ROM SAF CDOP-2

Radio Occultation Processing Package (ROPP) ROPP-9 Beta Test Report

Version 9.0

28 February 2017

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/TR/ROPP/002

DOCUMENT AUTHOR TABLE

	Author(s)	Function	Date	Comment
Prepared by:	Ian Culverwell	ROM SAF Project Member	28/02/2017	
Reviewed by:	Chris Burrows	ROM SAF Project Member	28/02/2017	
Approved by:	Kent B. Lauritsen	ROM SAF Project Manager	28/02/2017	

DOCUMENT CHANGE RECORD

Issue / Revision	Date	By	Description
4.0	4/11/2009	HL	Release version for ROPP-4 (v4.0)
5.0	28/6/2011	IC	Release version for ROPP-5 (v5.0)
6.0	7/12/2011	IC	Release version for ROPP-6 (v6.0)
7.0	18/7/2013	IC	Release version for ROPP-7 (v7.0)
8.0	31/12/2014	IC	Release version for ROPP-8 (v8.0)
9.0	28/02/2017	IC	Release version for ROPP-9 (v9.0)

List of Contents

Executive Summary

This document records the results of the beta testing of the Radio Occultation Processing Package (ROPP) prior to its ninth full release, known as ROPP-9 (v9.0), or hereafter as **ROPP-9.**

The majority of the testing of ROPP-9 has been done in-house as part of the standard development and testing procedures under the 'Test Folder' system used for the first release and updates (ROPP-1 v1.0, v1.1, v1.2) and subsequent releases (ROPP-2 v2.0 through to ROPP-8 v8.0). The tests performed under this procedure and the results from them are described in the Test Plan and Test Folder Report, respectively. They are briefly summarised here. The focus of this report, however, is the external (ie non-Met Office) testing of the beta (pre-release) version of ROPP-9.

ROPP9.0 was built from ROPP8.0 in two stages. The intermediate stage, ROPP8.1, was a minor, private release which differed from ROPP8.0 solely by the inclusion of code to diagnose Planetary Boundary Layer Height (PBLH). The hope is that eventually these diagnostics will be useful for climate monitoring and model development, process studies, and even, perhaps, assimilating into NWP models.

Since ROPP8.1 was a private release, for sole use by the ROM SAF Project Team, no formal beta testing was undertaken.

The other major new component of ROPP9.0 is a Wave Optics Propagation Tool (WOPT). This tool simulates the excess phase delays observed at the LEO that are induced by a spherically symmetrical neutral atmosphere, by directly calculating the scattering of radio waves propagating through a given (discretised) refractivity field. The hope is that this will form a testbed for various sensitivity studies connected with the essential physics of a radio occultation measurement.

ROPP9.0 also includes numerous smaller upgrades, additions and bug-fixes. These are fully documented in the ROPP Release Notes [RD.9]. Chief among these developments are a revised statistical optimisation scheme ('BAROCLIM(3)') in the PP module, a new method of specifying model background errors in the 1DVAR module ('RSFC'), numerous changes to the I/O routines – in particular, to the EUMETSAT data reader – and a generalisation and improvement to the forward model interface.

The formal ROPP9.0 beta review has been undertaken by an acknowledged expert in the field of boundary layer height detection with RO data. In addition, some experienced ROPP users downloaded the beta release and carried out their own testing. And, as usual, the beta release was passed through the ROPP testing folder. This report records and comments on the results of all these tests.

1. Introduction

1.1 Purpose of document

This document reports on the beta testing of the Radio Occultation Processing Package (ROPP) prior to its ninth full release, known as ROPP-9 (v9.0), or hereafter as **ROPP-9.**

The majority of the testing of ROPP-9 has been done in-house as part of the standard development and testing procedures under the 'Test Folder' system used for the first release and updates (ROPP-1 v1.0, v1.1, v1.2) and subsequent releases (ROPP-2 v2.0 through to ROPP-8 v8.0). The tests performed under this procedure and the results from them are described in the Test Plan and Test Folder Report, respectively. They are briefly summarised here. The focus of this report, however, is the external (ie non-Met Office) testing of the beta (pre-release) version of ROPP-9.

ROPP9.0 was built from ROPP8.0 in two stages. The intermediate stage, ROPP8.1, was a minor, private release which differed from ROPP8.0 solely by the inclusion of code to diagnose Planetary Boundary Layer Height (PBLH). The hope is that eventually these diagnostics will be useful for climate monitoring and model development, process studies, and even, perhaps, assimilating into NWP models.

Since ROPP8.1 was a private release, for sole use by the ROM SAF Project Team, no formal beta testing was undertaken.

The other major new component of ROPP9.0 is a Wave Optics Propagation Tool (WOPT). This tool simulates the excess phase delays observed at the LEO that are induced by a spherically symmetrical neutral atmosphere, by directly calculating the scattering of radio waves propagating through a given (discretised) refractivity field. The hope is that this will form a testbed for various sensitivity studies connected with the essential physics of a radio occultation measurement.

ROPP9.0 also includes numerous smaller upgrades, additions and bug-fixes. These are fully documented in the ROPP Release Notes [RD.9]. Chief among these developments are a revised statistical optimisation scheme ('BAROCLIM(3)') in the PP module, a new method of specifying model background errors in the 1DVAR module ('RSFC'), numerous changes to the I/O routines – in particular, to the EUMETSAT data reader – and a generalisation and improvement to the forward model interface.

The formal ROPP9.0 beta review has been undertaken by an acknowledged expert in the field of boundary layer height detection with RO data. In addition, some experienced ROPP users downloaded the beta release and carried out their own testing. And, as usual, the beta release was passed through the ROPP testing folder. This report records and comments on the results of all these tests.

The readership of this document is the ROM SAF Project Team members (including the ROPP Development Team) and DRI Reviewers and should be read in conjunction with other parts of the ROPP-9 Project Documentation set.

1.2 What is ROPP?

Objective: *To provide Users with a comprehensive software package, containing all necessary functionality to pre-process RO data from CGS Level 1b files or ROM SAF Level 2 files, plus RO-specific components to assist with the assimilation of these data in NWP systems.*

The ROPP is a package of software (as source code) and supporting build and test scripts, data files and documentation, which will aid users wishing to process, quality-control and assimilate radio occultation data into their NWP models. Whilst aimed at the GRAS instrument on Metop-A and B, as far as is practicable, the software is generic, in that it can handle any other GPS–LEO configuration radio occultation mission (GRAS, CHAMP, GRACE, SAC-C, COSMIC, TerraSAR-X, TanDEM-X, C/NOFS, ROSA, PAZ, etc). The LEO–LEO configuration will not be supported in the current ROM SAF CDOP, but in principle such support could be included at a future time if any mission with this configuration is likely to be launched.

The ROPP concept, overall development strategy and overview of content is described in [RD.2]. For details of the package, the ROPP User Guide [RD.3] should be consulted. [RD.7] documents the in-house validation of the software, results from which can be browsed in a 'Test Folder'.

1.3 Applicable & Reference documents

1.3.1 Applicable documents

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

[AD.1] Proposal for Continuous Development and Operations Phase II (ROM SAF CDOP-2) as endorsed by Council 29 June 2011.
SAF/GRAS/DMI/MGT/CDOP2/001

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

[AD.3] ROPP beta test licence.
SAF/ROM/METO/LIC/ROPP/001

1.3.2 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).
Ref: SAF/ROM/METO/ADD/ROPP/001

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

[RD.3] User Guides for the ROPP IO, PP, APPS, FM and 1DVAR modules
Ref: SAF/ROM/METO/UG/ROPP/002,004,005,006 and 007 resp.

[RD.4] ROPP Test Folder Documentation
Ref: SAF/ROM/METO/TD/ROPP/001

[RD.5] Beta Test Report (v1.0 – v8.0)
Ref: SAF/GRAS/METO/TR/ROPP/002

[RD.6] Development procedures for software deliverables
Ref: NWPSAF-MO-SW-002

[RD.7] ROPP Test Plan
Ref: SAF/ROM/METO/TP/ROPP/001

[RD.8] ROPP Test Folder Report
Ref: SAF/ROM/METO/TR/ROPP/001

[RD.9] ROPP Release Notes
Ref: SAF/ROM/METO/SRN/ROPP/001

[RD.10] ROM SAF Visiting Scientist Proposal 30
Ref: SAF/ROM/DMI/MGT/VS30/001

[RD.11] ROM SAF Visiting Scientist Report 30
Ref: SAF/ROM/DMI/REP/VS/30

[RD.12] ROM SAF Report 24, The calculation of planetary boundary layer heights in ROPP
Ref: SAF/ROM/METO/REP/RSR/024

[RD.13] Ao, C. O., D. E. Waliser, S. K. Chan, J.-L. Li, B. Tian, F. Xie, and A. J. Mannucci: Planetary boundary layer depths from GPS radio occultation profiles, J. Geophys. Res., 117, D16117, doi:10.1029/2012JD017598, 2012.

1.4 Acronyms, Abbreviations & Initialisms

ADD	Architectural Design Document
BLAS	Basic Linear Algebra Subprograms
BUFR	Binary Universal Form for the Representation of meteorological data (WMO)
CDOP	Continuous Development and Operational Phase (SAFs)
CGS	Core Ground Segment (EUMETSAT)
CHAMP	CHallenging Mini-satellite Payload (Germany)
CLIMAP	Climate and Environment Monitoring with GPS-based Atmospheric Profiling (EU)
COSMIC	Constellation Observing System for Meteorology Ionosphere and Climate (USA/Taiwan)
DMI	Danish Meteorological Institute (ROM SAF Host)
DRI	Delivery Readiness Inspection
EC	Environment Canada
ECMWF	European Centre for Medium-range Weather Forecasts
ESA	European Space Agency
EU	European Union
EUMETcast	EUMETSAT NRT dissemination service via commercial digital video broadcast technology
EUMETSAT	EUropean organisation for the exploitation of METeorological SATellites (Darmstadt, Germany)
FM94	WMO Form no. 94 (i.e. BUFR)
Galileo	European GNSS system from 2008 (EU/ESA)
GFZ	GeoForschungsZentrum (Potsdam, Germany)
GLONASS	Globalnaya Navigatsionnaya Sputnikovaya Sistema (Russia)
GNSS	Global Navigation Satellite System (generic GPS/GLONASS/Galileo)
GNOS	GNSS Occultation Sounder (China)
GPS	Global Positioning System (USA)
GRACE	Gravity Recovery and Climate Experiment (Germany/US)
GRAS	GNSS Receiver for Atmospheric Sounding (METOP)
GTS	Global Telecommunications System (WMO)

HP-UX	Unix operating system for Hewlett Packard workstations
IEEC	Institut d'Estudis Espacials de Catalunya
LAPACK	Linear Algebra PACKage
MetDB	Meteorological Data Base (Met Office)
MetO	Met Office (of the UK)
METOP	METeorological OPerational satellite (EUMETSAT)
MS-DOS	Microsoft Disk Operating System ('Command Line' application under the Windows O/S)
netCDF	network Common Data Form (Unidata)
NMS	National Meteorological Service
NWP	Numerical Weather Prediction
NRT	Near-Real Time
OS (O/S)	Operating System
PAZ	Spanish Earth Observation Satellite, carrying a Radio Occultation Sounder
PCD	Product Confidence Data
PES	Re-Existing Software
PFS	Product Format Specification (Level 1b data from GCS)
POD	Precision Orbit Determination
RMDCN	Regional Meteorological Data Communications Network (Europe)
RO	Radio Occultation
ROM SAF	Radio Occultation Meteorology Satellite Application Facility (EUMETSAT)
ROPP	Radio Occultation Processing Package
SAC-C	Satelite de Applicaciones Cientificas – C (Argentina)
SAF	Satellite Application Facility (EUMETSAT)
SG	Steering Group
SNR	Signal to Noise Ratio
TanDEM-X	German Earth Observation Satellite, carrying a Radio Occultation Sounder
TBC	To Be Confirmed
TBD	To Be Determined
TerraSAR-X	German Earth Observation Satellite, carrying a Radio Occultation Sounder
UCAR	University Center for Atmospheric Research (Boulder, CO, USA)
VAR	Variational (NWP data assimilation technique)
VS	Visiting Scientist (EUMETSAT SAF Scheme)
WMO	World Meteorological Organisation
WWW	World Weather Watch (WMO Programme)

2. Background

ROPP is developed under Work Package WP3420 of the ROM SAF CDOP-2 [AD.1]. The WP foresees the development and release of ROPP in a number of cycles as the required functionality increases, and in a priority order of user requirements for implementation.

The development, testing and release cycle for ROPP follows the guidelines for NWP SAF software deliverables [RD.6].

ROPP-1 (Version 1.0) was released in April 2007, following a similar beta test programme, with a formal review (DRI) and approval for release by the GRAS SAF (as it was then) Steering Group (SG) [RD.5, v1.0]. Update releases were delivered in April 2008 (v1.1) and September 2008 (v1.2). With the agreement of the SG, such minor updates are not required to be formally beta tested, reviewed under a DRI or specifically approved for release, though the extensive Test Folder pre-release testing process is the same as for major releases.

ROPP-2 (Version 2.0) was released in December 2008, following a beta test activity, with a formal review (DRI) and approval for release by the GRAS SAF Steering Group (SG). The beta test activity for ROPP-2 included an external, independent testing phase undertaken by Dr. Josep Aparicio under a Visiting Scientist contract [RD.5, v2.0].

ROPP-3 (Version 3.0) was released in July 2009, following a beta test activity, with a formal review (DRI) and approval for release by the GRAS SAF Steering Group (SG). The beta test activity for ROPP-3 included an external, independent testing phase undertaken by Dr Michael Borsche and Dr. Josep Aparicio under a Visiting Scientist contract [RD.5, v3.0].

ROPP-4 (Version 4.0) was released in January 2010, following a beta test activity, with a formal review (DRI) and approval for release by the GRAS SAF Steering Group (SG). The beta testing for ROPP-4 comprised an email questionnaire to a wide range of users, as well as more thorough testing of a beta release of the candidate code by Dr Luiz Sappucci and Dr Sean Healy [RD.5, v4.0].

ROPP-4 (Version 4.1) was released in July 2010. Being an updated version of ROPP-4 rather than a full release, it did not undergo formal beta testing.

ROPP-5 (Version 5.0) was released in July 2011, following a beta test activity, with a formal review and approval for release by the GRAS SAF Steering Group (SG). The beta testing for ROPP-5 comprised an external independent test of the feasibility of using ROPP in the operational environment CPTEC, carried out by Dr Luiz Sapucci at that institution. In addition, several experienced ROPP users were invited to download the beta release code and test it on local machines. This testing is documented in this report [RD.5, v5.0].

ROPP-6 (Version 6.0) was released in February 2012. Although a full release, it had (limited) extra functionality but numerous small revisions to the existing code base. Formal beta testing was restricted to one external user, Dr Johannes Fritzer (Wegener Center for Climate and Global Change, and Institute for Geophysics, Astrophysics and Meteorology, both at the University of Graz, Austria). In addition, several experienced ROPP users were invited to download the beta release code and test it on local machines. This testing is documented in this report [RD.5, v6.0].

ROPP-6 (Version 6.1) was released in April 2013. Being an updated version of ROPP-6 rather than a full release, it did not undergo formal beta testing. Its principal components were a reader of GRIB and ASCII background data, the calculation of dry temperature by the ropp_pp tools, as well as its incorporation within the general ROPP level 2a data structure, and a reader of 'grouped' netCDF-4 format level 1a data files from EUMETSAT.

ROPP-7 (Version 7.0) was a full release. Its main improvement is the inclusion of tropopause height diagnostics. Formal beta testing was undertaken by Dr Torsten Schmidt (Helmholtz Centre Potsdam, GFZ German Research Centre for Geosciences, Potsdam, Germany), an acknowledged expert in this field. In addition, some experienced ROPP users were invited to download the beta release code and test it on local machines. This testing is documented in this report.

ROPP-7 (Version 7.1) was an intermediate release. Being an updated version of ROPP-7 rather than a full release, it did not undergo formal beta testing. Its principal components were the facility to model L1 and L2 bending angles, improved vertical integration in the forward model, and improvements to the readers of 'grouped' netCDF-4 format level 1a data files from EUMETSAT.

ROPP-8 (Version 8.0) was a full release. Its main improvements over ROPP7.1 were developments to the background profile extractor from GRIB datasets; small generalisations to the 1D forward model; improvements to the automatic testing procedures; introduction of useful shell return codes; introduction of seasonally dependent observation error covariance matrices; and numerous bug-fixes and sundry other minor developments. Beta testing was carried out by expert users inside and outside the SAF.

ROPP-8 (Version 8.1) was a private release for the ROM SAF. Its only difference from ROPP8.0 was the inclusion of a tool to diagnose the planetary boundary layer height (PBLH) from profiles of bending angle, refractivity or dry temperature, or model background temperatures or humdities.

ROPP-9 (Version 9.0) is a full release. It contains the PBLH diagnostics of ROPP-8.1, together with a revised statistical optimisation scheme ('BAROCLIM(3)') in the PP module, a new method of specifying model background errors in the 1DVAR module ('RSFC'), numerous changes to the I/O routines – in particular, to the EUMETSAT data reader – and a generalisation and improvment to the forward model interface. All these changes are fully documented in the ROPP Release Notes [RD.9]. The formal beta review has been undertaken by an acknowledged expert in the field of boundary layer height detection with RO data. In addition, some experienced ROPP users downloaded the beta release and carried out their own testing. And, as usual, the beta release was passed through the ROPP testing folder. This report records the results of all this testing, and lists the responses of the ROPP Development Team to any issues raised.

3. Formal External Beta Testing

Dr. Feiqin Xie (Texas A&M University-Corpus Christi Texas, USA) was invited to undertake the formal external (ie, non-ROM SAF) beta testing of ROPP-9, as part of ROM SAF CDOP Visiting Scientist Proposal No. 30 [RD.10]. The objectives of this VS activity were:

1. *To install, assess and report on the ease of use, capability and usefulness of the ninth major release of the Radio Occultation Processing Package, ROPP-9.0; and*

2. *To assess the robustness and scientific integrity of the ROPP planetary boundary layer height diagnostic (PBLH).*

Dr Xie's report is published as [RD.11]. His conclusions are reproduced verbatim below. (His recommendations, and the responses to them, are discussed in detail later.)

Conclusions

In this report, the PBLH diagnostics based on three RO parameters (refractivity, bending angle and dry temperature) and three model parameters (specific humidity, relative humidity and temperature) during March-April-May of 2013 are generated from the ROPP PBLH diagnostic tool in the beta release of ROPP-9.0. Overall, the ROPP application tool is robust and easy to use, and the documentation is clear and very well-structured.

The seasonal climatology of six ROPP PBLHs is analyzed. The ROPP RO parameter based PBLH diagnostics are directly compared to the model parameter based PBLH. The ROPP PBLH diagnostics are also compared to the PBLH product generated by Xie at TAMU-CC that uses an independent but similar PBLH detection algorithm. Overall, the two PBLH products are very consistent except some positive bias in ROPP PBLH over the polar region, specially over the Antarctic and the Greenland. The difference over the polar region seems to be related to the special treatment of the PBLH detection algorithm in ROPP package in the presence of a shallow surface inversion (ROM-SAF Report 24, ROM SAF, 2016c). Clarification of such special treatment is needed.

The ROPP PBLH diagnostics are also compared to the CALIPSO PBLH that are derived based on the height of low level cloud or aerosol layers during the same three-month period. Overall, very similar pattern of global PBLH climatology is shown comparing the CALIPSO and the ROPP PBLH diagnostics, except the polar regions. Similar positive bias in ROPP PBLH over the Antarctic and the Greenland as seen in the comparison with the TAMU-CC data.

Generally, then, ROPP-9 was straightforward to build and run. As expected, Dr Xie had some comments on the scientific validity of the PBLH processing.

More specific comments from the reviewer, and the ROPP Development Team’s responses, follow. Only those points of Dr Xie’s report that require comment or action from the ROPP Development Team are reproduced here. Readers are invited to consult [RD.11] for supplementary information, especially the figures to which the reviewer refers.

3.1 General Assessment of the Beta Release of ROPP-9.0

3.1.1 Download of the Beta Release of ROPP-9.0

The beta release of ROPP-9.0 software and the associated documentation were successfully downloaded from ROM SAF website without encountering problems. Both the single file tarball and the individual files of the software package were separately downloaded and unpacked without any issue.

So far, so good.

3.1.2 Installation and General Utility of the Software Package

The beta release of ROPP-9.0 were successfully installed on the computer server at Texas A&M University – Corpus Christi (TAMU-CC) with the following configuration:

- CPU: Intel(R) Xeon(R) CPU E5-2440 0 @ 2.40GHz
- OS: Scientific Linux release 6.8 (Carbon)
- Compiler: gfortran

The step-by-step installation guideline described in ROPP Release Notes (Version 9.0) is well structured and is clear and easy to follow.

Ditto.

The only suggestion is to make it clear that the default shell environment is in *bash* while using "*export*" command. Alternatively, the "*setenv*" command for C-shell (*csh*) or T-shell (*tcsh*) environment could be added, such as follows:

- export ROPP_ROOT=/usr/local # bash
- setenv ROPP_ROOT /usr/local # csh/tcsh

The only places in ROPP that `export ROPP_ROOT` are build scripts like buildpack and build_deps, all of which are 'hashbanged' by `#!/bin/bash`, so they should be safe – if bash is available. And Sec 5.5 of the ROPP Release Notes says:

5.5 The buildpack script
Included in the main package (and in ropp_build) is a Bash shell script file buildpack. The user need not be using Bash as an interactive shell, but Bash needs to be installed (usually at /bin/bash) for the script to work; almost all POSIX-based systems should have Bash installed by default, and for many Linux distros, Bash is the default login shell. On the rare systems that do not have Bash installed, follow the main commands for the desired package installation section of this script.

Section 5.3 of the same document says:

5.3 Environment variables
The build system uses an environment variable ROPP_ROOT as the root path for all file installations. By default, this will be set to $HOME by the buildpack script. Pre-define this variable before installing any 3rd party packages or ROPP modules; for instance:
> export ROPP_ROOT=/usr/local
noting that the user must have appropriate file permissions to write to such a location.

Action 3.1.2.1: Add the reviewer's suggestion about using `setenv ROPP_ROOT /usr/local` if using csh or tsch to Sec 5.3 of the ROPP Release Notes.

Normally, key various ROPP environment variables, including `$ROPP_ROOT`, are defined by running the setroppenv script, which is included in the ROPP distribution.

Action 3.1.2.2: Add a note to setroppenv about using `setenv` for C-shell or T-shell users.

Action 3.1.2.3: Publicise setroppenv's existence in the ROPP Release Notes.

README.build also talks about exporting ROPP_ROOT, so

Action 3.1.2.4: Add a note about using `setenv` for C-shell or T-shell users to README.build.

Dr Xie goes on to say:

> The main package used for this report is the ROPP Application package (ROPP_APPS), which provides various ROPP APPS library. More specifically, the ROPP application tool are used to derive the diagnostics of the planetary boundary layer height. The test folder and the readme files are very helpful for the user to understand the usage of the shell script and Fortran programs as well as the input and output data format. The output data in NetCDF format is especially helpful and convenient for the users to carry out further data processing and analysis.

This is encouraging.

3.2 Assessment of the PBLH Diagnostics

3.2.1 Generate PBLH Diagnostics with ROPP

After some discussion of the generation of PBLH diagnostics for a dataset of about 190 000 radio occultation profiles between March and May 2013 which was provided by the ROM SAF, Dr Xie notes:

> No major issues were found during the processing of the large dataset. In a few cases, however, some missing parameters in the ROPP input data file could lead to failed PBLH diagnostics generation. It might be worth an effort to develop more robust algorithm that will allow to skip the processing of the missing parameter(s) in the input file.

The reviewer is alluding to a long standing occasional issue with processing ROPP multifiles, which is that a missing 'level' of data, corresponding to a dimension in the netCDF file, in the first profile of ingested data means that the output multifile is also missing that level/dimension of data. Hence if any later processing generates a profile with that level/dimension of data, there is nowhere to put it in the output multifile. A quick workaround for this rare occurence, which was suggested to – and successfully used by – the reviewer, is to reorder the input data so that the first profile is a 'full' one.

Action 3.2.1.1: Add low priority ROPP ticket #475 to find a better solution to this rare problem.

3.2.2 Evaluate the Scientific Integrity of ROPP PBLH Diagnostics

3.2.2.1 ROPP PBLH

The reviewer laments the small number of RO profiles that can be used for PBLH diagnosis:

> The percentage of deep penetrating RO profiles that reach the lowest 300 m above the local surface is mapped in Fig. 3. A very low percentage (only 10-20%) deep penetrating soundings are seen over the tropics. Over 50% and even higher are seen over mid-latitude and polar regions, respectively. Note again, most of the deep penetrating RO profiles are from COSMIC and TerraSAR-X.

Ain't that the truth.

A study of the mean and mean absolute deviation (MAD) led the reviewer to comment that

The distinct dipole PBLH structure are shown in all six PBLH diagnostics, which features a transition from a shallow PBLH (~1 km) near the stratocumulus regime over the subtropical eastern ocean near the western coast of the continent, to a much deeper PBLH (~2 km) off shore into the trade cumulus regime (Guo et al., 2011, Xie et al., 2012). It is worth noting that large difference are shown among six PBLH diagnostics especially over the tropics and the polar regions. For example, the $PBLH_{\alpha}$ and $PBLH_{Tdry}$ show deeper PBLH over tropics as comparing to all other PBLHs. The three model PBLH all show very shallow PBLH near the topical ITCZ (Intertropical Convergence Zone) region.

This is similar to what was found in RSR 24 ([RD.12]).

Dr Xie also notes

It is also interesting to note the large MAD found in refractivity, dry temperature and relative humidity based PBLH over Antarctic, which will be discussed further in later section.

And, after comparing the various PBLH diagnostics to $PBLH_N$:

On the other hand, large negative bias is seen in $PBLH_{\alpha}$ over the Antarctic, which needs further study.

See later.

3.2.2.2 TAMU-CC PBLH

The reviewer also compared the ROPP results with those produced by similar, but not identical, algorithms at his home institution (Texas A&M University – Corpus Christi (TAMU-CC hereafter)). These algorithms, unlike the ROPP ones, distinguish between h_{penet}, the depth below which the profile must penetrate for it to be considered usable for diagnosing PBLH, and h_{min}, the minimum allowed PBLH. (In ROPP they are both set to 300 m.) As a result he makes the following guarded recommendation:

It could be useful to include the minimum penetration height (h_{penet}) as a separated input parameter, instead of a hard-coded parameter. A slight increase the threshold could help increase the useful RO soundings for regional studies. For example, a threshold of 500 m was used in Xie et al. (2012) over subtropical southeast Pacific Ocean. In addition, two more input parameters: the minimum and maximum cut-off height (e.g., h_{min} = 300 m, h_{max} = 5 km) could be introduced, which specifies the valid vertical range of RO profiles to be used for PBLH derivation. The input parameter h_{min} could be different from h_{penet}. Also it could be very useful to change it to be even less than 300 m when trying to identify the surface inversion that are often seen over polar regions and the nocturnal PBL over land. However, the RO sounding quality near the surface is still lack of understanding and need more studies.

Action 3.2.2.1: Add the suggestion of including h_{penet}, h_{min} and h_{max} as input parameters to the PBLH tools to ROPP ticket #446 ('PBLH devt2: Revise setting of max PBLH').

Action 3.2.2.2: Add a suggestion to examine the sensitivity of the results to the these three parameters to ROPP ticket #446 ('PBLH devt2: Revise setting of max PBLH').

To help to distinguish between well and poorly defined PBLHs, the reviewer also suggested that ROPP should calculate and output Ao et al.'s *sharpness parameter* ([RD.13]),

$$\sigma(X) \quad := \quad \max(|dX/dz|) \;/\; \mathrm{rms}(dX/dz)$$

where X(z) is the vertical profile under consideration, and both statistics (min and rms) are taken over the range of valid PBLHs (h_{min} = 300 m to h_{max} = 5 000 m). This sounds like an excellent idea. It could potentially even feature as an element of the PBLH QC flag.

Action 3.2.2.3: This suggestion of including the sharpness parameter has been added to ROPP ticket #445 ("PBLH devt1: Threshold for strength of local min/max in PBLH").

The ROPP and TAMU-CC PBLHs were found to be broadly similar, apart from:

> However, much shallower PBLHs are seen in both poles in all three RO PBLHs along with the model $PBLH_T$ as compared to ROPP PBLH diagnostics.

And, for the mean absolute deviation,

> Whereas, much smaller MAD is seen over polar regions for all three RO PBLHs and the $PBLH_T$.

Again, after directly mapping the differences between the two sets of diagnostics, Dr Xie finds:

> The two PBLHs are overall very consistent with each other, especially seen in $PBLH_q$ and $PBLH_{rh}$. However, the TAMU-CC PBLH shows lower PBLH as compared to ROPP product over the polar region and some over tropics on the other four PBLH diagnostics.

So there is a clear difference between the two PBLHs in polar regions. To try to understand why, the reviewer plotted the refractivity and temperature, and their vertical gradients, for six Antarctic profiles, together with the TAMU-CC and ROPP diagnosed PBLHs. The result, Figure 12, is reproduced below. The TAMU-CC algorithms search for the maximum vertical gradients in profiles that are almost monotonically decreasing with height over such very cold surfaces, at least above the surface layer. They therefore 'drain down' towards the minimum possible PBLH – namely, 300 m. The ROPP diagnostics, on the other hand, generally seem to miss this maximum and find a smaller local maximum gradient further up the profile. This may be due to different smoothing being applied in the two algorithms, or perhaps the interpolation of the profile data to higher (10 m) vertical resolution, as undertaken by the TAMU-CC algorithm, is playing a part. There is a suggestion that the ROPP algorithm, which uses a quadratic fit to the data around the maximum gradient to site the PBLH precisely, is placing it just below the 300 m threshold in situations where the gradients are largest at the bottom. This results in the PBLH being ruled invalid. The algorithm then investigates the next largest local maximum vertical gradient as a potential PBLH, and this is far up the profile in these situations. It is not clear that this is necessarily a bad thing, however, since a PBLH that is strongly tied to the arbitrary minimum possible value is clearly not a scientifically desirable quantity, even if, fortuitously, the procedure gives accurate values in such circumstances. But whatever the reason, the difference needs to be understood.

Action 3.2.2.4: Raise ROPP ticket #476 ("PBLH devt3: Reconcile ROPP and TAMU-CC PBLH diagnostics") to investigate the discrepancy between the two diagnostics.

Action 3.2.2.5: Add a note to the ROPP9.0 Release Notes saying that ROPP-derived PBLHs in polar regions, especially those over land, should be treated with caution.

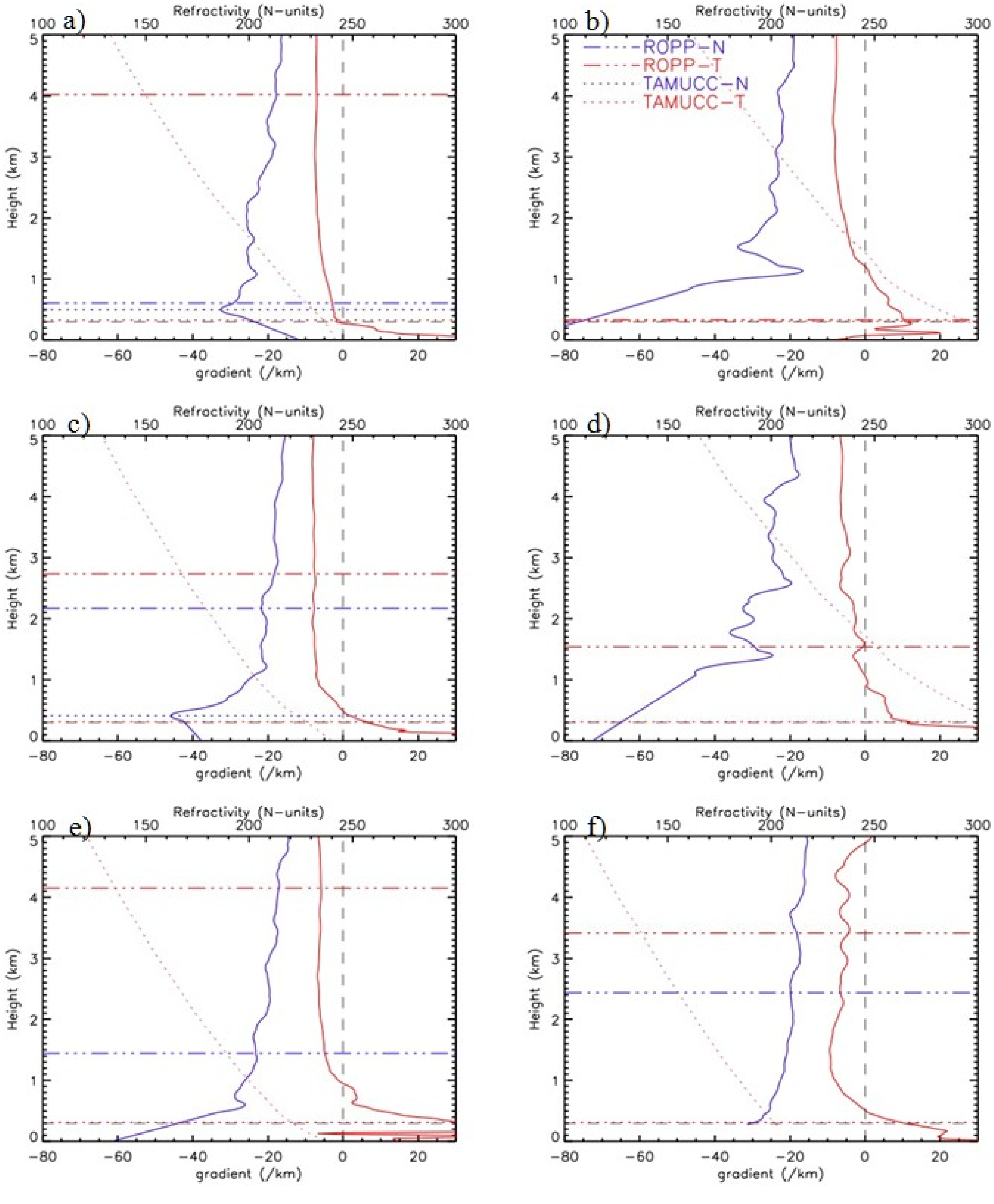

Figure 12. Six typical GPS RO refractivity profile (red-dotted) and the vertical gradients of RO refractivity and model temperature over the Antarctic (85°S-90°S). The $PBLH_N$ and $PBLH_T$ for both ROPP (dash-dotted) and TAMU-CC (dotted) are marked by the horizontal lines.

3.2.2.3 CALIPSO PBLH

Finally, Dr Xie (and colleagues) calculated PBLHs for the same period (Mar-May 2013) from measurements made by the The Cloud-Aerosol Lidar and Infrared Pathfinder Satellite Observations (CALIPSO) lidar, which derives cloud and aerosol profiles from the backscattering of a linearly polarized laser. The idea is that low clouds and aerosols are often trapped within boundary layers, so that the upper limits of their vertical extents serve as proxies for the PBLH. When the reviewer compared the CALIPSO PBLHs and the ROPP PBLHs, he found

Very similar PBLH structure, especially over the ocean are shown in comparison to the ROPP PBLH climatology. Over mid to high latitude, however, CALIPSO shows generally higher PBL except over the Antarctic and the Greenland.

And

Overall, the tropical and subtropical oceans shows the minimum difference, except near the ITCZ. Whereas, ROPP is generally lower than CALIPSO over mid and high latitudes (e.g., north of 50°N and south of 50°S). Note, however, a systematic positive bias in ROPP PBLH over the Antarctic and the Greenland is shown, which is consistent with the difference from TAMU-CC PBLH.

All this ties up with the earlier conclusions. See Actions 3.2.2.4 and 3.2.2.5.

The use of hard experimental data as a source of 'truth' is invaluable in an area often limited to intercomparisons of various theoretical measures of PBLH, or 'verification' against model-dependent reanalyses. The ROM SAF sincerely thanks Dr Xie and his colleagues for going to the trouble of generating such a useful reference dataset.

3.3 Recommendations

The reviewer made the following recommendations. The ROM SAF's responses are given.

Some recommendations are summarized below for the PBLH diagnostic tool package:

1. Make the “minimum penetration height (h_{penet})” as an input parameter, which will allow more RO sounding profiles to be used for the elevated PBL study.

   Response: See Action 3.2.2.1.

2. Introduce a new parameter called “h_{min}” along with the “h_{max}” to specify the valid vertical range in searching for the height of minimum (or maximum) gradient as the PBLH.

   Response: See Action 3.2.2.1.

3. Make the vertical sampling of RO profiles to be 100 m or even smaller, e.g., 10-50 m to minimize the interpolation errors in the gradient method. This could be especially important for bending angle profile.

   Response: Generally, the vertical resolution of observed or model fields is not under the user's control, of course. But if the reviewer means that the fields should be interpolated to a higher resolution before calculating the vertical gradients, then it is not clear at first sight that this would be helpful. (Linear interpolation from a coarse grid to a fine grid would generate piecewise constant gradients with discontinuities at the data points. The current ROPP method is based on linear interpolation of the gradients between their (2nd order

accurate) values halfway between the data points. It is not clear that this is worse. However:
Action 3.3.1: Open ROPP ticket #477 ("PBLH devt4: Investigate sensitivity of PBLH to vertical resolution") to look into this.

4. Add the "sharpness parameter" for each profile used for PBLH calculation.

   Response: See Action 3.2.2.3.

5. Clarify the PBLH detection algorithm in the presence of surface inversion that are often observed over polar regions. The current ROPP PBLH algorithm is not consistent with the normal gradient method that simply detect the height of the minimum (or maximum) gradient as the PBLH, within the vertical range of 300 m to 5 km. It could lead to complication of interpreting the PBLH diagnostics.

   Response: See Action 3.2.2.4.

   - Might be worthy of developing PBLH algorithm to detect the temperature inversion top height as the PBLH in the presence of surface inversion. It could be useful for model temperature and humidity profiles, which however, could be challenge to apply on RO sounding profiles due to the limited vertical resolution and the restricted deep penetration capability.

   Response: We feel that defining PBLH as the location of dT/dz=0 (rather than the maximum of dT/dz) in the presence of a surface inversion is probably a step too far at this stage. Effort would probably be best invested in resolving the difference between TAMU-CC and ROPP PBLH diagnostics in regions, like the polar land masses, which are frequently shrouded by surface inversions. (The reviewer admits that this would be difficult to apply to observational RO data, which is, after all, the main point of this.) We might note, however, that the PBLH diagnostics in ROPP have been designed to be flexible enough to allow this sort of generalisation. Indeed, one could in principle apply different PBLH algorithms for different fields in different regions (at different times of the day, or year, or surface heights, or ...) . However, to ensure that the point does not get missed:

   Action 3.3.2: Open ROPP ticket #478 ("PBLH devt5: Investigate possibility of redefining PBLH in regions of surface inversions") to look into this.

4. Systematic beta testing

The beta release of ROPP9.0 has been passed through the standard ROPP "test folder", which has been extended to include tests of the new vertical interpolation scheme and of the direct modelling of the L1 and L2 bending angles. The reference data for the ropp_pp tests have also been updated as a result of a bugfix to the statistical optimisation scheme.

The results are fully discussed in the Test Plan [RD.7] and the Test Folder Report [RD.8], but in summary we have the following set of passes and failures:

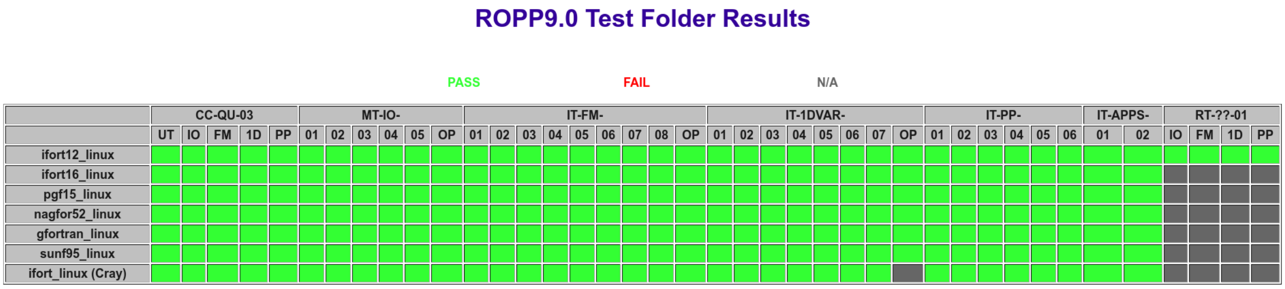

Fig 1: Summary of results of passing ROPP-9 beta through the ROPP test folder (64-bit Linux platforms).

Compared to ROPP8.0 (see [RD.5, v8.0]), there is one more unavailable test result: that of IT-1D-VAR-OP on the Cray XC40. This is because it takes too long for ropp_1dvar to process one day's worth of data on a single node of this massively parallel multiprocessor supercomputer. (ROPP is not parallelised. And the XC40 was not in use when ROPP8.0 was being tested.) But this test is passed on all other platforms/compilers, and all other tests pass on XC40.

Figure 1 shows that ROPP is robust and reliable over a wide range of compilers and platforms. Figure 2, which shows the dependency packages that can be built with each compiler, implies that most users should be able to find a compiler to build the functionality (BUFR, GRIB, netCDF-4 etc) that they need. The situation is rather better than at ROPP8.0 (see the corresponding figure in [RD.5, v8.0]). Now it is only nagfor52 that has problems with some non-essential external dependencies (even if compilation of the netCDF-Fortran library needs to be nursed through sometimes); previously, netCDF-4 was unbuildable with nagfor52 and (an earlier release of) sunf95.

ROPP9.0 Availability of External Dependencies

BUILDS OK   USABLE   FAILS TO BUILD OR TO BE BUILT ON   N/A or UNKNOWN

	ZLIB (1.2.8)	HDF5 (1.8.16)	NETCDF		BUFR		GRIB-API (1.14.5)
			netCDF-Core (4.4.0)	netCDF-Fortran (4.4.3)	MetDB (24.0.2)	ECMWF (000387)	
ifort12_linux	(1a)	(1a)	(1a)				
ifort16_linux	(1a)	(1a)	(1a)	(2)			
pgf15_linux	(1a)	(1a)	(1a)				
nagfor52_linux	(1a)	(1a)	(1a)	(3)		(4)	(5)
gfortran_linux	(1a)	(1a)	(1a)	(6)			
sunf95_linux	(1a)	(1a)	(1a)				
ifort15_linux (Cray)	(1b)	(1b)	(1b)				

(1a) C-library, compiled with **gcc (4.4.7).**

(1b) C-library compiled with **gcc (4.9.1).**

(2) Initially fails to recognise `nf_inq_path`; builds OK if this failure is followed by a manual `make install`.

(3) Initially fails with an undefined reference to `` `exit_' ``; builds OK if this failure is followed by a manual `make install`.

(4) Apparently builds OK, but ropp_io cannot use it because the build system cannot find `bufren` in the `bufr` library.

(5) Error: same_int_long.f90, line 4: KIND value (8) does not specify a valid representation method. (nagfor uses unusual KIND values.)

(6) Initially fails to recognise `nf_endian_big` etc; builds OK if this failure is followed by a manual `make install`.

Fig 2: Summary of available compilers for ROPP9.0 dependency packages.

5. Informal testing

ROPP9.0 has been informally tested by users within the ROM SAF. Here are their thoughts.

5.1 Sean Healy, ECMWF

This user tried to build ROPP9.0 on his linux box. Here is his report, repeated verbatim with permission.

I downloaded ropp-9.0.tar.gz from the beta test page.

1) gunzip ropp-9.0.tar.gz

Fine

2) tar -xvf ropp-9.0.tar

Fine

3) Building

I initially tried

buildpack netcdf gfortran

but it failed because I wasn't aware of the need for hdf5.
So I downloaded hdf5-1.8.18 and tried.

buildpack hdf5 gfortran

Failed again in one of the checks.

```
Testing  istore
============================
 istore  Test Log
============================
Test sizes:  SMALL MEDIUM LARGE
Testing istore create                                              PASSED
Testing istore extend: 10                                          PASSED
Testing istore extend: 10x10                                       PASSED
Testing istore extend: 10x10x10                                    PASSED
Testing istore extend: 10000                                       PASSED
Testing istore extend: 2500x10                                     PASSED
Testing istore extend: 10x400x10                                   PASSED
Testing istore sparse: 5                                           PASSED
Testing istore sparse: 3x4                                         PASSED
Testing istore sparse: 2x3x4                                       PASSED
Testing istore sparse: 30                                          PASSED
Testing istore sparse: 7x3                                         PASSED
Testing istore sparse: 4x2x3                                       PASSED
Testing istore sparse: 50x50x50                                    PASSED
*FAILED*
0.19user 0.59system 0:12.06elapsed 6%CPU (0avgtext+0avgdata 12240maxresident)k
32inputs+2120496outputs (0major+3983minor)pagefaults 0swaps
make[4]: *** [istore.chkexe_] Error 1
make[4]: Leaving directory `/home/st/sti/ropp-9.0/hdf5-1.8.18/test'
make[3]: *** [build-check-s] Error 2
make[3]: Leaving directory `/home/st/sti/ropp-9.0/hdf5-1.8.18/test'
```

```
make[2]: *** [test] Error 2
make[2]: Leaving directory `/home/st/sti/ropp-9.0/hdf5-1.8.18/test'
make[1]: *** [check-am] Error 2
make[1]: Leaving directory `/home/st/sti/ropp-9.0/hdf5-1.8.18/test'
make: *** [check-recursive] Error 1

*** Failed to build HDF5 library
```

I then downloaded hdf5-1.10.0-patch1

and tried again.

\>>>>>>>>>>>>>>>>>>>>>>
Qu. HDF5 really slows down the build process. Why do we need it in ROPP?
\>>>>>>>>>>>>>>>>>>>>>>

Answer: The netCDF libraries that are recommended for use with ROPP are netCDF-4, and it was agreed at the last meeting of ROPP Governance Group that build support for 'classic' netCDF-4 (netCDF-3 mode) would be dropped. This implies a requirement for HDF5 and (optionally) zlib.

Another hdf5 failure.

```
Use Case tests failed with 4 errors.
2.30user 2.19system 0:34.01elapsed 13%CPU (0avgtext+0avgdata 21940maxresident)k
660736inputs+5198184outputs (0major+139127minor)pagefaults 0swaps
make[4]: *** [test_usecases.sh.chkexe_] Error 1
make[4]: Leaving directory `/scratch/rd/sti/ropp-9.0/hdf5-1.10.0-patch1/test'
make[3]: *** [build-check-s] Error 2
make[3]: Leaving directory `/scratch/rd/sti/ropp-9.0/hdf5-1.10.0-patch1/test'
make[2]: *** [test] Error 2
make[2]: Leaving directory `/scratch/rd/sti/ropp-9.0/hdf5-1.10.0-patch1/test'
make[1]: *** [check-am] Error 2
make[1]: Leaving directory `/scratch/rd/sti/ropp-9.0/hdf5-1.10.0-patch1/test'
make: *** [check-recursive] Error 1

*** Failed to build HDF5 library

------------------------------------------------------------------
** An error occured. Check messages to find the problem
------------------------------------------------------------------
```

I then tried hdf5-1.8.16

buildpack hdf5 gfortran

```
------------------------------------------------------------------
Done. Check messages to confirm successful build & installation
------------------------------------------------------------------
```

\>>>>>>>>>>>>>>>>>>>>>>>>>>

It works. Sensitivity to release. Package noted in release notes. My fault.

\>>>>>>>>>>>>>>>>>>>>>>>>>>

OK, but it's concerning that there is this sensitivity.

Back to the build

buildpack netcdf gfortran

Passed!

buildpack ropp_utils gfortran

Passed

buildpack ropp_io gfortran

Fail.

```
configure: WARNING:
configure: WARNING: PACKAGE NETCDF NOT FOUND
configure: WARNING: THIS PACKAGE REQUIRES NETCDF TO BE INSTALLED FIRST.
configure: WARNING: *** NOTE:                                          ***
configure: WARNING: *** Users wishing to install ROPP_IO must first have ***
configure: WARNING: *** the NETCDF package installed before building     ***
configure: WARNING: *** this package. See ROPP Release Notes or ROPP     ***
configure: WARNING: *** User Guide for further details.                  ***
configure: WARNING:
configure: error: Module NETCDF not found
```

But, I'm not using the correct netcdf versions. I had to download.

netcdf-c-4.4.0.tar.gz, netcdf-fortran-4.4.3.tar.gz

buildpack netcdf gfortran
buildpack netcdff gfortran

Again all release notes.

buildpack ropp_io gfortran

pass

buildpack ropp_pp gfortran

pass

buildpack ropp_fm gfortran

pass

buildpack ropp_1dvar gfortran

pass

Summary on Build
****

It works but the changed dependencies on other software means a user can't just do what they did last time, without checking the user documentation.

Lazy users will find this more difficult. More positively, the release notes do contain the information. For example, I made a mistake by picking the wrong hdf5 release, but the correct information was in the release notes.

I THINK WE SHOULD EMPHASISE THE IMPORTANCE OF THE CHANGES LISTED IN THE RELEASE NOTES. PREVIOUSLY, I'VE GOT AWAY WITHOUT PAYING MUCH ATTENTION TO THEM.

Response: OK, we will stress this more in the Release Notes and Change Logs. (Part of the problem is that there is so much documentation that users will decline to read any of it. Less is sometimes more.)

The HDF5 makes it slow to build.

Response: Agreed. Note that it is not necessary to install hdf5 if, as is quite likely, it already exists on the user's system in a location that the compiler 'link' clause '`-lhdf5 -lhdf5_hl`' can pick-up. Of course, if this version of hdf5 clashes with the user's netCDF-Core installation then a compatible version would need to be installed.

##

####################
Some extra testing.
####################

I thought this would be easy because all the tests have been passed.

Response: Naivety like this can be touching sometimes.

In

/scratch/rd/sti/ropp-9.0/ropp_pp/tests/

I ran

test_pp_wopt.sh

```
Running t_pp_wopt_1 (PP WOPT; quick options) ...
./../tools/ropp_pp_wopt_tool: error while loading shared libraries: libnetcdff.-
so.6: cannot open shared object file: No such file or directory
./ropp_pp_compare: error while loading shared libraries: libnetcdff.so.6: cannot
open shared object file: No such file or directory
... examine t_pp_wopt_1.log for details
```

>>>>>

Did I set ROPP_ROOT correctly????

The problem might be I'm running on scratch, because of the size.

Re-installing. No luck.

It turns out I needed this.

```
export LD_LIBRARY_PATH=/scratch/rd/sti/ropp-9.0/gfortran/lib/:$LD_LIBRARY_PATH
```

to run test_pp_wopt.sh

##

I think this is bit confusing. Running buildpack ropp_pp gfortran calls all the tools fine - THEY PASS - but then you can't just "cd" into the directory and run the tool thats been tested.

HOWEVER, AGAIN THE INFORMATION IS IN THE RELEASE NOTES.

Response: This is a cultural difference associated with the use of shared object libraries. If many users object, we should probably revert the default position to be the building of static libraries, as for previous releases.

Action: Discuss the shared/static library issue at ROPP 9.0 DRI.

Documentation of ROPP PP User Guide

###############

Page 47, start

The 55 cases.

All 55 1D and 2D refractivity profiles are available from the ROM SAF web pages (ROM SAF, 2016).)

Perhaps change this to "All 55 1D cases will be available". They are not there yet. I will try to do it next week.

OK, will be done.

Page 49, top

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

Change to "simplifications"?

OK, will be done.

5.2 Santi Oliveras, IEEC

This user tried to build ROPP9.0 on a PC running Ubuntu, and a Debian computer. Here is his report, repeated verbatim with permission.

Hi Ian,
I've successfully installed Full ROPP in my PC (Ubuntu 16.04) with gfortran.
I've downloaded and installed all dependences (except ECMWF bufr).

See the logs in the attached file: ubuntu_16.04_gfortran_logs.tgz

My installation commands at my PC where:

```
bond:~/ropp/gfortran> history | grep build|grep -v less
  112  12:13  ./buildzlib_ropp gfortran
  115  12:14  ./buildhdf5_ropp gfortran
  117  12:28  ./buildnetcdf4_ropp gfortran
  120  12:30  ./buildnetcdf4_ropp gfortran
  147  12:38  ./buildpack netcdf gfortran
  148  12:51  ./buildpack netcdff gfortran
  150  12:53  ./buildpack mobufr gfortran
  153  12:54  ./buildpack grib gfortran
  158  13:07  ./build_ropp gfortran
```

Response: Close study of these files revealed a failure from trying to use `buildnetcdf4_ropp`, which tries to use configuration files like `netcdf4_configure_<compiler>_<os>`, which no longer exist at ROPP9.0. (They're all netcdf4 by default now.) The user had realised this and got around the problem by manually installing netCDF.

Action: `buildnetcdf4_ropp` has been removed from the distribution at r5187.

Apart from that, everything was fine.

I've successfully installed Full ROPP in a debian Computer (Debian 7.11) with gfortran.
See the logs in the attached file: debian_7.11_gfortran_logs.tgz

My installation commands at the debian Computer:

```
oliveras@kilimanjaro:~/ROPP/ropp-9.0$ history | grep build|grep -v less
  505  ./buildzlib_ropp gfortran
  506  ./buildhdf5_ropp gfortran
  507  ./buildpack netcdf gfortran >& ~/ropp/gfortran/buildpack_netcdf.log
  509  ./buildnetcdf_ropp gfortran
  510  ./buildmobufr_ropp gfortran
  511  ./buildgrib_ropp gfortran
  512  ./build_ropp gfortran
```

Response: Everything was fine: a clean sweep.

This user made several suggestions for making it clear to the general user on how to build ROPP. As a result, the following lines have been added to the Release Notes:

Other shell wrapper scripts **`build*_ropp, build_deps`** and **`build_ropp`** are provided which can be used to further automate the build process by calling **`buildpack`** with a predetermined sequence of packages or compilers and which save a copy of all screen output to a disk log file. Review and edit to suit your requirements. Using these tools, a complete check out of ROPP from scratch can be effected by running (in order):

> **`buildzlib_ropp <compiler>`**

> **`buildhdf5_ropp <compiler>`**

> **`buildnetcdf_ropp <compiler>`**

> **`buildmobufr_ropp <compiler>`** or **`buildecbufr_ropp <compiler>`**

> **`buildgrib_ropp <compiler>`**

> **`build_ropp <compiler>`**

Or, even more quickly:

> **`build_deps <compiler> zlib hdf5 netcdf netcdff mobufr/ecbufr grib`**

> **`build_ropp <compiler>`**

We thank the user for this advice, which we think is likely to be very helpful.

5.3 Stig Syndergaard, DMI

This user attempted to build ROPP 'from scratch' with the ifort compiler, but with static libraries (as were used with earlier versions of ROPP). After much effort from all sides, it turns out that to build netCDF-Fortran with static libraries, the following lines need to be added to the default mini-configuration scripts **netcdff_configure_<compiler>_<os>**, so that the netCDF-Fortran package can link to its underlying netCDF-Core package:

```
LD_LIBRARY_PATH="$PREFIX/lib" \
LIBS="-lhdf5 -lhdf5_hl -lnetcdf" \
```

Action: A note to this effect has been added to the Release Notes.

The difficulties suffered by this user strengthen the need, as noted in Sec 5.1, to discuss the shared/static library issue at the ROPP 9.0 DRI.

5.4 Axel von Engeln, EUMETSAT

This user built and ran ROPP9.0 without incident. The one point he mentioned was that the 'summary table' of the automatic make tests is missing from the `ropp_io` module.

Response: This is a known issue. It's annoying, because the `ropp_io` module is probably the one that needs the summary table of tests most, because this is the one for which the number of tests run is the most variable, due to the (un)availability of external dependencies. The problem is that the tools tested in this module generate output in a variety of formats, including BUFR and ascii. This complicates the 'automatic comparison' procedure which is implicit in the generation of these test result tables. But the issue has not been forgotten about; indeed it is still logged for inclusion in ROPP10.0 as ROPP ticket #273.

5.5 Ian Culverwell, Met Office

This user attempted to build ROPP on a PC using the gfortran compiler via the Cygwin linux emulator. There were difficulites[1] building the recommended version of HDF5 (v1.8.16), or the version used at ROPP8.0 (v1.8.8), but the 'native' version, which can be installed as part of the Cygwin download, allowed netCDF-Core and netCDF-Fortran to be built on top of it.

Action: Add a note on this to the Release Notes.

The ECMWF GRIB and BUFR libraries also failed to build. (The same was true, at least for ECMWF BUFR, at ROPP8.0.) Tant pis. ***All the modules of ROPP built OK.***

One strange point: the 'automatic testing', undertaken as part of the build process, suggested that the ROPP Forward model tests of the '`--direct_ion`' option had not been run:

[1] All sorts of nonsense about

```
h5ls.c:2154:9: error: unknown type name 'CONSOLE_SCREEN_BUFFER_INFO'
         CONSOLE_SCREEN_BUFFER_INFO scr;
```

and

```
  H5make_libsettings.c:186:30: error: dereferencing pointer to incomplete type 'struct
passwd'
      if((comma = HDstrchr(pwd->pw_gecos, ','))) {
```

```
************************* SUMMARY OF ROPP_FM TEST RESULTS **************************
------------------------------------------------------------------------------------
|                Test name    |              Description        |    Run? |  PASS? |
------------------------------------------------------------------------------------
|                  t_fascod_1 |                     FM FASCOD 1D |     Run |  PASS  |
|               t_fascod_tl_1 |                  FM_TL FASCOD 1D |     Run |  PASS  |
|               t_fascod_ad_1 |                  FM_AD FASCOD 1D |     Run |  PASS  |
|                  t_fascod_2 |               FM FASCOD 1D -comp |     Run |  PASS  |
|               t_fascod_tl_2 |            FM_TL FASCOD 1D -comp |     Run |  PASS  |
|               t_fascod_ad_2 |            FM_AD FASCOD 1D -comp |     Run |  PASS  |
|                  t_fascod_3 |             FM FASCOD 1D -new_op |     Run |  PASS  |
|               t_fascod_tl_3 |          FM_TL FASCOD 1D -new_op |     Run |  PASS  |
|               t_fascod_ad_3 |          FM_AD FASCOD 1D -new_op |     Run |  PASS  |
|                  t_fascod_4 |       FM FASCOD 1D -comp -new_op |     Run |  PASS  |
|               t_fascod_tl_4 |    FM_TL FASCOD 1D -comp -new_op |     Run |  PASS  |
|               t_fascod_ad_4 |    FM_AD FASCOD 1D -comp -new_op |     Run |  PASS  |
|                  t_twodop_1 |                          FM TWOD |     Run |  PASS  |
|                  t_twodtl_1 |                       FM_TL TWOD |     Run |  PASS  |
|                  t_twodad_1 |                       FM_AD TWOD |     Run |  PASS  |
|                  t_twodop_2 |                    FM TWOD -comp |     Run |  PASS  |
|                  t_twodtl_2 |                 FM_TL TWOD -comp |     Run |  PASS  |
|                  t_twodad_2 |                 FM_AD TWOD -comp |     Run |  PASS  |
|                    t_iono_1 |                     FM L1 and L2 | Not run | ------ |
|                 t_iono_tl_1 |                  FM_TL L1 and L2 | Not run | ------ |
|                 t_iono_ad_1 |                  FM_AD L1 and L2 | Not run | ------ |
|                   t_fm_1D_1 |           FM 1D; default options |     Run |  PASS  |
|                   t_fm_1D_2 |          FM 1D; compress factors |     Run |  PASS  |
|                   t_fm_2D_1 |           FM 2D; default options |     Run |  PASS  |
|                   t_fm_2D_2 |          FM 2D; compress factors |     Run |  PASS  |
|                 t_fm_iono_1 |              FM iono; L_neutral |     Run |  PASS  |
|                 t_fm_iono_2 |               FM iono; L1 and L2 |     Run |  PASS  |
```

Yet the log files clearly show that they had been run:

```
Testing ropp_fm with direct modelling of L1 and L2
Running t_fm_iono_1 (FM iono; L_neutral) ...
*** Results log of t_fm_iono_1 (FM iono; L_neutral) ***
./../tools/ropp_fm_bg2ro_1d  -f
../data/bgr20090401_000329_M02_2030337800_N0007_YYYY.nc  -o
bgr20090401_000329_M02_2030337800_N0007_YYYY_neut.nc  -d
-----------------------------------------------------------------------
```

```
                     ROPP Forward Model
-----------------------------------------------------------------------
INFO (from ropp_fm_bg2ro_1d):  Processing profile    1 of      1
INFO (from ropp_fm_bg2ro_1d):  (BG_20090401000329_UNKN_U999_DMI_)
----------------------------------------------------------------------
                     ROPP FM File Comparison Tool
----------------------------------------------------------------------
... (from ropp_fm_compare):  Comparing
bgr20090401_000329_M02_2030337800_N0007_YYYY_neut.nc and
../data/bgr20090401_000329_M02_2030337800_N0007_YYYY_iono_ref.nc: the results of running
test t_fm_iono_1 (FM iono; L_neutral)
... (from ropp_fm_compare):  Both files contain    1 profiles
... (from ropp_fm_compare):  No significant differences between
bgr20090401_000329_M02_2030337800_N0007_YYYY_neut.nc and
../data/bgr20090401_000329_M02_2030337800_N0007_YYYY_iono_ref.nc
****************************
**********  PASS  **********
****************************
... examine t_fm_iono_1.log for details
```

Action: Open ROPP ticket #482 to look at this (at a very low priority).

6. Acknowledgements

On behalf of the ROM SAF, and the wider community of ROPP users, we sincerely thank Dr Fei Xie for reviewing the beta release of ROPP8.0, and Sean Healy, Santi Oliveras, Stig Syndergaard and Axel von Engeln for testing the provisional release on a variety of platforms and compilers.