ROM SAF Report 19
Ref: SAF/ROM/DMI/REP/RSR/019
Web: www.romsaf.org
Date: 30 April 2014

The EUMETSAT
Network of
Satellite
Application
Facilities

ROM SAF
Radio Occultation Meteorology

ROM SAF Report 19

Implementation of the ROPP two-dimensional bending angle observation operator in an NWP system

Sean Healy

ECMWF

SUBMITTED

Document Author Table

	Name	***Function***	***Date***	***Comments***
Prepared by:	S. Healy	ROM SAF Project Team	30 April 2014	
Reviewed by:	C. Burrows	Met Office	1 May 2014	
Reviewed by:	S. English	EUMETSAT	12 June 2014	
Approved by:	K. B. Lauritsen	ROM SAF Project Manager	19 May 2014	

Document Change Record

Issue/Revision	***Date***	***By***	***Description***
1.1	8 July 2014	S. Healy	Minor update of section numbers

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 GRAS radio occultation data from the Metop satellites and RO data from other missions. The ROM SAF delivers bending angle, refractivity, temperature, pressure, and humidity profiles in near-real time and offline for NWP and climate users. The offline profiles are further processed into climate products consisting of gridded monthly zonal means of bending angle, refractivity, temperature, humidity, and geopotential heights together with error descriptions.

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

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

Intellectual Property Rights

Abstract

The Radio Occultation Processing Package (ROPP) includes a two-dimensional (2D) bending angle operator. This has been tested in the ECMWF numerical weather prediction system, with a view to operational implementation possibly during 2014. This report outlines how the 2D operator is implemented at ECMWF. Issues related to parallel computing architectures are discussed. We explain why problems associated with the 2D "occultation plane" spanning more than processor (or core) do not arise at ECMWF. This is because the observations are split into "pools" containing roughly equal numbers of each observation type for load-balancing, and the forward modelling is decomposed into distinct horizontal and vertical interpolation tasks. Recent results with the 2D operator are presented showing an improvement in the bending angle departure statistics with respect to observations, indicating that the forward model errors are reduced. However, the additional computational cost during the 4D-Var minimization is large. New ideas to reduce the computational cost of the ROPP 2D operator are discussed, and a new approach is suggested based on the incremental formulation of 4D-Var.

Contents

1 Introduction

GPS radio occultation (GPS-RO) measurements have a two-dimensional (2D) limb geometry (Figure 1.1) which ideally should be accounted for when they are assimilated into numerical weather prediction (NWP) systems. However, assimilating observations is always a trade-off between forward model accuracy versus computational complexity and cost. Currently, the major operational global NWP centres assimilate GPS-RO measurements with one-dimensional (1D) observation operators, and inflate the total observation errors to partially compensate for this source of forward model error. Given that most centres have reported positive results with 1D operators, this is clearly a reasonable approach. However, the largest impact has been seen for upper-tropospheric and stratospheric temperatures. The implementation of 2D operators is potentially a way of extending this impact further into the troposphere.

A number of two-dimensional (or non-local) operators have been suggested (e.g., Eyre, 1994; Poli, 2004; Syndergaard *et al.,* 2005; Sokolovskiy *et al.,* 2005; Healy *et al.,* 2007), but producing the operator is only one aspect of the problem. Technical questions have arisen, querying how such operators can be integrated efficiently into existing operational data assimilation systems. Apart from the recent exception of Zhang *et al.,* (2014), this issue is not generally addressed in the GPS-RO scientific papers.

The ROM SAF's Radio Occultation Processing Package (ROPP) includes a 2D bending angle operator (Healy *et al.,* 2007), which has been implemented and tested in the ECMWF NWP system (Integrated Forecast System, IFS). This report is aimed at NWP users of ROPP who are considering implementing the 2D operator. It describes how the 2D operator has been implemented in the ECMWF system. The report complements the information provided in the ROPP user-guide, which provides more detail on the computations contained within 2D bending angle operator. It also addresses the question of computational expense, and examines new approaches to speed up the operator.

The main difference when assimilating bending angle measurements with a 2D operator, rather than a 1D operator, is that the NWP information must be available at multiple locations within a 2D slice, defined by the "occultation plane", to perform the 2D bending angle computation. In contrast, the 1D operator only requires a single profile at a given location to compute the bending angles. The occultation plane is defined geometrically by the position of the GPS and LEO satellites and the local centre of curvature, which might be offset from the centre of the earth (see Syndergaard 1998). The need for NWP information at multiple locations to forward model an observation value requires modification of the data assimilation code at a fairly fundamental level, because it is highly likely it will have been designed assuming only one NWP profile is required per observation value.

Another potential complication arises because NWP systems are run on parallel computing architectures, where more than one computer or processing "core" is used to solve the problem. So, for example, the forward modelling of all the observations assimilated in the four-dimensional variational assimilation (4D-Var) system is distributed over multiple computers (cores). The details of the parallel programming system are usually hidden from satellite

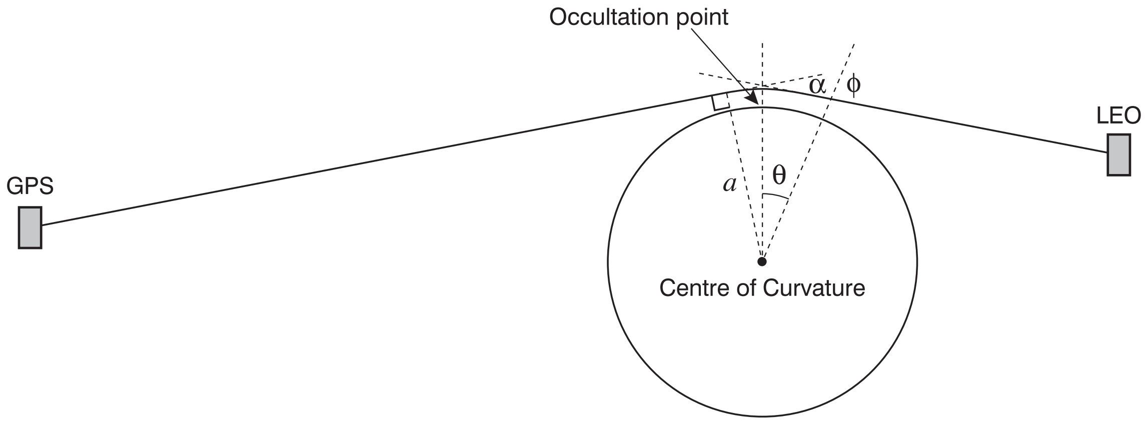

Figure 1.1: The GPS radio occultation geometry. Around $\sim$ 67 % of the bending occurs on a 450 km section of path, centred on the occultation point assuming an exponentially decaying atmosphere.

data scientists when implementing observation operators, because they are part of the basic infrastructure of the data assimilation system. The aim here is to introduce some parallel computing concepts relevant to the 2D forward operator implementation. In particular, this report will deal with the situation illustrated in Figure 1.2, where the occultation plane crosses a geographical boundary, across which 1D forward model computations are usually performed on different processors (cores). This leads to frequently asked questions of the form:

"What happens when the two-dimensional slice in the occultation plane requires profiles from more than one processor?"

For example, Zhang *et al.,* (2014) recently noted that parallelization of a non-local operator is difficult when the ray-path intercepts several "sub-domains" in a limited area model. It will be explained why this particular problem does not arise with the current ECMWF NWP system because of the way the forward modelling is partitioned into separate "horizontal" and "vertical" interpolation tasks.

In section 2 we will briefly introduce the 4D-Var problem for context. The relevant parallel programming concepts, architectures and terminology required here will be summarised in section 3, using various aspects of the 4D-Var forward modelling as examples with simplified pseudocode. The implementation of the 2D operators at ECMWF is described in section 4. Results from recent testing and potential cost savings are describe in section 5. This is followed by the summary.

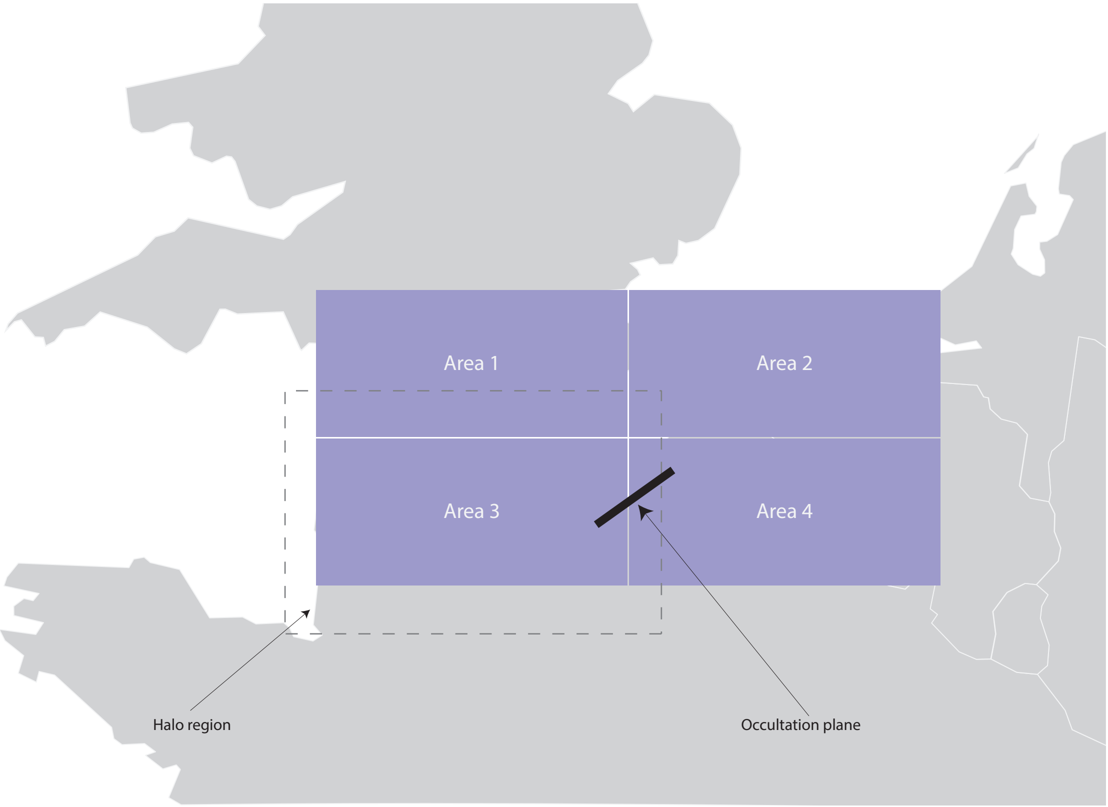

Figure 1.2: An occultation plane crossing a geographical domain which has been subdivided into four “areas” (sub-domains) for illustration. The occultation plane, denoted by the thick black line, intersects both Area 3 and Area 4. A key question is how to forward model this case with a 2D operator, when 1D observations in each area are normally forward modelled on separate computers (cores). The dashed line around Area 3 is a “Halo Region”. These are required for all areas to enable interpolation close the boundary of the domain, but they have no specific relevance to the 2D modelling.

2 The Data assimilation Problem

In data assimilation, we aim to find the atmospheric state, $\mathbf{x}$, which minimizes the cost function,

$$\begin{aligned} J(\mathbf{x}) &= (\mathbf{x}-\mathbf{x_b})^T\mathbf{B^{-1}}(\mathbf{x}-\mathbf{x_b})+(\mathbf{y}-H(\mathbf{x}))^T\mathbf{R^{-1}}(\mathbf{y}-H(\mathbf{x})) \\ &= J_b+J_o \end{aligned} \tag{2.1}$$

where $\mathbf{x_b}$ is a short-range forecast, $\mathbf{y}$ is the vector of all observations, H is the composite forward model, including the integration of the forecast model, which maps the model state space to observation space for all observation types; $\mathbf{B^{-1}}$ and $\mathbf{R^{-1}}$ are the inverses of the assumed error covariance matrices for the short-range forecast and observation vector, respectively.

For the purpose of this report, it is useful to decompose the observation operator $H = H_v H_h$, where H_h represents the horizontal interpolation of the model state to the observation's time and location, and H_v is the operation on the interpolated vertical profiles, such as calculating radiances or bending angles. For the case of four-dimensional variational assimilation (4D-Var) we will assume that the NWP forecast model integration is part of H_h.

The ECMWF 4D-Var computation is performed in a parallel computing architecture.

3 Some concepts of parallel programming

ECMWF provides training courses in parallel computing. See for example:

http://www.old.ecmwf.int/services/computing/training/material/com_hpcf.html

Information from two lectures, the *"Concepts of Parallel Computing"* by George Mozdzynski and *"An Introduction Parallel Programming"* by Paul Burton have been used to compile this report. Parallel computing can be defined as:

The simultaneous use of more than processor or computer to solve a problem.

Parallel computing techniques are necessary to solve the NWP 4D-Var minimization problem efficiently. ECMWF uses a hybrid system, based on the two parallel computing architectures illustrated in Figure 3.1:

- Distributed Memory
- Shared memory

In a distributed memory system, the computational problem is split into a set of processes or "tasks", each performing a subset of the original computation on a separate "core". This is achieved by splitting the algorithm and/or data into subsets that can performed in parallel. The term "core" is simply used as a standalone computer here, but they are sometimes called "processors" or "processing elements" [1]. Figure 3.1a shows four cores. These cores do not share memory, so if information from one core is required by another, this requires communication or "message passing" via a network, which can be expensive. Message Passing Interface (MPI) is a software specification for libraries used in the communication between the tasks on a distributed memory system. The subsets of computation performed on the distributed systems are called "MPI tasks". Figure 3.1b shows a shared memory parallel architecture. In this case, each core works on a subset of the computational problem, but no communication is required.

The ECMWF NWP system combines the distributed memory and shared memory approaches in a hybrid system. A hybrid system with 3 MPI tasks and 12 cores is illustrated in Figure 3.2. The use of a hybrid system is probably most easily understood with an example based on a simplified version of the computation of (observed minus simulated) departures in 4D-Var. For the moment we will only consider the vertical interpolation, H_v, and assume the NWP forecast model has been run, and that the information has already been interpolated to the observation locations.

Consider initially computing H_v on a pure MPI system, like Figure 3.1a. In the operational 4D-Var problem, we assimilate of order 10 million observation values in a 12 hour assimilation window. This number is comprised of various observation types (or observation sets),

[1] There is some ambiguity on the terminology used here.

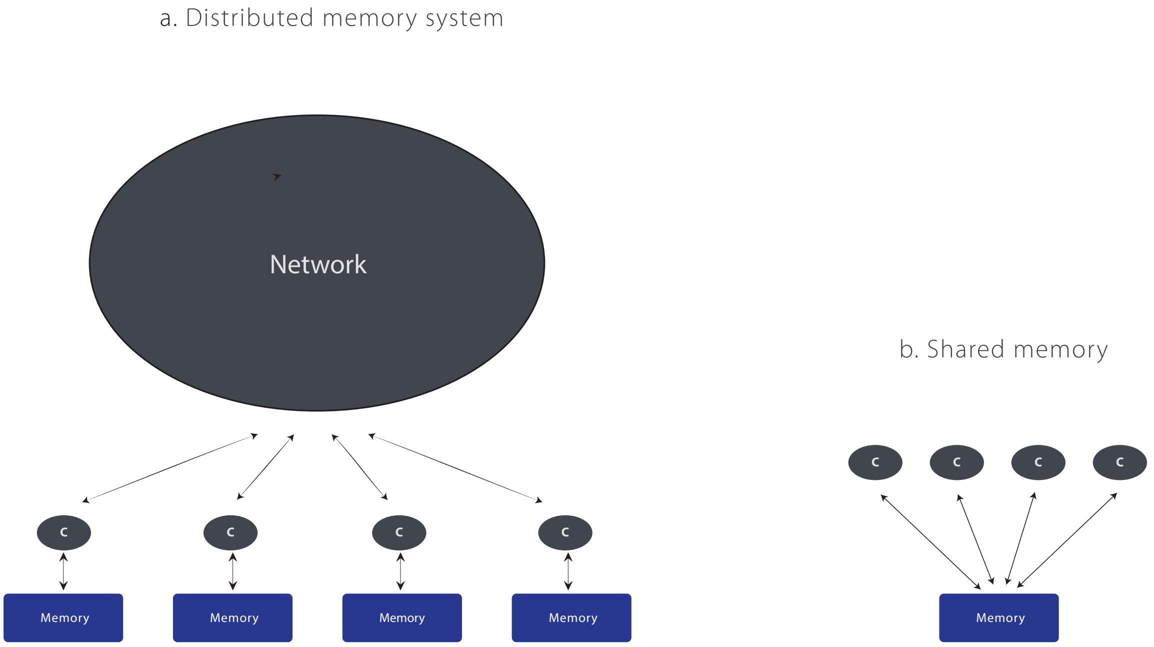

Figure 3.1: Two parallel computing architectures a) Distributed memory and b) shared memory. “C” denotes core.

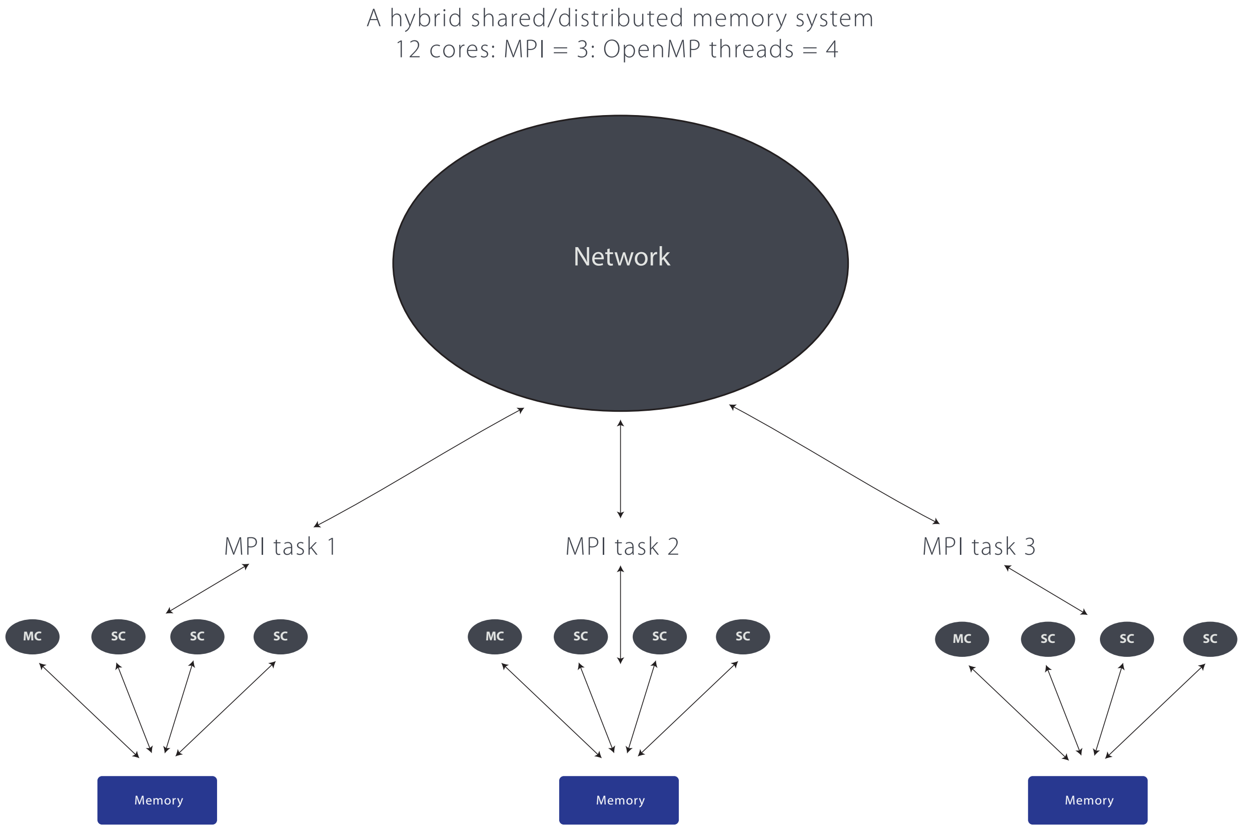

Figure 3.2: A hybrid distrubuted/shared memory parallel computing system, as used at ECMWF. This example has 3 MPI tasks, each with 4 threads. “MC” denotes master core and “SC” denotes slave core.

such as conventional, satellite radiances, GPS-RO bending angles, etc. The observations are split into a set of “pools”, where each pool contains roughly the same number of each observation type. The geographical locations in each observation pool are *randomly* distributed across the globe. The forward modelling, H_v, for each pool of observations is performed as a distinct MPI task.

The composition of the observation pool, with roughly equal numbers of each observation type, ensures that the computational cost of the forward modelling is roughly the same for each MPI task. This is an example of “load balancing”, which aims to ensure that each core of the supercomputer performs roughly the same amount of work.

The H_v MPI tasks can be performed in parallel, because there should be no inter-dependencies (after the model integration/horizontal interpolation) when forward modelling the subset of observations in each distinct observation pool. For each MPI task we forward model and compute the departures for `N_OBS_TYPE` types of observation, such as conventional data, radiances, GPS-RO measurements, etc. in the data pool. We can loop through each observation type, for example radiosondes, `i`, in the pool, for the set of observations in `obs(i)`,

```
! Compute observed minus simulated departures for ob. type "i".

DO i = 1, N_OBS_TYPE

   CALL compute_departures(i,obs(i),state(i),departures(i))

ENDDO
```

using the interpolated NWP information for observation typei stored in state(i) to compute the departure values in departures(i) with the the forward operator appropriate for observation type, i. This section of “pseudocode code” is not the actual code code used at ECMWF, but a simplified version that illustrates the main points discussed here.

This loop is reasonable if each MPI task is performed on a single core, similar to the configuration shown in Figure 3.1a, but it can be speeded up in a hybrid system, like that shown in Figure 3.2. We can employ a shared memory approach for each MPI task to speed up this loop for each pool, because there should be no reason why, for example, the forward modelling of the radiances in the observation pool must be completed before starting to forward model the GPS-RO measurements. This is a second level of parallelization.

In a hybrid system, the observation type loop can be distributed across multiple “threads”, which share the same memory. In Figure 3.2, 12 cores are used in total. Each MPI task has 4 threads, composed of a “master core” (MC) and three “slave cores” (SC). By default, the computations on an MPI task are performed on the master core, MC, until compiler directives are encountered which introduce the slave cores, SC. These new compiler commands, or directives, for parallel programming on a shared memory system are known as “OpenMP”.

When using a hybrid distributed/shared memory system with OpenMP, the code above can re-written as:

```
! Compute observed minus simulated departures.
! OpenMP : Distribute loop over NTHREAD threads
! OpenMP : Private variables : i

DO i = 1, N_OBS_TYPE

   CALL compute_departures(i,obs(i),state(i),departures(i))

ENDDO
```

These new commands distribute the computations in the loop over the OpenMP threads, using both the master and slave cores, and lists any variables that are “private” on any given thread. For example, loop variables must be private to ensure that the same loop index is not being update by multiple threads. This code enables the forward modelling of the observation types to be performed in parallel for each MPI task using all the available threads. Once the loop is completed, the program reverts to using just the master core until new OpenMP directives are encountered.

The question of how to split a computational problem between the number MPI tasks and the number threads is a trade-off. On the one hand, reducing the number of MPI tasks, and increasing the number of threads, should reduce the need for message passing, which is expensive. However, increasing the number of threads can make it difficult to use the slave cores efficiently.

In general, we use the default numbers of MPI tasks and threads when running the 2D operator in the ECMWF 4D-Var system, and have not attempted to optimize this configuration specifically for the 2D operator. For information, at ECMWF the 4D-Var minimization with a 12 hour assimilation window is split into 240 MPI tasks, each with 8 cores.

4 Implementing the 2D operator

In the previous section the computation of the vertical interpolation, H_v, was used to illustrate the hybrid shared/distributed memory architecture used at ECMWF, but the horizontal interpolation to the observation location, H_h, was ignored. The horizontal interpolation is often a source of concern when implementing 2D operator. As noted earlier, a potential problem is illustrated in Figure 1.2, where part of the occultation plane is in Area 3, and part in Area 4. In this picture, each core is often *assumed* to be responsible for all the forward modelling – both H_h and H_v – in a specific region on the globe, and a problem is thought to arise if the occultation plane crosses a boundary between the regions. The combined horizontal and vertical forward modelling $H = H_v H_h$ is clearly considered within a distributed memory framework. However, this picture does not apply at ECMWF because the basic *assumption* is incorrect.

In the context of implementing the 2D operators, there is an important feature of the ECMWF system that should be emphasised:

There is no assumption – or requirement – that the H_v and H_h interpolations required for the assimilation of a given observation are performed on the same core.

In fact, the H_v and H_h computations are actually performed as part of different sets of MPI tasks, with the mutual dependencies dealt with through message passing. In the terminology used at ECMWF, H_h is computed in "model space" MPI tasks, and then H_v is computed in a set of "observation space" MPI tasks. In model space, each MPI task is responsible for performing all of the horizontal interpolations required for a well defined geographical region. We note the need for a "halo" around each geographical region to enable *any* interpolations near the boundary, but this has no specific relevance to the implementation of 2D operators. The concept of separating the model space and observation space computations, was designed before development and implementation of 2D operators, with the general purpose of load balancing the forward modelling (e.g., Hamrud, 1998). A similar approach is not adopted in the Met Office 4D-Var computation, for example.

The ECMWF system works in the following way. A pre-processing routine loops through all the observation locations in each observation pool, and then uses a relatively simple algorithm to determine which model space MPI task is responsible for interpolating the NWP information to each observation location (See Figure 4.1). The information mapping where the H_v and H_h steps are actually computed is stored in a set of tables. Using message passing, the observation locations from the pools are sent to the MPI tasks responsible for performing the horizontal interpolations. The horizontal interpolations are performed, and then the interpolated NWP profiles are passed back to the cores performing respective H_v MPI tasks.

The tangent-linear and adjoint computations are decomposed exactly in the same way, with the corresponding tangent linear and adjoint routines always being computed on the same cores. Clearly, in the adjoint computations the direction of communication is reversed.

Hence, the adjoint of the observation space computations are passed to the cores responsible for the adjoint of the model space (horizontal interpolation) computations. This enables the combined impact of all the observations in a given area to be accumulated.

It is interesting to note that the load balancing of the model space computations may be poor, with some geographical regions requiring many more profiles than others. However, this is not a major problem because the cost of the horizontal interpolations is generally only a few percent of the vertical interpolations (Hamrud, 1998).

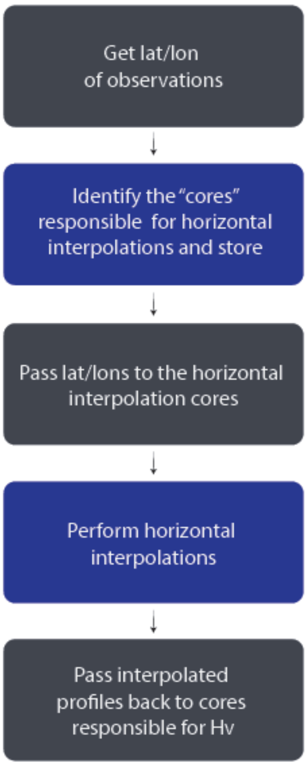

Figure 4.1: The main steps when interpolating in model space, after the data has been split into the observation pools.

The key point to be emphasised here is that the interpolated NWP profiles required for an arbitrary H_v task are randomly distributed across the globe. The H_h tasks are already computed elsewhere, and then message passed to enable the computation of H_v. By default in the ECMWF system, there is an *assumption* that only one NWP profile is required per observation, but this is not fundamental. The introduction of the 2D operator merely increases the number of profiles required to compute H_v for each pool, but the additional, new locations do not affect the computation of H_v in any other important way.

When using the using 2D operator we generalise the code and enable `N_HORIZ` interpolated NWP profiles for each observation location. Currently, by default `N_HORIZ=1` for all observation types, except for GPS-RO when the 2D operator is employed. We have to compute the number of interpolated NWP profiles required in each pool, because this is no longer equal to the number of observations locations. This is done using a loop of the form:

```
number_of_profiles = 0

! loop through each observation type, i, in pool

DO i =  1, N_OBS_TYPE

! assume the number of observations of type i is numb_obs(i)
! and they each require N_HORIZ(i) profiles
!
   number_of_profiles = number_of_profiles+(numb_obs(i)*N_HORIZ(i))

ENDDO
```

The locations of the NWP profiles within the occultation plane needed for the 2D operator are computed using the location of the tangent point and the limb azimuthal angle provided in the operational BUFR files. The computation of the positions within the plane is performed using the ROPP routine `ropp_fm_2d_plane.f90`. The preprocessing loop noted above – which identifies the cores responsible the for horizontal interpolation – simply loops through the locations in the occultation plane and stores the relevant core information for each point in the plane. The fact that more than one core may be required to interpolate to the locations within the 2D occultation plane has no impact on the routine. The horizontal interpolations are performed on the relevant model space core, and then the interpolated profiles are message passed back to the observation space core responsible for H_v. By construction all required data data will be available when computing H_v, and there are no specific difficulties associated with the 2D operator.

We note that Zhang *et al.,* (2014) circumvent the problem in a different way in their local area system, by making the entire refractivity field available to every core when using the non-local phase operator.

Changes within `compute_departures`

The pseudocode used to explain the hybrid shared/distributed memory system can be adapted to illustrate how the 2D operators are called in the ECMWF system, once the horizontal interpolations, H_h, have been performed. As before, we compute the (observed minus simulated) departures for each observation type, but introduce the new variable `N_HORIZ` which defines how many NWP profiles each observation type requires to perform the forward modelling, H_v.

```
! Compute departure.
! OpenMP : Distribute loop over NTHREAD threads
! OpenMP : Private variables : i

DO i = 1, N_OBS_TYPE

   CALL compute_departures(i,obs(i),state(i),departures(i),N_HORIZ(i))

ENDDO
```

This change requires some revision of the arrays used in the `compute_departures` routine, with the addition of an extra dimension. Most importantly for the GPS-RO assimilation, the temperature, pressure, humidity and geopotential height arrays have three dimensions determined by the number of observations (`NOBS`), number of vertical model levels (`NVERT`) and the number of points in the plane (`N_HORIZ`). The routine that computes the pressures and geopotential heights of the model levels also requires some revision.

Some of the basic structure of the `compute_departures` routine is the following (Also see Figure 4.2):

```
SUBROUTINE compute_departures(i, ..., N_HORIZ)

INTEGER i ! obstype. EG, conventional or GPS-RO or radiance, ...

! model data

REAL, DIMENSION(NOBS,NVERT) :: pres, temp, shum, zgeop

! new arrays for 2D operator

REAL, DIMENSION(NOBS,NVERT,N_HORIZ) :: &
press_2d,temp_2d,shum_2d,zgeop_2d

! observation arrays (unchanged for 2D operator).

REAL, DIMENSION(NOBS,NLEVELS_PER_PROFILE) :: &
observed_values,simulated_bending_angles,depart

REAL, DIMENSION(NOBS,NLEVELS_PER_PROFILE) ::
& impact_param   ! the impact parameters

!
! This pseudocode only aims to highlight the main steps.
! The observation operator used will depend on obstype, i,
! usually selected with "IF" or "CASE" statements.
! Just assume obstype is GPS-RO here.
!

IF (N_HORIZ > 1) THEN

! we need the height/pressure of the model levels

   CALL COMPUTE_PRESSURE_HEIGHTS_2D(press_2d,temp_2d,shum_2d,zgeop_2d)

   CALL GPSRO_OPERATOR_2D&
   (press_2d,temp_2d,shum_2d,zgeop_2d,impact_param,simulated_bending_angles)

ELSE

! we need the height/pressure of the model levels

   CALL COMPUTE_PRESSURE_HEIGHTS_1D(press,temp,shum,zgeop)

   CALL GPSRO_OPERATOR_1D&
   (press,temp,shum,zgeop,impact_param,simulated_bending_angles)

ENDIF
```

```
! Note observation arrays sizes are the same for the 1D and 2D operator
! so these next two routines are the same for both the 1D and 2D.

CALL DEPARTURES(observed_values,simulated_bending_angles,departures)

! compute contribution to observation cost function, JO.

CALL COMPUTE_COST_FUNCTION(departure,JO)


END SUBROUTINE
```

Clearly, consistent modifications have to be made the corresponding tangent-linear and adjoint routines.

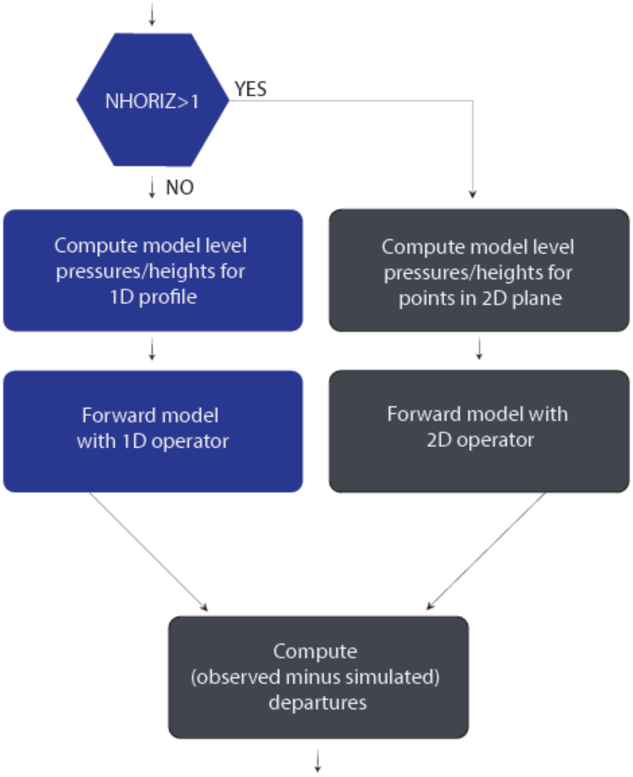
NHORIZ>1
YES
NO
Compute model level pressures/heights for 1D profile
Compute model level pressures/heights for points in 2D plane
Forward model with 1D operator
Forward model with 2D operator
Compute (observed minus simulated) departures

Figure 4.2: The main steps in the forward model routine where either 1D or 2D operators is available.

5 Recent testing in the ECMWF system

It is useful to review recent testing with the 2D operator in the ECMWF system, because this has highlighted some potential issues with the computational cost of the approach. We have run experiments covering the period January 25, 2013 to March 31, 2013, comparing the impact of assimilating GPS-RO with a 1D and 2D operator. The experiments have been performed T1279 ($\sim$ 16 km sampling), in order to to maximise the potential benefits of the 2D operator. The ECMWF incremental 4D-Var system with a 12 hour assimilation window is used to assimilate the data. The 2D operator uses an occultation plane comprised of interpolated NWP profile information at `N_HORIZ=31` locations, separated by 40 km. Both the 1D and 2D operators include tangent-point-drift.

Experiments have been run using the full observing system and assimilating just GPS-RO observations. In general, the 2D operator improves the GPS-RO departure statistics, as illustrated for example in Figure 5.1 for COSMIC-5 departures in the northern hemisphere. This shows the change the background and analysis noise normalised departure statistics, $((o-b)/\sigma_o)$ and $(o-a)/\sigma_o)$, averaged over one month. The standard deviations of the background departures are reduced by $\sim 5\%$ of the assumed error σ_o between the 10 km and 20 km impact height interval. The horizontal bars indicate that the improvements are statistically significant at 95 $\%$ level. This is encouraging because it illustrates that the forward model error is being reduced. The 2D operator has a broadly neutral impact on forecast scores relative to the full system, but has a statistically significant positive impact in the GPS-RO only experiments. However, perhaps the most important result in these experiments that the 2D operator increases the total wallclock time required to minimize the linearised 4D-Var cost function by $\sim 29\%$ (See Table 5.1). An increase in cost of this magnitude would stop the operational implementation of the 2D operator, and so computational savings have to be considered.

The following three changes have been investigated at ECMWF.

5.1 Implementation of the 2D operator in incremental 4D-Var

The incremental formulation of 4D-Var (Coutier *et al.,* 1994) provides some justification and scope for potential computational savings. The ECMWF 4D-Var system now typically performs three non-linear trajectory runs at full horizontal resolution (T1279, $\sim$ 16 km) when using a 12 hour assimilation window. These trajectory runs are also called the “outer loop”. The observation departures in the outer loop, $\mathbf{d}$, are computed using the non-linear forward model, H,

$$\mathbf{d} = \mathbf{y^o} - H(\mathbf{x_g}) \tag{5.1}$$

where $\mathbf{y^o}$ and $\mathbf{x_g}$ are the vectors containing the observations and non-linear trajectory, respectively.

The 4D-Var cost function is linearised about the non-linear trajectory and minimized. The

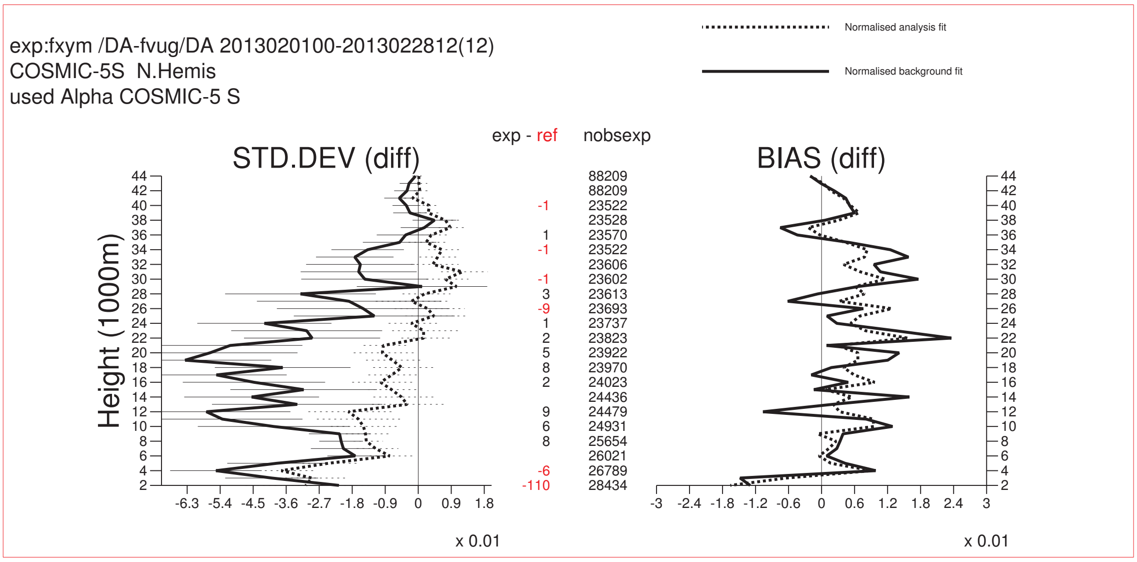

Figure 5.1: The COSMIC-5 noise normalised departure statistics for the northern hemisphere on impact heights, averaged over one month. All other observations are included in these experiments. The results are displayed as differences in the standard deviation and mean of the departures (2D results minus 1D results). The solid lines are for the (O-B) departures and the dotted lines are the (O-A) departures. Negative standard deviation values indicate that the 2D operator is reducing forward model error. The thin horizontal lines show the 95 $\%$ confidence interval.

Table 5.1: A summary of the cumulative reductions in wallclock time for the second 4D-Var inner loop as result of the proposed changes in the 2D operator. The changes are 1) reducing the number of profiles in the inner loop occultation plane to 7 2) Computing the ray-path with the midpoint method and 3) batching the bending angles into groups of 11 in the vertical. The combined impact is to reduce the additional cost of the 2D operator from 29% to 2.3%.

Description	Wallclock time (s)	Percentage Increase vs 1D
1D operator	438	–
2D operator	563	29
2D, N_HORIZ=7	472	7.8
2D, N_HORIZ=7, Midpoint	452	3.2
2D, N_HORIZ=7, Midpoint, TPD batching	448	2.3

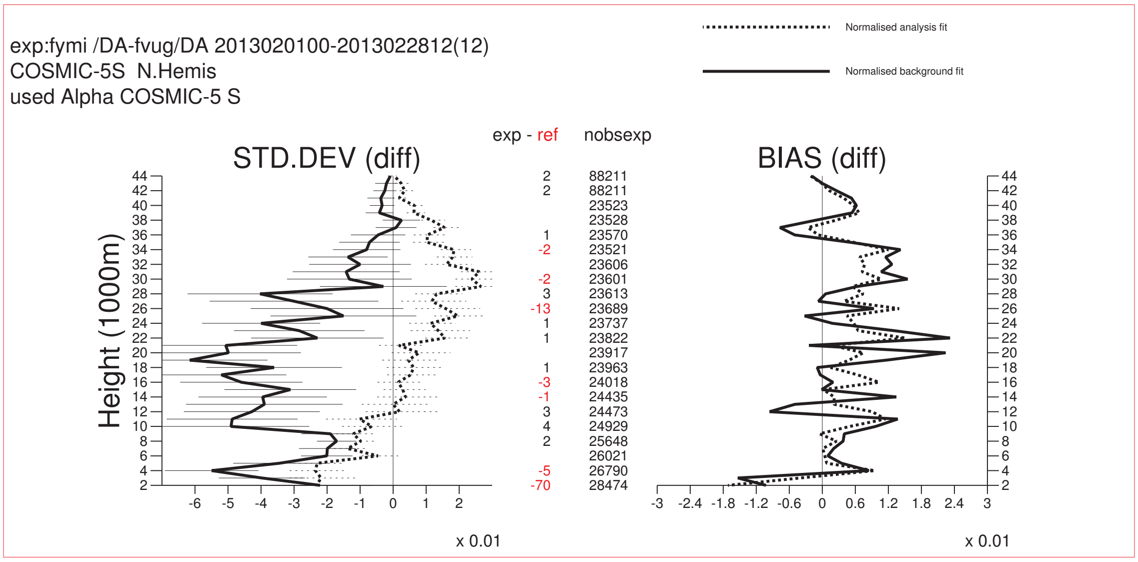

Figure 5.2: As Figure 5.1 comparing the 2D and 1D departure statistics, but the inner loop profile number has been reduced to 7 in the 2D computation.

minimization is performed iteratively in an “inner loop” at a lower horizontal resolution than the full trajectory run. This change in resolution reflects the fact the 4D-Var system is primarily correcting large scale errors in the forecast. The inner loop resolution is usually T159 and T255, corresponding to $\sim$ 125 km and 80 km, respectively. The departures in the inner loop are updated with

$$\mathbf{y^o} - H(\mathbf{x_g} + \delta\mathbf{x}) = \mathbf{d} - \mathbf{H}\delta\mathbf{x} \tag{5.2}$$

where $\mathbf{H}$ is the tangent linear of the forward model and $\delta\mathbf{x}$ is the increment.

Given the difference in resolutions used in the 4D-Var inner and outer loops, it is debatable whether the tangent-linear and adjoints of the 2D operator used in the inner loop require the same number of profiles in the occultation plane as the full 2D model used in the higher resolution outer-loop. We have experimented with reducing the number of profiles from `N_HORIZ=31` to `N_HORIZ=7` in the inner loop. The `N_HORIZ=7` profiles have a 200 km separation and therefore span the full 1200 km planar section spanned by the 31 profiles with 40 km separation used in the outer-loop. This change does not appear to significantly degrade the background departure statistics (see Figure 5.2), and it provides a considerable computational saving during the minimization (See Table 5.1). There is a small degradation in the analysis fit to the GPS-RO, probably because the ability to adjust the horizontal gradients is reduced when `N_HORIZ=7`. However, this degradation is small when compared to the difference in the size of the background and analysis departure statistics. The additional cost during the minimization of using the 2D operator is reduced from 29% to 7.8%. This approach is likely to be implemented operationally at ECMWF during 2014. However, it also interesting to note that we have found that the 4D-Var converges successfully when the 2D operator is used in the outer loop and the 1D operator is used in the inner loop. This suggests that a hybrid 2D/1D GPS-RO assimilation scheme is feasible.

5.2 Simplification of the numerical solution of the ray-path equations

The 2D operator is based on a numerical solution of the differential equations defining the ray-path in circular polar co-ordinates (r and θ) (e.g., page 149, Rodgers, 2000)

$$\frac{dr}{ds} = \cos\phi \tag{5.3}$$

$$\frac{d\theta}{ds} = \frac{\sin\phi}{r} \tag{5.4}$$

$$\frac{d\phi}{ds} \simeq -\sin\phi\left[\frac{1}{r} + \left(\frac{\partial n}{\partial r}\right)_{\theta}\right] \tag{5.5}$$

where s is the distance along the ray-path, n is the refractive index, ϕ is angle between the local radius vector and the tangent to the ray-path. This computation is performed up to a user prescribed impact height (50 km), and then the code reverts to a 1D computation.

The numerical solution of the equations in ROPP uses a 4th order Runge-Kutta (RK4) approach (Press *et al.,* 1992). For each step along the 2D ray-path the required derivatives given above are computed four times. These are produced via external subroutine calls in order to simplify the F90 code. Repeated subroutine or function calls can be a significant computational overhead. In fact, the first implementation of the ROPP 1D operator at ECMWF contained repeated external calls to a Gaussian error function. This was identified as a significant computational cost at ECMWF, even though the function itself was quite simple, comprising of a cubic polynomial times an exponential. As a result, the Gaussian error function was moved into the main routine, as in the ROPP 1D operator code. It is possible to do something similar with the 2D operator, moving the gradient computations into the main routine, although this makes the code far messier. Another simplification is to use the midpoint method for solving the equations. Formally, this less accurate than the RK4 because the truncation error is larger (Press *et al.,* 1992), but it is much easier to implement and requires only two gradient computations per step.

A midpoint approach is being tested at ECMWF, and the results suggest that it does not significantly degrade the departure statistics (Figure 5.3) and it reduces the computational cost (Table 5.1).

5.3 Implementation of Tangent Point Drift

Consider a standard GRAS BUFR file containing 247 levels on a fixed set of impact heights. Each of the 247 bending angles has its own location (latitude,longitude), but there is also a single representative location in the header of the BUFR file. In the original implementation of the 1D GPSRO bending angle operator at ECMWF, an occultation was assigned to the single location given in the header and all the bending angles were computed with this NWP information. This approximation was also used in the initial testing with the 2D operator (Healy *et al.,* 2007). Other centres implemented tangent-point-drift (TPD) in the their operator (Cucurull *et al.,* 2007, Poli *et al.,* 2009).

In 2011 ECMWF introduced TPD and this clearly improved the GPS-RO departure statistics and some stratospheric temperature and wind forecast scores, particularly near the

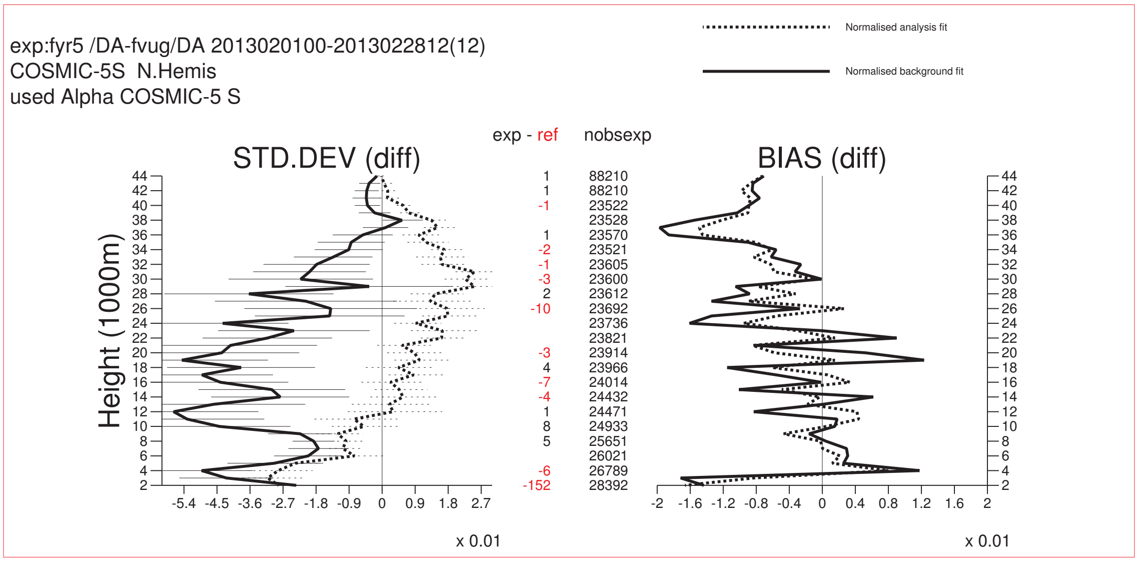

Figure 5.3: As Figure 5.1 comparing the 2D and 1D departure statistics, but the inner loop profile number has been reduced to 7 in the 2D computation, and the mid-point method is used to calculate the ray-path.

south pole (unpublished). A similar impact in the NCEP system has also been quantified by Cucurull (2012). However, there will clearly be some computational cost associated with requiring 247 NWP profiles to forward model an occultation, rather than just one. Furthermore, the implementation of the 2D operator means we require a plane of NWP profiles for each bending angle. In the recent testing with the 2D operator at ECMWF that includes TPD, we have used 31 profiles in each plane. The profiles have a 40 km separation, meaning each 2D plane spans 1200 km. The total number of NWP profiles required to model a single GRAS occultation with the 2D operator becomes 7657 (=247 $\times$ 31).

One way of reducing the number of profiles required is to batch the data vertically into subsets, and compute each bending angle in the subset using the same NWP information. This has been tested at ECMWF, batching the data into group of 11 bending angles spanning $\sim$ 2 km in the vertical. There is no obvious degradation in the departure statistics as a result of this change (See Figure 5.4) and it provides some time savings (Table 5.1.

A more sophisticated approach would be to have the width of the batching interval vary with height, so that the batches near the surface contain fewer bending angles. This idea reflects the fact that the TPD separations are greater in the lower troposphere than in the stratosphere. This will best tested in the future.

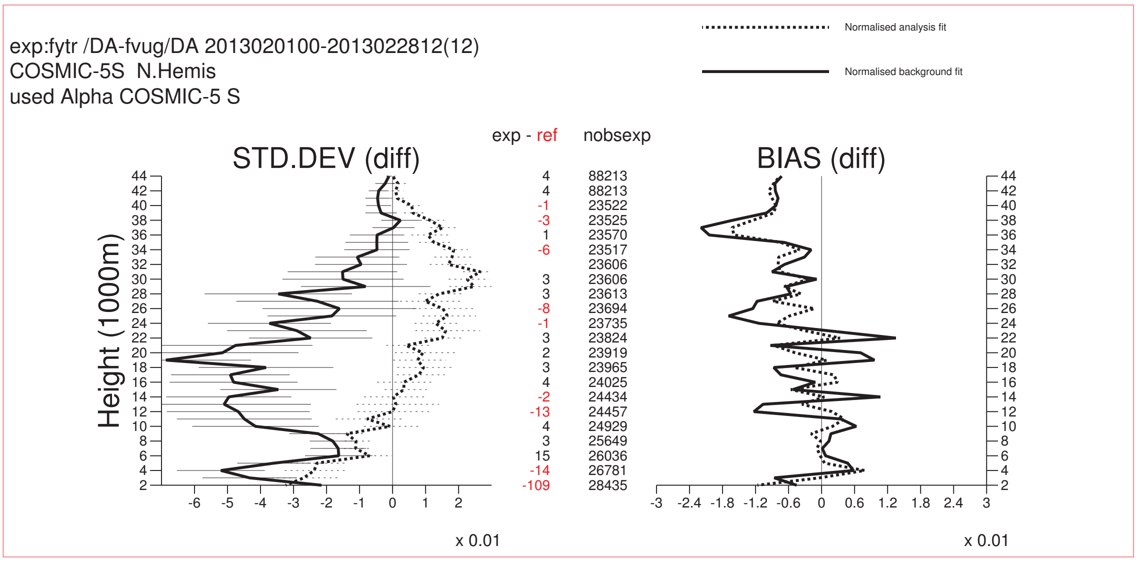

Figure 5.4: As Figure 5.1 comparing the 2D and 1D departure statistics, but 1) the inner loop profile number has been reduced to 7 in the 2D computation 2) the mid-point method is used to calculate the ray-path and 3) the bending angles are batched into groups of 11.

6 Summary

The use of 2D bending angle observation operators should improve the assimilation of GPS-RO measurements by reducing the forward model error. The ROM SAF ROPP software package includes a 2D operator that is currently being tested for operational implementation at ECMWF. This clearly improves the GPS-RO departure statistics (Figures 5.1 to 5.4), but some computational cost savings are necessary.

The technical implementation of the 2D operator in the ECMWF parallel computing system has been described. The potential problem of the GPS-RO occultation plane spanning more that one processing core does not arise in the ECMWF system. This is because the forward model problem is split into horizontal and vertical interpolations, H_h and H_v, respectively, and there is no requirement that these are performed on the same core. The use of 2D operators increases the number of interpolations preformed in H_h, but it does not change the H_v computation in any significant or fundamental way because all the globally distributed, interpolated profiles required for each observation pool are available as a result of the message passing. We note that the ECMWF approach was adopted in the 1990's to improve load balancing, and it is fortuitous that it simplifies the implementation of the 2D operators.

The computational cost of the 2D operator has been discussed. To make the operator affordable in an operational context, we have found it necessary to reduce the horizontal sampling of the profiles within the 2D occultation plane during the inner-loop computations. This appears to be consistent with the incremental formulation of 4D-Var. It is also interesting to note that a 2D operator in the outer loop and 1D operator in the inner loop also appears to be possible. Simplifications of the ray-path integration and batching the observations in the vertical have also been discussed.

Acknowledgements

I would like to thank Mats Hamrud for modifying the 4D-Var code at ECMWF to enable the use of 2D operators, and Deborah Salmond for her comments.

Bibliography

[1] Courtier, P., J.-N. Thépaut, and A. Hollingsworth, 1994: A strategy for operational implementation of 4D-Var, using an incremental approach. *Quart. J. Roy. Meteorol. Soc.*, **120**, 1367–1388.

[2] Cucurull, L., 2012: Sensitivity of NWP model skill to the obliquity of the GPS radio occultation soundings. *Atmosph. Sci. Lett.*, **13**, 55–60 doi: 10.1002/asl.363.

[3] Cucurull, L., J. C. Derber, R. Treadon, and R. J. Purser, 2007: Assimilation of Global Positioning System Radio Occultation Observations into NCEP's Global Data Assimilation System. *Mon. Wea. Rev.*, **135**, 3174–3193.

[4] Eyre, J. R., 1994: Assimilation of radio occultation measurements into a numerical weather prediction system. Technical Memorandum 199, ECMWF, Reading, UK.

[5] Hamrud, M., 1998: Parallel aspects of ECMWF's integrated forecasting system (IFS) with special empahisis on data assimilation. In *Towards Teracomputing*, World Scientific Publishing, 61–66.

[6] Healy, S. B., J. R. Eyre, M. Hamrud, and J.-N. Thépaut, 2007: Assimilating GPS radio occultation measurements with two-dimensional bending angle observation operators. *Quart. J. Roy. Meteorol. Soc.*, **133**, 1213–1227.

[7] Poli, P., 2004: Effects of horizontal gradients on GPS radio occultation observation operators. 2: A fast atmospheric refractivity gradient operator (FARGO). *Quart. J. Roy. Meteorol. Soc.*, **130**, 2807–2825.

[8] Poli, P., P. Moll, D. Puech, F. Rabier, and S. B. Healy, 2009: Quality control, error analysis, and impact assessment of FORMOSAT-3/COSMIC in numerical weather prediction. *Terr. Atmos. Ocean*, **20**, 101–113.

[9] Press, W., S. Teukolsky, W. Vetterling, and B. Flannery, 1992: *Numerical recipes in C – The art of scientific computing*, 2nd ed. Cambridge University Press, Cambridge, New York.

[10] Rodgers, C. D., 2000: *Inverse methods for atmospheric sounding: Theory and practice*. World Scientific Publishing, Singapore, New Jersey, London, Hong Kong.

[11] Sokolovskiy, S. V., Y.-H. Kuo, and W. Wang, 2005: Assessing the accuracy of a linearized observation operator for assimilation of radio occultation data: Case simulations with a high-resolution weather model. *Mon. Wea. Rev.*, **133**, 2200–2212.

[12] Syndergaard, S., 1998: Modeling the impact of the Earth's oblateness on the retrieval of temperature and pressure profiles from limb sounding. *J. Atmos. Sol.-Terr. Phys.*, **60**, 171–180.

[13] Syndergaard, S., E. R. Kursinski, B. M. Herman, E. M. Lane, and D. E. Flittner, 2005: A refractive index operator for assimilation of occultation data. *Mon. Wea. Rev.*, **133**, 2650–2668.

[14] Zhang, X., Y.-H. Kuo, S.-Y. Chen, X.-Y. Huang, and L.-F. Hsiao, 2014: Parellization strategies for the GPS radio occultation data assimilation with a nonlocal operator. *J. Atmos. Ocean. Tech.*, submitted.

ROM SAF (and GRAS SAF) Reports

SAF/GRAS/METO/REP/GSR/001	Mono-dimensional thinning for GPS Radio Occulation
SAF/GRAS/METO/REP/GSR/002	Geodesy calculations in ROPP
SAF/GRAS/METO/REP/GSR/003	ROPP minimiser - minROPP
SAF/GRAS/METO/REP/GSR/004	Error function calculation in ROPP
SAF/GRAS/METO/REP/GSR/005	Refractivity calculations in ROPP
SAF/GRAS/METO/REP/GSR/006	Levenberg-Marquardt minimisation in ROPP
SAF/GRAS/METO/REP/GSR/007	Abel integral calculations in ROPP
SAF/GRAS/METO/REP/GSR/008	ROPP thinner algorithm
SAF/GRAS/METO/REP/GSR/009	Refractivity coefficients used in the assimilation of GPS radio occultation measurements
SAF/GRAS/METO/REP/GSR/010	Latitudinal Binning and Area-Weighted Averaging of Irregularly Distributed Radio Occultation Data
SAF/GRAS/METO/REP/GSR/011	ROPP 1dVar validation
SAF/GRAS/METO/REP/GSR/012	Assimilation of Global Positioning System Radio Occultation Data in the ECMWF ERA-Interim Re-analysis
SAF/GRAS/METO/REP/GSR/013	ROPP PP validation
SAF/ROM/METO/REP/RSR/014	A review of the geodesy calculations in ROPP
SAF/ROM/METO/REP/RSR/015	Improvements to the ROPP refractivity and bending angle operators
SAF/ROM/METO/REP/RSR/016	Simplifying EGM96 Undulation calculations in ROPP
SAF/ROM/METO/REP/RSR/017	Simulation of L1 and L2 bending angles with a model ionosphere
SAF/ROM/METO/REP/RSR/018	Single Frequency Radio Occultation Retrievals: Impact on Numerical Weather Prediction
SAF/ROM/METO/REP/RSR/019	The implementation of a two-dimensional bending angle observation operator at ECMWF

ROM SAF Reports are accessible via the ROM SAF website: http://www.romsaf.org