T_{i-1}}{\log(p_i/p_{i-1})}\log(p_{\text{tph}}/p_{i-1})
\end{aligned}$$

- Calculate the cold point tropopause to be the height of the minimum of the temperature between TPH$_{\text{max}}$ and TPH$_{\text{min}}$. If this differs from the lapse rate derived TPH by more than 2 km, then redefine the cold point TPH to be the height of the minimum temperature within 2 km either side of the lapse rate-defined TPH. A cold point tropopause is only really meaningful in the tropics, so if the absolute value of the latitude is greater than 30°, set bit `TPH_QC_data_invalid` of the cold point TPH QC flag (ie `ro_data%lev2c%tph_tdry_cpt_flag`) and leave `ro_data%lev2c%tph_tdry_cpt = ropp_MDFV`.

- Check that the TPH is greater than TPH$_{\text{min}} = 2.5(3+\cos(2\text{lat}))$ km. If not, set bit `TPH_QC_too_low` of the QC flag.

- Check that the TPH is greater than TPH$_{\text{max}} = 2.5(7+\cos(2\text{lat}))$ km. If not, set bit `TPH_QC_too_high` of the QC flag.

- Calculate the overall profile minimum temperature and its geometric height.

- Copy the respective QC flags to `ro_data%lev2c%tph_tdry_lrt_flag`, `ro_data%lev2c%tph_tdry_cpt_flag` and `ro_data%lev2c%prh_tdry_cpt_flag`. Set the diagnosed lapse rate TPH and TPT, cold point TPH and TPT, and entire profile heights and temperatures to their equivalents in the ROPP structure, as defined in Table 3.2.

- **The dry temperature-derived TPHs are therefore geometric heights.**

3.4 Temperature

ROPP follows the lapse rate method described by Reichler et al. (2003). Since this is expressed in terms of pressure, it is directly applicable to the Level 2b fields of an ROPP profile. The algorithm in full is as follows.

- Ensure the geopotential heights z_i are in ascending order.
- Check that some level 2b data exist. If not, return control to `ropp_apps/tools/ropp_apps_tph_tool`.
- Set QC flag = 0.
- Check the numerical robustness of the input data $T(p(z))$:

  Are there at least three triplets (T_i, p_i, z_i) of valid data? This means p_i and T_i non-missing and z_i non-negative. If not, try to generate some positive geopotentials z_i from the background profile, assuming it to be in ECMWF format. Re-check the existence of valid input data.

  Are all the pressures > 0, as required by the algorithm?

  Is a valid latitude defined?
- If any of these tests fail, set bit `TPH_QC_data_invalid` of the QC flag.
- Check the scientific robustness of the input data $T(p(z))$:

  Do the geopotential heights go down to at least $\text{TPH}_{\text{min}} = 2.5(3 + \cos(2\text{lat}))$ km (or 5 km if latitude is undefined)? If not, set bit `TPH_QC_prof_depth` of the QC flag.

  Do the geopotential heights go up to at least $\text{TPH}_{\text{max}} = 2.5(7 + \cos(2\text{lat}))$ km (or 20 km if latitude is undefined)? If not, set bit `TPH_QC_prof_height` of the QC flag.
- If the QC flag is not zero, stop processing and return to calling program.
- Calculate the Exner pressure $\Pi_i = (p_i/p_{\text{ref}})^{\kappa}$, where $\kappa = R_{\text{dry}}/C_p \approx 0.285$ and $p_{\text{ref}} = 1000$ hPa.
- Calculate (-1 times) the lapse rate according to $-\Gamma_{i+1/2} = (-g/C_p)\frac{T_{i+1}-T_i}{\Pi_{i+1}-\Pi_i}\frac{\Pi_{i+1}+\Pi_i}{T_{i+1}+T_i}$.
- If `ropp_apps_tph_tool` is invoked with the '-d' option, add (-1 times) the lapse rate, and the corresponding geopotential heights, to the ROPP data structure and thence to the output file.
- For each point i of the profile: if $-\Gamma_{i+1/2}$ and its average over the 2 km above are both greater than $-\Gamma_{\text{WMO}}$ = -2 K/km, *and* $-\Gamma_{i-1/2}$ is less than $-\Gamma_{\text{WMO}}$, so that $i - 1/2$ and $i + 1/2$ straddle the critical lapse rate, then a first estimate for the index of the lapse rate based TPH is taken to be i. The looping over i stops.

- Calculate the TPH and TPT by linear interpolation of Γ with Π, followed by linear interpolation of $\log p$ with z (since by definition the temperature is varying slowly near the tropopause):

$$\begin{aligned}
2\Pi_{\text{tph}} &= (\Pi_i + \Pi_{i-1}) + \frac{\Pi_{i+1} - \Pi_{i-1}}{\Gamma_{i+1/2} - \Gamma_{i-1/2}}(\Gamma_{\text{WMO}} - \Gamma_{i-1/2}) \\
p_{\text{tph}} &= p_{\text{ref}}\,\Pi_{\text{tph}}^{1/\kappa} \\
z_{\text{tph}} &= z_{i-1} + \frac{z_i - z_{i-1}}{\log(p_i/p_{i-1})}\log(p_{\text{tph}}/p_{i-1}) \\
T_{\text{tph}} &= T_{i-1} + \frac{T_i - T_{i-1}}{\log(p_i/p_{i-1})}\log(p_{\text{tph}}/p_{i-1})
\end{aligned}$$

- Calculate the cold point tropopause to be the height of the minimum of the temperature between TPH_{max} and TPH_{min}. If this differs from the lapse rate derived TPH by more than 2 km, then redefine the cold point TPH to be the height of the minimum temperature within 2 km either side of the lapse rate-defined TPH. A cold point tropopause is only really meaningful in the tropics, so if the absolute value of the latitude is greater than 30°, set bit `TPH_QC_data_invalid` of the cold point TPH QC flag (ie `ro_data%lev2c%tph_temp_cpt_flag`) and leave `ro_data%lev2c%tph_temp_cpt = ropp_MDFV`.

- Check that the TPH is greater than $\text{TPH}_{\text{min}} = 2.5(3 + \cos(2\text{lat}))$ km. If not, set bit `TPH_QC_too_low` of the QC flag.

- Check that the TPH is greater than $\text{TPH}_{\text{max}} = 2.5(7 + \cos(2\text{lat}))$ km. If not, set bit `TPH_QC_too_high` of the QC flag.

- Calculate the overall profile minimum temperature and its geopotential height.

- Copy the respective QC flags to `ro_data%lev2c%tph_temp_lrt_flag`, `ro_data%lev2c%tph_temp_cpt_flag` and `ro_data%lev2c%prh_temp_cpt_flag`. Set the diagnosed lapse rate TPH and TPT, cold point TPH and TPT, and entire profile heights and temperatures to their equivalents in the ROPP structure, as defined in Table 3.2.

- **The temperature-derived TPHs are therefore geopotential heights.**

3.5 Covariance transformation

The covariance transform in Eqn (3.2) is estimated numerically as follows.

- The lowest index i_L satisfying $z(i_L) > z_L = \max(z_b, z - a)$ is found.

- The highest index i_U satisfying $z(i_U) < z_U = \min(z_t, z + a)$ is found.

- The body of the integral is estimated using the trapezium rule:

$$\int_{z(i_L)}^{z(i_U)} f(z')\,[f(z') - f(z)]\,dz' \approx (1/2)\sum_{i=i_L}^{i_U-1}(f_i h_i + f_{i+1}h_{i+1})(z_{i+1} - z_i) \tag{3.3}$$

where $h_i = f_i - f_j$. (j is the index of the point for which the covariance transform is being computed, corresponding to z in the integral formulation Eqn (3.2) — ie, $z = z_j$.)

- A correction is made at the lower limit by linearly extrapolating $f(z)$ below $z(i_L)$:

$$\begin{aligned}\int_{z_L}^{z(i_L)} f(z')\,[f(z') - f(z)]\,dz' \quad &\approx \quad \Delta z f(i_L)(f(i_L) - f(j)) - \\ &\quad m(\Delta z)^2(f(i_L) - f(j)/2) + \\ &\quad m^2(\Delta z)^3/3 \end{aligned} \tag{3.4}$$

where $m = (f(i_L+1) - f(i_L))/(z(i_L+1) - z(i_L)) \approx f'(z_L)$ and $\Delta z = z(i_L) - z_L > 0$.

- A correction is made at the upper limit by linearly extrapolating $f(z)$ above $z(i_U)$:

$$\begin{aligned}\int_{z(i_U)}^{z_U} f(z')\,[f(z') - f(z)]\,dz' \quad &\approx \quad \Delta z f(i_U)(f(i_U) - f(j)) + \\ &\quad m(\Delta z)^2(f(i_U) - f(j)/2) + \\ &\quad m^2(\Delta z)^3/3 \end{aligned} \tag{3.5}$$

where $m = (f(i_U) - f(i_U-1))/(z(i_U) - z(i_U-1)) \approx f'(z_U)$ and $\Delta z = z_U - z(i_U) > 0$.

- The covariance transform at z_j is then given by the sum of Eqns (3.3), (3.4) and (3.5), divided by $2a$.

The $f(z') - f(z)$ term in the integrand of Eqn (3.2) is largest in magnitude at the limits of integration. This makes it important to handle 'edge effects' carefully, as otherwise the resulting numerical estimates of the integral are sensitive to the resolution of the input data, and show 'jags' as large terms drop in or out of the sums when the calculation moves from one level to another.

The choice of $2a = 25$ km for bending angle and refractivity are the result of some experimentation with a variety of occultation profiles. Users may alter them by editing `tph_cov_width` in `ropp_apps_tph_bangle.f90` or `ropp_apps_tph_refrac.f90` respectively.

3.6 Calculating the TPH diagnostics

For `ropp_apps_tph_tool` to try to calculate all four TPHs, the user need simply call

```
ropp_apps_tph_tool <inputROPPfile> -o <outputROPPfile>
```

The user may instead choose to calculate just the bending angle-, refractivity-, dry temperature- or temperature-based TPH by calling `ropp_apps_tph_tool` with the '-b', '-n', '-y' or '-t' flags respectively. (Two or more of these can be requested at the same time; specifying none is equivalent to requesting all four of them.) Invoking the '-d' option causes more diagnostics to be written to standard output, as well as the covariance transform of bending angle and refractivity and/or the lapse rate of dry temperature and temperature to be added to the output netCDF file.

3.7 Examples of the TPH diagnostics

Figure 3.1 shows all six tropopause heights plotted for a GRAS occultation with a co-located ECMWF background profile from 1 May 2009. (These are the example data provided with the ROPP distribution.) It can be seen that the covariance transforms of the bending angle and refractivity are strongly peaked at the tropopause, the former slightly more sharply (but less smoothly) than the latter. The lapse rates of dry and wet temperature can also be seen to define the tropopause reasonably sharply. The six tropopause heights are reasonably close to each other, each being within 500 m of the average. (Note that for a strict comparison, the impact parameter-based bending angle tropopause height should be reduced to account for the finite refractivity in the profile. This reduction amounts to about 200–500 m at the TPH. In addition, the temperature-based TPH is a geopotential height, which should be converted to a geometric height for a direct comparison. This difference is less than 50 m at 15 km.) The cold point temperature-based TPHs are larger than their lapse rate counterparts, because $\Gamma = 0$ K/km is higher up the profile than $\Gamma = 2$ K/km.

Further examination of an early version of the ROPP TPH routines, including comparison against GRACE-A and TerraSAR-X dry temperature tropopause heights, can be found in the beta review of ROPP 7.0 (Schmidt, 2013).

References

Lewis, H. W., A robust method for tropopause altitude identification using GPS radio occultation data, *Geophys. Res. Lett.*, *36*, L12 808, 2009.

Reichler, T., Dameris, M., and Sausen, R., Determining the tropopause height from gridded data, *Geophys. Res. Lett.*, *30*, 2042, 2003.

Schmidt, T., Visiting Scientist Report 22: Beta testing of ROPP 7.0, SAF/ROM/DMI/REP/VS22/001, Version 1.0, 2013.

TPH parameter definitions

Element	Type(kind)	Definition	Values
tph_bangle	Real(8)	Tropopause impact parameter (m)	`ropp_MDFV` initially/incalculable; otherwise a ‘valid’ TPH.
tpa_bangle	Real(8)	Tropopause bending angle (rad)	`ropp_MDFV` initially/incalculable; otherwise a ‘valid’ TPA.
tph_bangle_flag	Int(2)	Bending angle TPH QC flag	`ropp_MIFV` initially/incalculable; 0 – 2^8-1 otherwise.
tph_refrac	Real(4)	Tropopause altitude (m)	`ropp_MDFV` initially/incalculable; otherwise a ‘valid’ TPH.
tpn_refrac	Real(8)	Tropopause refractivity (N-unit)	`ropp_MDFV` initially/incalculable; otherwise a ‘valid’ TPN.
tph_refrac_flag	Int(2)	Refractivity TPH QC flag	`ropp_MIFV` initially/incalculable; 0 – 2^8-1 otherwise.
tph_tdry_lrt	Real(4)	Tropopause altitude (m) (lapse rate)	`ropp_MDFV` initially/incalculable; otherwise a ‘valid’ TPH.
tpt_tdry_lrt	Real(4)	Tropopause dry temperature (K)	`ropp_MDFV` initially/incalculable; otherwise a ‘valid’ TPT.
tph_tdry_lrt_flag	Int(2)	Dry temperature TPH QC flag	`ropp_MIFV` initially/incalculable; 0 – 2^8-1 otherwise.
tph_tdry_cpt	Real(4)	Tropopause altitude (m) (cold point)	`ropp_MDFV` initially/incalculable; otherwise a ‘valid’ TPH.
tpt_tdry_cpt	Real(4)	Tropopause dry temperature (K)	`ropp_MDFV` initially/incalculable; otherwise a ‘valid’ TPT.
tph_tdry_cpt_flag	Int(2)	Dry temperature TPH QC flag	`ropp_MIFV` initially/incalculable; 0 – 2^8-1 otherwise.
prh_tdry_cpt	Real(4)	Entire profile cold point altitude (m)	`ropp_MDFV` initially/incalculable; otherwise a ‘valid’ PRH.
prt_tdry_cpt	Real(4)	Entire profile cold point dry temperature (K)	`ropp_MDFV` initially/incalculable; otherwise a ‘valid’ PRT.
prh_tdry_cpt_flag	Int(2)	Entire profile cold point QC flag	`ropp_MIFV` initially/incalculable; 0 – 2^8-1 otherwise.
tph_temp_lrt	Real(4)	Tropopause geopotential height (m) (lapse rate)	`ropp_MDFV` initially/incalculable; otherwise a ‘valid’ TPH.
tpt_temp_lrt	Real(4)	Tropopause temperature (K)	`ropp_MDFV` initially/incalculable; otherwise a ‘valid’ TPT.
tph_temp_lrt_flag	Int(2)	Temperature TPH QC flag	`ropp_MIFV` initially/incalculable; 0 – 2^8-1 otherwise.
tph_temp_cpt	Real(4)	Tropopause geopotential height (m) (cold point)	`ropp_MDFV` initially/incalculable; otherwise a ‘valid’ TPH.
tpt_temp_cpt	Real(4)	Tropopause temperature (K)	`ropp_MDFV` initially/incalculable; otherwise a ‘valid’ TPT.
tph_temp_cpt_flag	Int(2)	Temperature TPH QC flag	`ropp_MIFV` initially/incalculable; 0 – 2^8-1 otherwise.
prh_temp_cpt	Real(4)	Entire profile cold point altitude (m)	`ropp_MDFV` initially/incalculable; otherwise a ‘valid’ PRH.
prt_temp_cpt	Real(4)	Entire profile cold point temperature (K)	`ropp_MDFV` initially/incalculable; otherwise a ‘valid’ PRT.
prh_temp_cpt_flag	Int(2)	Entire profile cold point QC flag	`ropp_MIFV` initially/incalculable; 0 – 2^8-1 otherwise.

Table 3.2: Elements of `ro_data%Lev2c` substructure relating to TPH

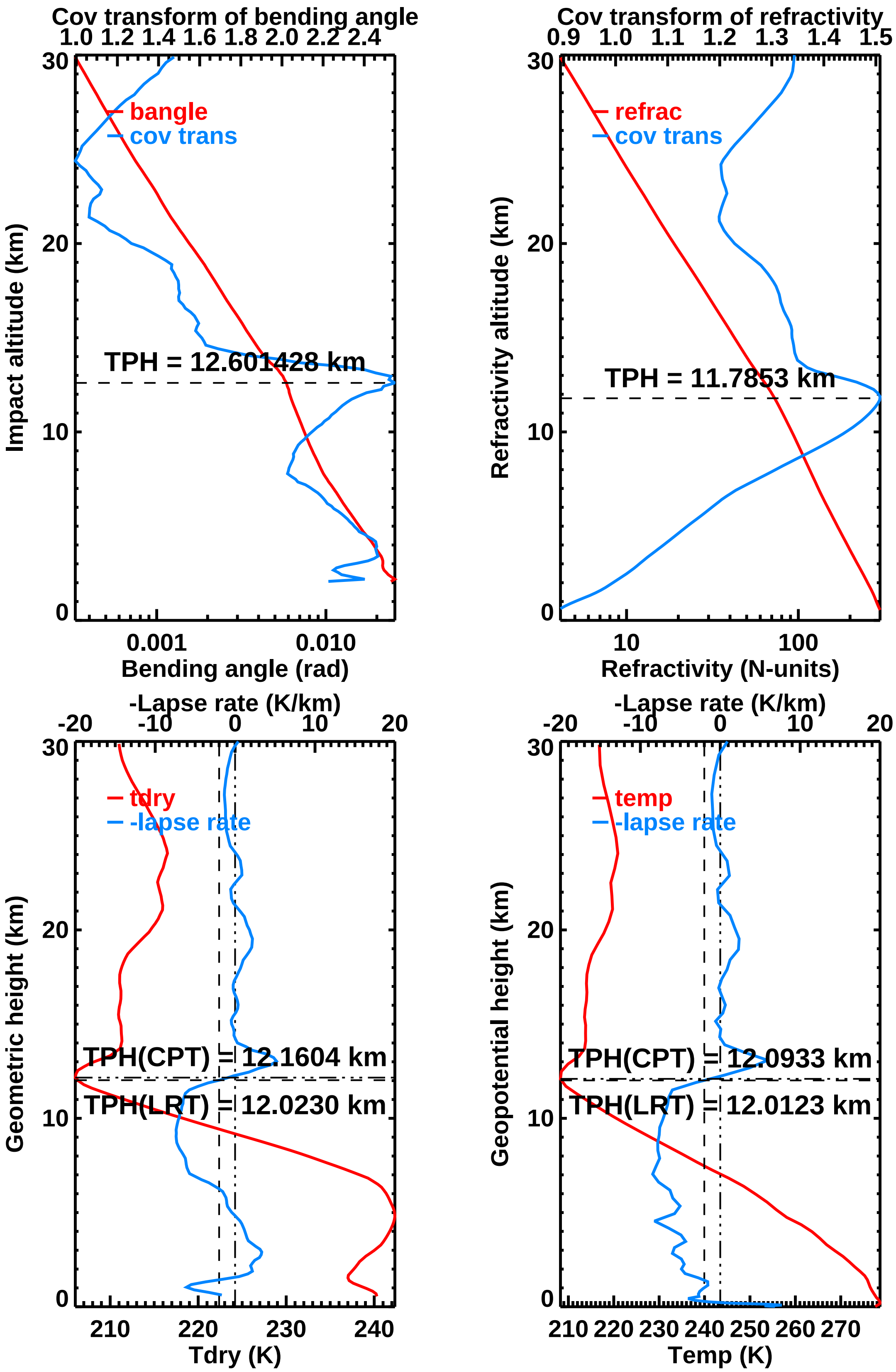

Figure 3.1: Example tropopause heights, 1 May 2009, latitude=27S. Top left: GRAS bending angle and covariance transform. Top right: GRAS refractivity and covariance transform. Bottom left: dry temperature and lapse rate. Bottom right: temperature and lapse rate from co-located ECMWF background.

4 ROPP Applications: Planetary Boundary Layer Height (PBLH) diagnostic

In view of the uncertainty — ontological, definitional and observational — in the 'true' Planetary Boundary Layer Height (PBLH), it was considered useful to generate as many PBLHs as possible in ROPP, so that the user can get some feel for the reliability of the resulting estimates. The ROPP applications module (`ropp_apps`) therefore includes a tool (`ropp_apps/tools/ropp_apps_pblh_tool`) to diagnose the PBLH from profiles of bending angle, refractivity, dry temperature, background temperature, background specific humidity and background relative humidity. These are, respectively, level 1b, 2a, 2a, 2b, 2b and 2b quantities. The first three are observational; the last three are based on model quantities. In all cases, the PBLH is diagnosed as the height of a minimum (or, in the case of temperature and dry temperature, a maximum) in the vertical gradient of the field in question. These definitions result from the work of Xie (Xie, 2014) who, as part of a ROM SAF Visiting Scientist Contract, investigated various PBLH diagnostics derived from high resolution ERA reanalysis data (Xie, 2014).

In fact, as a result of discussions with experts on PBLH, two boundary layer heights are recorded (if possible): the ones associated with the strongest and second strongest extrema in th vertical gradients. (Many profiles have even more than two such possible PBLHs.)

The corresponding dependent variable at the diagnosed PBLH is also recorded: the boundary layer bending angle, refractivity etc.

Each PBLH is associated with a quality control flag, which is initialised at `ropp_MIFV` = -999 but is otherwise encoded 'bit-wise' as

$$\texttt{pblh\_qc\_flag} = \sum_{r=0}^{13} l(r) 2^r \tag{4.1}$$

where the function $l(r)$ is specified in Table 4.1. Thus, if the QC flag is zero or greater than $2^5 = 32$, the diagnosed PBLH is therefore considered to be 'good', although any confidence the user may place on the resulting value should be tempered by consideration of:

- The similarity to the PBLHs derived from other variables of the same profile;
- The nature and number of other possible PBLHs that might have been detected from the same profile — in particular, the one calculated (and stored in the `ROprof` profile structure and output file) for the second strongest minimum (or maximum for T and T_{dry}) in the profile; and
- The geographical location of the profile, as stored in the QC flag ('land' or 'coastal' values are perhaps more likely to be more extreme and/or variable than those over subtropical oceans, for example).

The Lev2c substructure of the ROprof data structure has been extended to hold these QC flags, as well as the other PBLH diagnostics listed in Table 4.2.

PBLH QC flag component definitions		
Component r	Description	Value $l(r)$
0	Input data validity check	0 if OK or irrelevant; 1 if not (eg height missing).
1	Input data depth check	0 if OK or irrelevant; 1 if profile not deep enough.
2	Input data height check	0 if OK or irrelevant; 1 if profile not high enough.
3	PBLH minimum height check	0 if OK or irrelevant; 1 if PBLH below threshold.
4	PBLH maximum height check	0 if OK or irrelevant; 1 if PBLH above threshold.
5	Missing longitude	0 if longitude present; 1 if not.
6	Missing latitude	0 if latitude present; 1 if not.
7	Double PBLHs	1 if precisely two PBLHs detected; 0 otherwise.
8	Multiple PBLHs	1 if at least three PBLHs detected; 0 otherwise.
9	Land location	1 if PBLH detected over land.
10	Coastal location	1 if PBLH detected over coast.
11	Polar location	1 if PBLH detected over polar regions.
12	Subtropical location	1 if PBLH detected over subtropical ocean.
13	Tropical location	1 if PBLH detected over tropical ocean.

Table 4.1: Definition of the components of `pblh_qc_flag` in Eqn (4.1). More than one component flag might be set for any particular PBLH, of course, in which case the recorded sum will need to be decoded using Eqn (4.1).

The various methods for calculating PBLH in ROPP are described in the following sections.

4.1 Bending angle

ROPP follows the advice of Xie (Xie, 2014) and defines the first (and second, if possible) bending-angle-based boundary layer heights, PBLH$_\alpha$, as the heights of minima of the vertical derivative of the bending angle:

$$PBLH_\alpha = \arg\min(\partial\alpha/\partial h) \tag{4.2}$$

in which α is the bending angle and h is the vertical geometric distance above the Earth's surface. Note that, unlike the case of tropopause height determination (see Sec 3), where the impact height at the tropopause is reasonably close, proportionately speaking, to the height above the surface, for PBLH determination one must work in terms of a geometrical distance at the outset.

The full algorithm in `ropp_apps/pblh/ropp_apps_pblh_bangle.f90` is as follows.

- Ensure the impact parameters a_i are in ascending order.
- Check that some level 1b data exist. If not, return control to `ropp_apps/tools/ropp_apps_pblh_tool`.
- Set QC flag $= 0$.
- Check the numerical robustness of the input data $\alpha(a)$:
    - Are there at least two pairs (a_i, α_i) of non-missing data? If not, set bit `PBLH_QC_data_invalid` of the QC flag.

  - Is a valid longitude defined? If not, issue a warning.
  - Is a valid latitude defined? If not, assume it to be zero and issue a warning.
  - Is a valid surface geopotential defined? If not, assume it to be zero and issue a warning.

- Convert impact parameters a to geometric heights h by calling subroutine `ropp_apps_impact2geom` in `ropp_apps/common/ropp_apps_utils.f90`, which iteratively solves

$$a = rn(r) \tag{4.3}$$

  where $n = 1 + 10^{-6}N$ is the refractive index of air, N is the refractivity, and $r = h + R_c + u$ where R_c is the radius of curvature and u is the undulation. If the refractivities or radius of curvature are unavailable, or Eqn (4.3) is otherwise insoluble, bit `PBLH_QC_data_invalid` of the QC flag is set and control is returned to the calling routine. (Missing undulations are set to zero and a warning is issued.)

- Convert impact heights to heights above surface by subtracting height of the surface geopotential.

- Check the scientific robustness of the input data profile $\alpha(a)$:

  - Is the maximum height at least 5000 m? If not, set bit `PBLH_QC_prof_height` of the QC flag. (5000 m is a reasonable maximum PBLH over, say, the Sahara at the end of a summer day.)
  - Is the minimum height at most 300 m? If not, set bit `PBLH_QC_prof_depth` of the QC flag. (Setting a lower limit of 300 m should avoid difficulties arising from the surface-based inversions that are sometimes seen over tropical and subtropical oceans, as highlighted by von Engeln and Teixeira (2004).)

- If the QC flag is not zero, stop processing and return to calling program.

- Apply 1–2–1 smoothing to the bending angles α, thus:

$$\alpha_i \mapsto (\alpha_{i-1} + 2\alpha_i + \alpha_{i+1})/4$$

- Calculate the vertical derivative Γ at the half-integer points:

$$\Gamma_{i+1/2} := (\partial\alpha/\partial h)_{i+1/2} = (\alpha_{i+1} - \alpha_i)/(h_{i+1} - h_i)$$

- Find the number of, and indices of, all the local minima of $\partial\alpha/\partial h$ between 300 m and 5000 m. If no minima are found, set bit `PBLH_QC_data_invalid` of the QC flag. If one PBLH is found, calculate its position by calling subroutine `ropp_apps_pblh_locate` in `ropp_apps/common/ropp_apps_utils.f90`. If two or more are found, calculate their positions by calling it twice, once for each of the strongest two minima. `ropp_apps_pblh_locate` estimates the location of the minimum (or maximum) of the gradient by fitting a quadratic through $(\Gamma_{i*-1/2}, h_{i*-1/2})$, $(\Gamma_{i*+1/2}, h_{i*+1/2})$ and $(\Gamma_{i*+3/2}, h_{i*+3/2})$, where $i*+1/2$ is the index of the local extremum in the vertical gradient Γ. The location of the minimum of this quadratic defines the estimated PBLH, $h*$, as well as the the bending angle at the

PBLH, $\alpha*$, which may be of interest. We find, if $i*+1/2$ is the location of the minimum of $\partial\alpha/\partial h$:

$$\begin{aligned} h* &= (1/2)(h_{i*}+h_{i*+1})-(a/2b) \\ \alpha* &= (1/2)(\alpha_{i*}+\alpha_{i*+1})+(a/2b)\left(-\Gamma_{i*+1/2}+(a^2/6b)\right) \end{aligned}$$

where

$$\begin{aligned} a &= \frac{\Gamma'_+\Delta h_- + \Gamma'_-\Delta h_+}{\Delta h_+ + \Delta h_-} \\ b &= \frac{\Gamma'_+ - \Gamma'_-}{\Delta h_+ + \Delta h_-} \quad \text{where} \\ \Gamma'_\pm &= \frac{\Gamma_{i*+1\pm1/2} - \Gamma_{i*\pm1/2}}{\Delta h_\pm} \quad \text{and} \\ \Delta h_\pm &= (1/2)\left(h_{i*+3/2\pm1/2} - h_{i*-1/2\pm1/2}\right). \end{aligned}$$

- Check that extrapolation errors haven't made the PBLH(s) less than 300 m; if they have, set it/them to `ropp_MDFV` and set bit `PBLH_QC_too_low` of the QC flag. (If two PBLHs are detected, only set this flag if both are too low.)
- Check that extrapolation errors haven't made the PBLH(s) greater than 5000 m; if they have, set it/them to `ropp_MDFV` ansd set bit `PBLH_QC_too_high` of the QC flag. (If two PBLHs are detected, only set this flag if both are too high.)
- Calculate the 'geographical region' of the profile according to its location and set bit `PBLH_QC_land` or `PBLH_QC_coast` etc of the QC flag accordingly.
- Copy the QC flag to `ro_data%lev2c%pblh_bangle_flag`. Set `ro_data%lev2c%pblh_bangle` equal to the firs (or only) diagnosed PBLH. Set `ro_data%lev2c%pbla_bangle` equal to the bending angle at the first diagnosed PBLH. Set `ro_data%lev2c%pblh_bangle2` and `ro_data%lev2c%pbla_bangle2` to be the corresponding fields at the second PBLH, if one was diagnosed (otherwise leave as `ropp_MDFV`).

4.2 Refractivity

The calculation of refractivity-based boundary layer height, PBLH$_N$ follows a very similar pattern to that of the bending-angle-based PBLH described in Sec 4.1, except that:

- The underlying definition is now of course

$$PBLH_N = \arg\min\left(\partial N/\partial h\right); \tag{4.4}$$

- The vertical coordinate h is the given refractivity altitude minus the height of the surface geopotential;
- Checks on the existence of radius of curvature or undulation are neither needed nor made;
- The first and second PBLHs are stored in `ro_data%lev2c%pblh_refrac` and `ro_data%lev2c%pblh_refrac2`.

- The refractivities at the first and second PBLH are stored in `ro_data%lev2c%pbln_refrac` and `ro_data%lev2c%pbln_refrac2`.

Full details can be found in `ropp_apps/pblh/ropp_apps_pblh_refrac.f90`.

4.3 Dry temperature

ROPP follows the advice of Xie (2014) and defines the first (and second) dry-temperature-based boundary layer heights, PBLH$_{T\mathrm{dry}}$, as the heights of *maxima* of the vertical derivative of the dry temperature:

$$PBLH_{T\mathrm{dry}} = \arg\max\left(\partial T_{\mathrm{dry}}/\partial h\right) \tag{4.5}$$

in which T_{dry} is the dry temperature and h is the vertical geometric distance above the Earth's surface. As for refractivity (see Sec 4.2), the vertical coordinate h is the given refractivity altitude minus the height of the surface geopotential.

The basic algorithm used by `ropp_apps/pblh/ropp_apps_pblh_tdry.f90` is very similar to that of `ropp_apps/pblh/ropp_apps_pblh_bangle.f90`, save that we are looking for maxima in $\partial T_{\mathrm{dry}}/\partial h$ rather than minima in $\partial\alpha/\partial h$. The only material difference is the generation of dry temperatures, if necessary, by subroutine `ropp_apps_calc_tdry` in `ropp_apps/common/ropp_apps_utils.f90`. These are estimated by downwards integration of the dry temperature equation (see Sec 5.6.3 of ROPP_PP User Guide (2021) for further details)

$$\mathrm{d}\log p/\mathrm{d}z = -\left(g_{\mathrm{wmo}}/R_{\mathrm{dry}}\kappa_1\right)N\exp(-\log p) = -CN\exp(-\log p) \tag{4.6}$$

where z is the *geopotential* height, so that we can use constant $g_{\mathrm{wmo}} = 9.80665$ ms^{-2}, $R_{\mathrm{dry}} = 287.05$ J K^{-1} kg^{-1} and $\kappa_1 = 0.776$ N-unit K Pa^{-1} and therefore $C = g_{\mathrm{wmo}}/R_{\mathrm{dry}}\kappa_1 = 0.044025$ Pa m^{-1} N-unit^{-1}. A simple climatological estimate of $\mathrm{d}T_{\mathrm{dry}}/\mathrm{d}z$ near the top of the profile, at $i = n-1$, allows Eqn (4.6) to be integrated to the surface by second order explicit midpoint differencing, thus:

$$\begin{aligned}
\log p_{i-1/2} &= \log p_i - (1/2)C(z_{i-1}-z_i)N_i\exp(-\log p_i), \quad \text{followed by} && (4.7)\\
\log p_{i-1} &= \log p_i - C(z_{i-1}-z_i)\sqrt{N_iN_{i-1}}\exp(-\log p_{i-1/2}), \quad \text{subject to} && (4.8)\\
p_{n-1} &= -N_{n-1}\frac{C+\kappa_1^{-1}(\mathrm{d}T_{\mathrm{dry}}/\mathrm{d}z)_{n-1}}{(\mathrm{d}\log N/\mathrm{d}z)_{n-1}} \approx -N_{n-1}\frac{C+\kappa_1^{-1}(\mathrm{d}T_{\mathrm{dry}}/\mathrm{d}z)_{n-1}}{\log(N_n/N_{n-2})/(z_n-z_{n-2})}, && (4.9)
\end{aligned}$$

from which T_{dry} can be found using

$$T_{\mathrm{dry}} = \kappa_1\exp(\log p)/N. \tag{4.10}$$

(The geometric mean estimate of $N_{i-1/2}$, $\sqrt{N_iN_{i-1}}$, is preferred in Eqn (4.8) because the refractivity is likely to vary more exponentially than linearly.)

After integrating Eqns (4.7) and (4.8) from $i = n-1$ to $i = 2$, the pressure (and thence dry temperature) at the top of the profile, $i = n$, is estimated from

$$\log p_n = \log p_{n-2} - C(z_n - z_{n-2})N_{n-1}\exp(-\log p_{n-1}). \tag{4.11}$$

This `ropp_apps` dry temperature calculation is slightly different (simpler) from that in the `ropp_pp` and `ropp_fm` modules, but it is good enough for the diagnostic purposes here.

The dry-temperature-based PBLH(s) are stored as elements `pblh_tdry` and `pblh_tdry2` of the `ro_data%lev2c` structure, while the corresponding dry temperatures are stored in `pblt_tdry` and `pblt_tdry2` in the same structure.

4.4 Temperature

The calculation of temperature-based boundary layer height, PBLH$_T$, in ROPP follows that of the dry-temperature-based PBLH described in Sec 4.3. The PBLH is again defined by the position of the maximum of $\partial T/\partial h$. Unlike the previous calculation, however, there is no facility to generate the data from other information: if the (level 2b) temperatures T are missing from the input file, PBLH$_T$ cannot be calculated.

The temperature-based PBLH(s) are stored as elements `pblh_temp` and `pblh_temp2` of the `ro_data%lev2c` structure, while the corresponding temperatures are stored in `pblt_temp` and `pblt_temp2` in the same structure.

4.5 Specific humidity

The calculation of specific-humidity-based boundary layer height, PBLH$_q$, in ROPP mirrors that of the temperature-based PBLH described in Sec 4.4, except that the PBLH is defined by the position of the *minimum* of $\partial q/\partial h$. Again, if the (level 2b) specific humidities q are missing from the input file, PBLH$_q$ cannot be calculated.

The specific-humidity-based PBLH(s) are stored as elements `pblh_shum` and `pblh_shum2` of the `ro_data%lev2c` structure, while the corresponding specific humidities are stored in `pblq_shum` and `pblq_shum2` in the same structure.

4.6 Relative humidity

The calculation of relative-humidity-based boundary layer height, PBLH$_\rho$ in ROPP follows that of the specific-humidity-based PBLH described in Sec 4.5: PBLH$_\rho$ is defined by the position of the minimum of $\partial\rho/\partial h$. Relative humidities ρ, however, are not part of the standard `ROprof` data structure, so they need to be calculated from the (level 2b) temperature, pressure and specific humidity, which must therefore be available for this diagnostic to be calculable. For this, `ropp_apps/pblh/ropp_apps_pblh_rhum.f90` uses the formulas in the ECMWF IFS documentation (ECMWF, 2018), namely:

$$\rho = \frac{qp}{e_s(T)\,(\varepsilon + q(1-\varepsilon))}\,100\% \qquad (4.12)$$

where p is the pressure, q is the specific humidity, $\varepsilon \approx 0.622$ is the molecular weight of water divided by that of dry air, and the saturated vapour pressure $e_s(T)$ is given by

$$e_s(T) = \alpha e_{sw}(T) + (1-\alpha)e_{si}(T) \tag{4.13}$$

where the 'quadratic ramp' α is defined as a function of temperature T in K by

$$\alpha = \begin{cases} 0, \quad T < 250.16 \text{ K} \\ \left((T-250.16)/(273.16-250.16)\right)^2, \quad 250.16 \text{ K} \leq T \leq 273.16 \text{ K} \\ 1, \quad 273.16 \text{ K} < T \end{cases} \tag{4.14}$$

and the saturated vapour pressures over pure water, e_{sw}, and pure ice, e_{si}, are functions of temperature which are given in function `esat` in `ropp_apps/pblh/ropp_apps_pblh_rhum.f90`.

The relative-humidity-based PBLH(s) are stored as elements `pblh_rhum` and `pblh_rhum2` of the `ro_data%lev2c` structure, while the corresponding relative humidities are stored in `pblq_rhum` and `pblq_rhum2`.

4.7 Calculating the PBLH diagnostics

For `ropp_apps_pblh_tool` to (try to) calculate all six PBLHs, the user need simply call

```
ropp_apps_pblh_tool <inputROPPfile> -o <outputROPPfile>
```

The user may instead choose to calculate just the bending angle-, refractivity-, dry temperature-, temperature-, specific humidity- or relative humidity-based PBLH by calling `ropp_apps_pblh_tool` with the '-b', '-n', '-y' '-t', '-q' or '-r' flags respectively. (Two or more of these can be requested at the same time; specifying none is equivalent to requesting all six of them.) Invoking the '-d' option causes more diagnostics to be written to standard output, as well as the dependent variable, independent variable and vertical derivative profiles to be added to the output netCDF file.

4.8 Examples of the PBLH diagnostics

Figure 4.1 shows all six pairs of PBL heights plotted for a COSMIC occultation (vertical resolution 150–200 m) and a colocated 70-level (vertical resolution 50–100 m) Met Office background profile from 1 April 2013, for a point in a subtropical stratocumulus region. (This the example dataset provided with the ROPP distribution.) In this carefully chosen case the principal PBLHs estimated by all six measures are reasonably close: 2.2 ± 0.2 km. This is not too far off von Engeln and Teixeira's ERA-derived, relative-humidity-based PBLH climatology (von Engeln and Teixeira, 2013).

Rarely is the conclusion so clear cut. Even in this special case, the 'observationally' defined PBLH measures (top row of Fig 4.1) suggest a second PBLH at around 3.1–3.2 km, which the model-defined PBLHs on the bottom row entirely fail to capture. Indeed, they suggest a much lower second PBLH at 0.6–0.7 km — to which, in turn, the observational measures are oblivious, perhaps because of the poorer vertical resolution.

A fuller summary of some preliminary results of using the PBLH schemes implemented in ROPP can be found in (ROM SAF, 2016). It concludes that, overall, the 'best' observationally based PBLH is the one that is based on refractivity, PBLH$_N$, while the 'best' model-based PBLH is the one that is based on specific humdity, PBLH$_q$. Further investigations into the variability, reliability, accuracy and utility of the ROPP PBLH diagnostics are planned, but even if they were complete this would not be the place to discuss the results in detail. The intention here is that the PBLH diagnostics encoded in ROPP are sufficiently robust and scientifically sound to allow such investigations to begin.

References

ECMWF, IFS documentation - Cy45r1. Part IV: Physical processes, IFS documentation, ECMWF, https://www.ecmwf.int/en/elibrary/18714-part-iv-physical-processes, 2018.

ROM SAF, Planetary Boundary Layer Height diagnostics in ROPP, SAF/ROM/METO/REP/RSR/024, 2016.

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

von Engeln, A. and Teixeira, J., A ducting climatology derived from the European Centre for Medium-Range Weather Forecasts global analysis fields, *Journal of Geophysical Research*, *109*, doi:10.1029/2003JD004 380, 2004.

von Engeln, A. and Teixeira, J., A Planetary Boundary Layer Height Climatology Derived from the ECMWF Reanalysis Data, *Journal of Climate*, *26*, 6575–6590, 2013.

Xie, F., Visiting Scientist Report 21: Investigation of methods for the determination of the PBL height from RO observations using ECMWF reanalysis data, SAF/ROM/DMI/REP/VS21/001, Version 1.0, 2014.

PBLH parameter definitions

Element	Type(kind)	Definition	Values
pblh_bangle	Real(4)	PBLH height (m)	`ropp_MDFV` initially/incalculable; otherwise a 'valid' PBLH.
pbla_bangle	Real(4)	PBLH bending angle (rad)	`ropp_MDFV` initially/incalculable; otherwise a 'valid' PBLA.
pblh_bangle2	Real(4)	Second PBLH height (m)	`ropp_MDFV` initially/incalculable; otherwise a 'valid' PBLH.
pbla_bangle2	Real(4)	Second PBLH bending angle (rad)	`ropp_MDFV` initially/incalculable; otherwise a 'valid' PBLA.
pblh_bangle_flag	Int(2)	Bending angle PBLH QC flag	`ropp_MIFV` initially/incalculable; 0 – 2^{14}-1 otherwise.
pblh_refrac	Real(4)	PBLH height (m)	`ropp_MDFV` initially/incalculable; otherwise a 'valid' PBLH.
pbln_refrac	Real(4)	PBLH refractivity (N-unit)	`ropp_MDFV` initially/incalculable; otherwise a 'valid' PBLN.
pblh_refrac2	Real(4)	Second PBLH height (m)	`ropp_MDFV` initially/incalculable; otherwise a 'valid' PBLH.
pbln_refrac2	Real(4)	Second PBLH refractivity (N-unit)	`ropp_MDFV` initially/incalculable; otherwise a 'valid' PBLN.
pblh_refrac_flag	Int(2)	Refractivity PBLH QC flag	`ropp_MIFV` initially/incalculable; 0 – 2^{14}-1 otherwise.
pblh_tdry	Real(4)	PBLH height (m)	`ropp_MDFV` initially/incalculable; otherwise a 'valid' PBLH.
pblt_tdry	Real(4)	PBLH dry temperature (K)	`ropp_MDFV` initially/incalculable; otherwise a 'valid' PBLT.
pblh_tdry2	Real(4)	Second PBLH height (m)	`ropp_MDFV` initially/incalculable; otherwise a 'valid' PBLH.
pblt_tdry2	Real(4)	Second PBLH dry temperature (K)	`ropp_MDFV` initially/incalculable; otherwise a 'valid' PBLT.
pblh_tdry_flag	Int(2)	Dry temperature PBLH QC flag	`ropp_MIFV` initially/incalculable; 0 – 2^{14}-1 otherwise.
pblh_temp	Real(4)	PBLH height (m)	`ropp_MDFV` initially/incalculable; otherwise a 'valid' PBLH.
pblt_temp	Real(4)	PBLH temperature (K)	`ropp_MDFV` initially/incalculable; otherwise a 'valid' PBLT.
pblh_temp2	Real(4)	Second PBLH height (m)	`ropp_MDFV` initially/incalculable; otherwise a 'valid' PBLH.
pblt_temp2	Real(4)	Second PBLH temperature (K)	`ropp_MDFV` initially/incalculable; otherwise a 'valid' PBLT.
pblh_temp_flag	Int(2)	Temperature PBLH QC flag	`ropp_MIFV` initially/incalculable; 0 – 2^{14}-1 otherwise.
pblh_shum	Real(4)	PBLH height (m)	`ropp_MDFV` initially/incalculable; otherwise a 'valid' PBLH.
pblq_shum	Real(4)	PBLH specific humidity (g/kg)	`ropp_MDFV` initially/incalculable; otherwise a 'valid' PBLQ.
pblh_shum2	Real(4)	Second PBLH height (m)	`ropp_MDFV` initially/incalculable; otherwise a 'valid' PBLH.
pblq_shum2	Real(4)	Second PBLH specific humidity (g/kg)	`ropp_MDFV` initially/incalculable; otherwise a 'valid' PBLQ.
pblh_shum_flag	Int(2)	Specific humidity PBLH QC flag	`ropp_MIFV` initially/incalculable; 0 – 2^{14}-1 otherwise.
pblh_rhum	Real(4)	PBLH height (m)	`ropp_MDFV` initially/incalculable; otherwise a 'valid' PBLH.
pblr_rhum	Real(4)	PBLH relative humidity (%)	`ropp_MDFV` initially/incalculable; otherwise a 'valid' PBLR.
pblh_rhum2	Real(4)	Second PBLH height (m)	`ropp_MDFV` initially/incalculable; otherwise a 'valid' PBLH.
pblr_rhum2	Real(4)	Second PBLH relative humidity (%)	`ropp_MDFV` initially/incalculable; otherwise a 'valid' PBLR.
pblh_rhum_flag	Int(2)	Relative humidity PBLH QC flag	`ropp_MIFV` initially/incalculable; 0 – 2^{14}-1 otherwise.

Table 4.2: Elements of `ro_data%Lev2c` substructure relating to PBLH

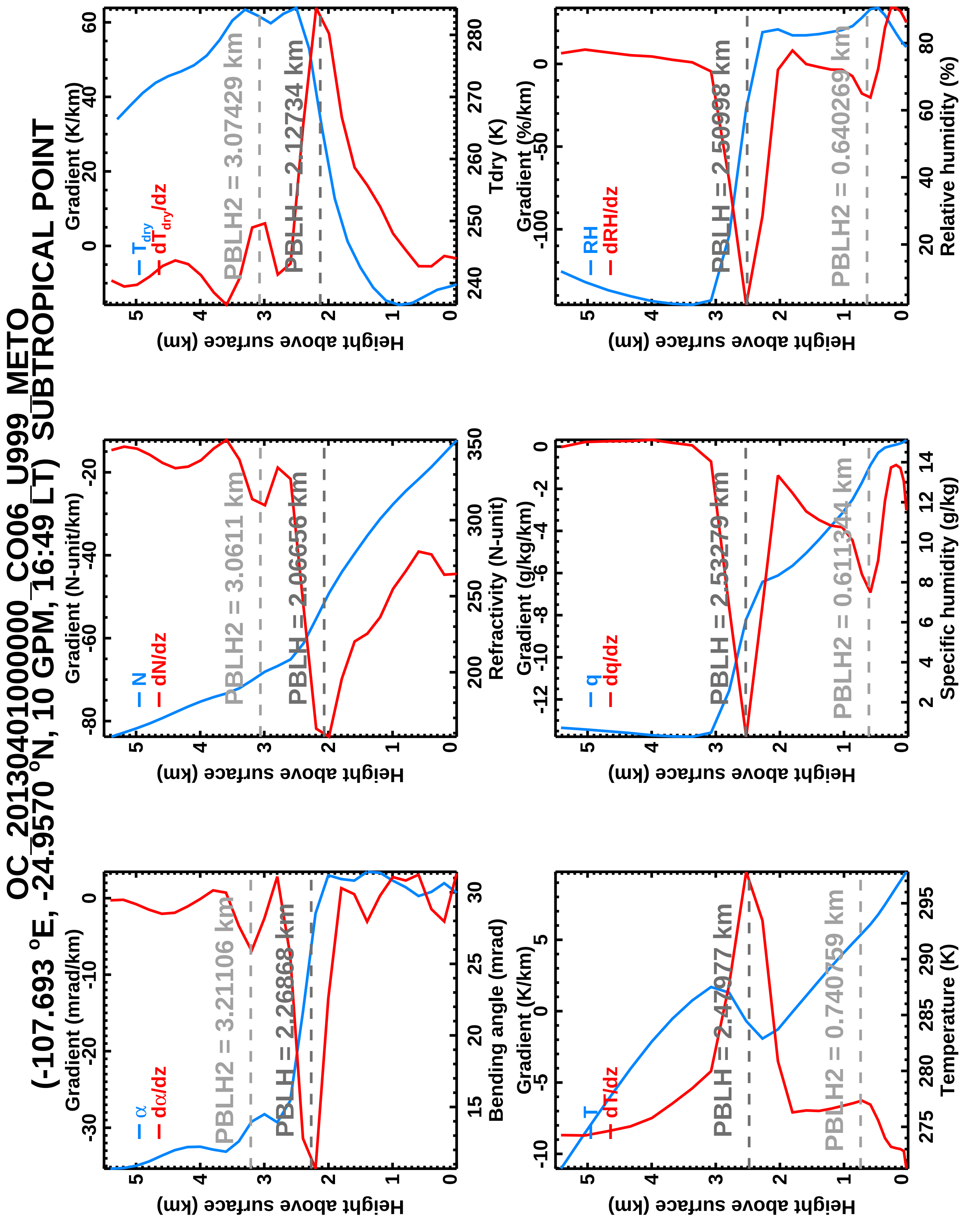

Figure 4.1: Example PBL heights, 1 April 2013, South Pacific stratocumulus region. Top row: COSMIC bending-angle-, refractivity- and dry-temperature-based PBLHs; bottom row: Met Office temperature, specific-humidity and relative-humidity-based PBLHs. Basic fields in blue (lower x-axis); vertical gradients in red (upper x-axis). PBLHs from the first and second strongest minima (or maxima for T_{dry} and T) are both shown.

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 netcdff <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`

---

[1] See `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 be built with

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

and the ECMWF ecCodes library would be 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 tool[2] 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):

---

[2] See 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 netcdff mobufr/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`, `netcdff`, `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 (-lnetcdff)
```

will be required. (Since netCDF-4.1.1, the netCDF C and Fortran routines have been split, with the latter held in `libnetcdff.a`. Hence, if compiling Fortran routines against a recent version of netCDF, `-lnetcdff` 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 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 `.cdl`) 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_1dvar`, `ropp_apps` and `ropp_fm` also allow the user to add sets of pre-defined variables to the `ROprof` 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 `ROprof` 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_apps` program files

The `ropp_apps` module provides tools to generate tropopause heights, and planetary boundary layer heights, from profiles of RO data variables including bending angle, refractivity, dry temperature, or (wet) temperature.

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_apps_tph_tool.f90 (requires `ropp_io`)
  ropp_apps_pblh_tool.f90 (requires `ropp_io`)

tests/

  test_apps_tph.sh (requires `ropp_io`)
  test_apps_pblh.sh (requires `ropp_io`)
  ropp_apps_compare.f90 (requires `ropp_io`, `ropp_utils`)
  ropp_apps_summary.f90 (requires `ropp_utils`)

- Integrated code

```
pblh/
     ropp_apps_pblh_bangle.f90 (requires ropp_io)
     ropp_apps_pblh_refrac.f90 (requires ropp_io)
     ropp_apps_pblh_tdry.f90 (requires ropp_io)
     ropp_apps_pblh_temp.f90 (requires ropp_io)
     ropp_apps_pblh_shum.f90 (requires ropp_io)
     ropp_apps_pblh_rhum.f90 (requires ropp_io)
     ropp_apps_pblh_region.f90
```

```
tph/
     ropp_apps_tph_bangle.f90 (requires ropp_io)
     ropp_apps_tph_refrac.f90 (requires ropp_io)
     ropp_apps_tph_tdry.f90 (requires ropp_io)
     ropp_apps_tph_temp.f90 (requires ropp_io)
     ropp_apps_cov_transform.f90

common/
     ropp_apps.f90
     ropp_apps_constants.f90
     ropp_apps_types.f90
     ropp_apps_utils.f90
     ropp_apps_version.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 `RO_prof` structure by `ropp_io_addvar`, is described in Table C.1.

ROprof (Additional variables requested by call to `ropp_io_addvar`, throughout ROPP)

Structure element	Description
`...%vlist%VlistD0d%name`	Name of 1st 0D extra variable
`...%vlist%VlistD0d%long_name`	Long name of 1st 0D extra variable
`...%vlist%VlistD0d%units`	Units of 1st 0D extra variable
`...%vlist%VlistD0d%range`	Range of 1st 0D extra variable
`...%vlist%VlistD0d%DATA`	Value of 1st 0D extra variable
`...%vlist%VlistD0d%next%name (etc)`	Name (etc) of 2nd 0D extra variable
`...%vlist%VlistD0d%next%next%name (etc)`	Name (etc) of 3rd 0D extra variable
`...%vlist%VlistD1d%name`	Name of 1st 1D extra variable
`...%vlist%VlistD1d%long_name`	Long name of 1st 1D extra variable
`...%vlist%VlistD1d%units`	Units of 1st 1D extra variable
`...%vlist%VlistD1d%range`	Range of 1st 1D extra variable
`...%vlist%VlistD1d%DATA`	Value of 1st 1D extra variable
`...%vlist%VlistD1d%next%name (etc)`	Name (etc) of 2nd 1D extra variable
`...%vlist%VlistD1d%next%next%name (etc)`	Name (etc) of 3rd 1D extra variable
`...%vlist%VlistD2d%name`	Name of 1st 2D extra variable
`...%vlist%VlistD2d%long_name`	Long name of 1st 2D extra variable
`...%vlist%VlistD2d%units`	Units of 1st 2D extra variable
`...%vlist%VlistD2d%range`	Range of 1st 2D extra variable
`...%vlist%VlistD2d%DATA`	Value of 1st 2D extra variable
`...%vlist%VlistD2d%next%name (etc)`	Name (etc) of 2nd 2D extra variable
`...%vlist%VlistD2d%next%next%name (etc)`	Name (etc) of 3rd 2D extra variable

Table C.1: Additional elements of `ROprof` 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: $\mathtt{MAX[err\_neut}^{1/2}/\alpha_N\mathtt{]}\times 100\%$
`...%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 ROprof structure if the `ropp_fm` tool `ropp_fm_bg2ro_1d` is called without the '`-f`' option, are described in Table C.3.

ROprof (Absence of '`-f`' option in call to `ropp_fm_bg2ro_1d`, in `ropp_fm`)	
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 `ROprof` 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
6th 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 Engeln	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{i/f}2ec{i/f} 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.