ROPP v 2.0

Beta Test Report

Josep M. Aparicio
Environment Canada

20081212

1) Summary

The ROPP (Radio Occultation Processing Package) is a suite of software tools prepared by the GRAS SAF as a deliverable to the GRAS community of users.
At the time of writing, the ROPP 2.0-beta had been prepared. The deliverable ROPP 2.0 is intended for distribution in short, and an external report was requested on the correctness, clarity and usability of the package by users outside the development team.

As this document will detail, the main conclusions are:

- Correctness: No major flaws have been detected. To the level tested, the software performs its intended function, and has a good or very good overall quality in this respect. Some minor improvements are possible, as recommended in the report. However this review finds that most of these are of low priority, and can safely be left for future releases.
- Clarity: The structure of the package can be improved for clarity. In most instances, this is due to choices made by the developers that are unnecessarily complex, when simpler choices would have been sufficient. Some minor corrections can be appropriate for this release, but the reviewer encourages a major effort towards future releases.
- Usability: The reviewer has identified the risk that the package may appear too complex for external users, which will hinder its adoption by the community. Given that the scientific quality of this package is good, the major recommendation is to improve the usability through its clarity and simplicity.

In the following, the report of actions and observations is written in normal typeface. Comments and opinions (including recommendations) are written in *italic* typeface.

2) The Package

ROPP 2.0-beta was initially made available to the reviewer Aug 29th, 2008. It was placed in the ROPP web page from GRAS SAF. General access requires registration. Access to the beta version requires supplementary approval. This approval had in this case already been granted. The layout of the web site is otherwise sufficiently clear.

NB. As commented below, the reviewer suggests the next modifications to the web page:

- *A FAQ document, whose access does not require registration.*
- *The reorganization of the package, to one file per version.*

In its initial version, the package was available as a collection of files (9) grouped under a directory. One of them is a tar of the other 8. Other .tar.gz files are also present. As it turns out, these are subpackages performing specific functions, and for certain uses only a subset is necessary.

NB It is a wide recommended practice to present the ensemble of a package as one .tar.gz file. This file should have an intuitively obvious name (including the name of the package and the version) and create during unpacking an upper level directory. If the delivered package contains several subpackages, these can best be separated as different subdirectories in the unpackaged structure. The rationale is that, when confronted with the repository site, the user can quickly have the certainty that the entire package is being downloaded. To split the package implies that the user will also wonder

- *how many subpackages exist*
- *how many he or she needs*
- *once downloaded, if they include all the necessary files for his or her job, or if some important things were left in the remaining sections*
- *it will also increase the complexity of the downloading, unpacking, compiling and ultimately the usage of the package*

The user needs to understand at some point, of course, all these matters. The recommended practice suggests that the best moment for these thoughts, for a package of the size of ROPP, is after downloading and unpacking, and not before.

The reviewer downloaded the global tar. This file unpacked the other .tar.gz files in the current directory.

NB The recommended practice is that the upper-level file should create a directory and unpack inside that directory.

The ensemble of unpacked files includes

- A copyright notice (*Ok*)
- A README (*A generalized version mismatch was noted. A message was sent to the ROPP team to suggest a number of modifications which would fix most of the above issues, including this one, i.e. to number the upper level package and directory, but avoid numbering the inner one, as, indeed, most of the documentation has only minor changes between releases*).

- Subpackage 1dvar (an elementary analysis system)
- Subpackage config (an ensemble of configuration scripts)
- Subpackage fm (i.e. the forward model, for refractivity and bending angle)
- Subpackage io (reading/writing files)
- Subpackage pp
- Subpackage tools

As suggested above, it is recommended that the subpackages be turned into subdirectories.

Each subpackage contains self-configuration files, supplementary copyright notices, README and INSTALL files, automatic Makefile setup and a number of code subdirectories.

A strategic issue was identified here: although certain subpackages may eventually be used in a self-contained way (i.e. as stand-alone executables), some others will likely be used mostly, or even only, within a context of integration with other user code. The strategic target for this latter kind of code is not to succeed in compiling it "out of the box" in the user's machine, but to help the user integrate it together with his or her own code. It is thus not necessarily critical, as a first action, to configure/make/install the different subpackages, as many user objectives are not the creation of executables, nor will most of the code be used without modification. One of the major users' objectives that ROPP must cover is to help them to cleanly separate the next two sets:

- *A subset of ROPP code that has been primarily developed to be integrated into the user's existing code. The developers may provide test and example code. However these executables are only educational.*
- *A subset of ROPP code that has been developed to be used as an executable "black box" (e.g. decoders), with little or (ideally) no modification by the user, and that should work "out of the box".*

ROPP contains both kinds of code. However, the structure of the subpackages does not follow exactly this distinction, nor is this properly documented. Since this is not an issue of correctness, it can safely be left for future releases. It is however strongly encouraged to address this for the next release, as it is critical for widespread adoption of the code by the community.

It is highly recommended the development of easily available elementary documentation which illustrates what ROPP is and is not, why the user may want it, and the different potential uses. An example may be a short but prominent Frequetly Asked Questions (FAQ) file, available without registering. A copy of the FAQ should be present also in the upper level directory. The FAQ must be very short and concise.

An effort must be invested in helping the user

- *Determine, at the earliest possible stage, what kind of use or uses will be made of the package, among the next:*
    - *To integrate certain specific tools/subroutines (e.g. operators) into the user's existing code.*
    - *To create self-contained executables to perform specific functions (e.g. encoders/decoders/recoders)*
    - *To illustrate, through examples, possible solutions to user needs.*

- *Direct the user to a few recommended options, rather than trying to solve all possible deployment environments.*
    - *Integration into existing code requires, by the user:*
        - *Identification of the package parts and their function*
        - *Understanding of the package dependency tree (not yet properly documented, and not self-evident during compilation)*
    - *The ROPP must provide, for the above, simple upper-level documentation about the parts suitable for integration, and their dependencies. An effort should be made to limit the dependencies of this code to a minimum, especially on third party libraries. The ROPP team should carefully analyze the third-party dependencies, and warn the user of the prerequirements. To some extent, the configure scripts do some checks on dependencies. However, it is unclear from the users' perspective what would solve an unsolved dependency.*
    - *Stand-alone operation works best if the deployment environment is very similar to the development environment. Although the reviewer encourages testing in a broad class of systems, this can never be complete. The user must be encouraged to deploy the system in a recommended setup.*
    - *Stand-alone operation must be clearly differentiated from code integration. The current setup of ROPP presents all code as equal, be it part of a decoder or a forward model.*
        - *Decoders require a very high level of testing and robustness. Prerequisites must be clearly stated.*
        - *Forward models and similar code should be able to compile (but not necessarily link) in a broad class of machines. It is instead important that the user readily identifies all truly relevant dependencies (the documentation is not sufficient here). The examples provided have dependencies (from other ROPP packages to third-party) that arise mostly from the **sample** user code, rather than from the forward model (which has much less). This forces the user to solve the dependencies just to compile the sample code, even if those will not be required on the end user system.*
        - *Dependency on external files/libraries should be minimal, as not always will the user be able to install those. This is especially true in operational environments.*
- *Clearly identify the prerequirements before user installation:*
    - *Required environment*
    - *Required compilers*
    - *Required libraries (netcdf and udunits were necessary, but not properly identified by the documentation)*

A critical documentation issue was identified also: some subpackages depend on others, and on third party software. The reviewer failed to find proper documentation for this issue:

- ropp-1dvar: requires
  - external typesizes, netcdf
  - internal: ropp-io
- ropp-io: requires
  - external typesizes, netcdf, udunits
  - internal: ropp-tools

Although the subpackages are presented as independent, there are in fact interdependencies. There are also dependencies of third party software that need to be resolved to compile the subpackages.
The user must be directed as a first instance to the base level among the subpackages (the reviewer identified that this was ropp-tools. The subpackage name actually suggests that this is a supplementary package, when in fact it is a fundamental one.

The subpackages have interdependencies, and some (notably tools, and then io) must be compiled and installed before attempting to compile the others. These interdependencies suggest that ROPP should either not be broken into subpackages, or the internal dependency tree must be presented at the upper levels of documentation.

The normal user will not have access to root privileges in the deployment system. ROPP can, in its current configuration, be compiled and installed without root privilege. However, this requires particular dexterity that the elementary documentation should cover. Most users will need to install, reinstall, upgrade or override some or all of the next before proceeding: netcdf, udunits, yacc, typesizes. Specially the last option (overriding in user space an upgraded version of a tool or library that already exists in the system) is particularly difficult, and the ROPP team should assume that the users will need to do some or all of those actions in user space.

The documentation must also include specification of the need to identify an installation directory for the prerequisites (external: netcdf, udunits, and internal: tools, io) and the need to inform the compiler where the required include files and libraries have just been placed. These directories will in general not be located in the default areas (e.g. /usr/local/), which are root-only. The user is responsible for the specification of these paths (using prefix and FCFLAGS) during the configure stage, before the make stage. However, ROPP should help the user to identify the information that ROPP needs and that only the user can provide. Specifically, ROPP should warn

- *what compiler (full path) is active*
- *what path is being used to search for libraries*
- *whenever a library is not found, print a message identifying*
  - *where it was searched*
  - *how to add a search path if it happens to reside elsewhere*
  - *hint how to install it if it is not at all present in the system*

The README, and ideally a prominent FAQ, should mention this, or most users will fail to compile the package. The reviewer has spent most of the time dedicated to compilation to resolve dependencies. The error messages during configuration are not sufficiently

clear. A configure error must not merely report a problem, but should address the user to the solution. For instance, the next error message would be helpful:
Package ropp-tools was not found. It must be installed before configuring this package. If it is already installed in your system, you must point to its include directory adding FCFLAGS="-I/directory" as an option to the configure script.

3) Correctness

The code has been revised and a number of outputs tested. The only major issue of correctness that has been identified has been the failure in all tested computers to recode certain netcdf files (UCAR format). After commenting this issue with members of the ROPP team, it was found that the recoding should be applied to only certain kind (family) of files. In such a case, the package should test whether the netcdf file is of the allowed family, and otherwise print a message indicating that only certain family or families can be processed.

All other recoders, once installed, have worked properly.
The reviewer has identified several instances in the geodesy subpackage where improvements could be possible:

- Geometric vs geopotential transformations: Several details were found not to be of sufficient accuracy. The expressions used conform to estimations that are standard in meteorological contexts. However, they are less accurate than the GPSRO technology is, and thus will become the limiting factor for the data. The reviewer considers this as "correct" in the sense that conforms to standard practice. However this should be addressed in future ROPP versions.
- Earth radius: The quantity estimated (Somigliana) is not the radius of curvature. It is not an accurate estimation of the radius of curvature (which depends not only on latitude, as coded, but also on the azimuth). It is not more accurate for the estimation of the geopotential than a fixed average radius. Instead, it is computationally more expensive. The error is comparatively small in the way and context that this quantity is used in this release. However, it could lead to very substantial errors if this quantity was used in other contexts within GPSRO applications where the Earth radius is also needed. This should be addressed as future releases of ROPP may very likely require this quantity for other purposes.

4) Review of requested issues

The beta test request included a number of specific issues for which comments were explicitly requested:

1. Documentation: Two kinds of documentation must be considered:
    - General (README, INSTALL, FAQ): These documents provide the first (critical) guide to what ROPP is and how to proceed. They should be short but informative. Essential information, such as prerequirements, supported platforms, etc, should be present. It is recommended that the authors revise these documents before release.
    - Detailed: It is in general appropriate. The authors should consider in the long term converging towards version-independent detailed documentation. This review does not consider that the detailed documentation is currently a major issue.

2. Unpacking & compilation: The authors state the following in the review request: “While some users may only wish to use some components from the complete package, it is recommended that all users build and test the whole package to assure themselves that the parts they require do what they expect them to do in an integrated environment before using individual components in another application.” The reviewer fully agrees with this statement, both as a general test and as an introduction to the package by the user. It is recommended that this, or a similar statement, be provided to the users within the general documentation.
    1. The package requires 3rd party tools, notably netcdf and udunits. The prerequirements are not properly documented.
    2. The installation scripts of ROPP’s subpackages do an excellent job testing the requirements. However, on failure, they are not sufficiently explicit as to which action would correct the situation.
    3. Notable points of failure are
        a. the absence of netcdf or udunits in the system
        b. the order in which the subpackages must be compiled
        c. certain subpackages must not only be compiled, but also installed, for other packages to be configured and compiled
        d. the visibility of the prerequirements from the compilation directory. The libraries and modules must either be installed in a default path (often not an option, as /usr/local or similar are in rootspace) or the path must be explicited. The elementary documentation should assume that this will be necessary.
    4. The reviewer succeeded in completing the entire testing in two machines:
        a. Linux-32bit, pgi Fortran, netcdf 3.6.1.
        b. Linux-64bit, gfortran, netcdf 4.0.
    5. The reviewer tested also two other machines, but it was not possible to solve all the prerequirents in these. These are operational environments, and installing too many libraries and packages is strongly discouraged. Tests were limited to code integration. The environments are
        a. IRIX, MIPS Fortran
        b. IBM, XLF
3. The tests were in general successful, although some errors were noticed. The errors happened to be minor, but the failures provide little information, thus the user will likely be confused. The netCDF file supplied, supposedly in UCAR format, did not transform properly to ROPP format. However all tests with atmPrf files originating from UCAR were processed properly.
4. Tests with own data were successful when applied properly. The ucar2ropp does not warn that it is supposed to process only atmPrf files.
5. The reviewer has access to variational assimilation tools. In 3DVar mode, these tools can perform the equivalent job to a 1DVar. We prefer to use our own, more realistic tools for actual applications, and consider the supplied 1DVar as useful only for illustration and development. It is comparable to other tools we had developed. For testing and development applications, this 1DVar tool is appropriate and we have not found relevant interfacing problems.

6. The ROPP package fills some voids in our RO suite. We had some development tools and converters. ROPP contains additional converters, and provides an alternative development environment. The format converter may prove particularly useful.
7. We recommend improving the introductory documentation and the information returned on error.
8. The ROPP will likely be used for research applications. It will likely be used also for monitoring applications (cross-testing with different tools). It will not be used in operational NWP, as we already have a comparable suite of operators, which have been validated, extensively tested and are already deployed operationally. Statistical monitoring (Observation-Background) will likely include ROPP as a comparison operator. These applications would be deployed in Linux environments (very similar to those more extensively tested).

5) Conclusions

- In order to release this version, the reviewer makes two major recommendations
    - To add some simple but prominent aids to installation, including prerequirements, directions for user space installation and clarification of the ROPP internal tree structure.
    - To investigate the failure of the ucar2ropp decoder.
- Several minor points for improvement were identified in the geodesy package. These do not affect however the overall correctness of the system.

6) Miscellanea

During the testing period of the ROPP-2.0-beta, it became apparent that a user-detectable error was present in ROPP, later identified as version 1. The error was brought to the attention of the GPSRO community as a bias in the North American region by certain sources of NRT GPSRO data. It turned out that, effectively, the bias was equivalent to the local height of the geoid and affected the entire globe.
The reviewer had previously verified several sources of data, and found no special geoid-related bias. If it existed, it could easily be identified, as it is possible to identify, through its meteorological signature, systematic vertical biases of a size of 5m, and probably as small as 1m. The local geoid undulation has sizes of several decametres, and at certain locations exceeding 100m. However, it averages to about 0.5m worldwide. Latitudinal averages are also small. It is thus non-trivial to detect this signature in large scale averages, unless targeted specifically (i.e. regional averages, or correlation vs. geoid height). However, it may be a major source of bias for regional NWP models.

The NRT data stream being received at the time showed a clear signature of a bias correlated with the geoid height (correlation over 0.95) in data received from METOP, CHAMP and GRACE. The cause seemed to be an error in an earlier version of ROPP. As the NRT streams were fixed (first METOP, then GRACE), the signature reduced by an approximate factor of 20 in comparisons against background meteorological fields. We have only CHAMP data prior to the fix. This factor reduces the bias to within the current tolerance in NWP (about 5-10 m).

This had been previously been tested with data processed by ROPP-2, and nothing special was found. The source data (unprocessed by ROPP) had been previously searched also, and no obvious error was found in any source.

However, a small residue that seems correlated to the geoid shape is still identified in all sources of data (UCAR, GFZ and EUMETSAT) peaking at or below 5m vertically, thus close to the noise level, and not outside the current tolerance. The pattern is different between sources, and of still small meteorological significance, thus not considered a major priority. Its source is so far unknown to the reviewer, and likely related to several approximations made by the different processors. Despite its small meteorological signature, the reviewer encourages the careful revision of the ROPP geodesy package (this would apply to other postprocessors as well), as the size of the possible signatures of geodesy is

- comparable to the current noise level of NWP data for systematic biases
- will be larger than the NWP noise level as NWP requirements become more demanding