ESA CONTRACT REPORT

Contract Report to the European Space Agency

Milestone 2 Tech Note - Parts 1/2/3: Operational Pre-processing chain, Collocation software development and Offline monitoring suite

Joaquín Muñoz Sabater,
Patricia de Rosnay, Anne Fouilloux

Milestone 2 Technical Note - Parts 1/2/3
ESA/ESRIN Contract 20244/07/I-LG

European Centre for Medium-Range Weather Forecasts
Europäisches Zentrum für mittelfristige Wettervorhersage
Centre européen pour les prévisions météorologiques à moyen terme

Series: ECMWF ESA Project Report Series

A full list of ECMWF Publications can be found on our web site under:
http://www.ecmwf.int/publications/

Contact: library@ecmwf.int

European Centre for Medium Range Weather Forecasts
Shinfield Park, Reading, RG2 9AX, England

Contract Report to the European Space Agency

Milestone 2 Tech Note - Parts 1/2/3: Operational Pre-processing chain, Collocation software development and Offline monitoring suite

Authors: Joaquín Muñoz Sabater,
Patricia de Rosnay, Anne Fouilloux
Milestone 2 Technical Note - Parts 1/2/3
ESA/ESRIN Contract 20244/07/I-LG

European Centre for Medium-Range Weather Forecasts
Shinfield Park, Reading, Berkshire, UK

September 2010

	Name	Company
First version prepared by (September 2010)	J. Muñoz Sabater	ECMWF
Quality Visa	E. Källén	ECMWF
Application Authorized by	N. Wright	ESA/ESRIN

Distribution list:

ESA/ESRIN
Luc Govaert
Susanne Mecklenburg
Steven Delwart
Norrie Wright
ESA ESRIN Documentation Desk

SERCO
Raffaele Crapolicchio

ESA/ESTEC
Catherine Bouzinac
Matthias Drusch

ECMWF
HR
Division & Section Heads

Abstract

Contracted by the European Space Agency (ESA), the European Centre for Medium-Range Weather Forecasts (ECMWF) is involved in global monitoring and data assimilation of the Soil Moisture and Ocean Salinity (SMOS) mission data. SMOS is the second Earth's Explorer mission of the ESA Living Planet Programme. Monitoring SMOS data is of special interest, as for the first time a new innovative remote sensing technique is used to observe soil moisture over continental surfaces and ocean salinity over oceans. To this purpose, ECMWF has developed an offline monitoring chain. Prior to monitoring of SMOS data, a series of pre-processing routines check all the available observations and select a consistent subset of data which input the Integrated Forecasting System (IFS) of ECMWF. As part of the monitoring chain the software includes also the routines that makes it possible to collocate SMOS observations with model grid points to allow a fair comparison with a model simulation in model space. This report is the second Milestone Technical Note / Progress Report of the ESA Request for Quotation RfQ 3-11640/06/I-LG. It is divided in three related parts, providing technical documentation of the SMOS pre-processing chain, the collocation software development and the offline monitoring suite with SMOS data.

Introduction

The introduction of a new type of satellite data in a Numerical Weather Prediction system is a challenging task. Even more so when the data comes from a new measuring technique which has never been tested before. This is the case of the SMOS research mission, being the second Earth's Explorer mission of the ESA Living Planet Programme. In SMOS, a 2D-interferometric radiometer consisting of 69 LICEF (Lightweight Cost Effective Front-end) receivers measure the electromagnetic radiation naturally emitted by the Earth's surface between 1.400 and 1.427 GHz. The objective is to measure soil moisture with a 4% accuracy and and a spatial resolution between 40-50 Km, whereas the required accuracy for ocean salinity is 0.1 psu averaged over a period of 10-30 days.

SMOS Near Real Time (NRT) products are processed at the European Space Astronomy Centre (ESAC) in Madrid (Spain) and sent to ECMWF via the SMOS Data Processing Ground Segment (DPGS) interface. The product used at ECMWF is the NRT brightness temperatures product. The NRT product constitutes a reprocessed Level-1b product and it differs with the last one in that they are geographically sorted swath-based maps of brightness temperatures. The geolocated product received at ECMWF is arranged in an equal area grid system called ISEA 4H9 (Icosahedron Snyder Equal Aread grid with Aperture 4 at resolution 9) [see SMOS_DMS_TN_5200 Document]. For this grid, over land the centre of the cell grids are at equal distance of 15 km with a standard deviation of 0.9 km. The resolution is coarser over oceans which present lower heterogeneities than continental surfaces.

In this technical note documentation about the offline monitoring chain developed at ECMWF using the NRT product is provided. It includes the pre-processing chain, where all the data is checked to be a consistent and valid set of values in the same format as the data is received: the Binary Universal Form for the Representation of meteorological data (BUFR) format. All the data that goes through the first system of filters is transformed in a format acceptable to the Integrated Forecasting System (IFS) structure, the so called Observational Data Base (ODB) format, and implementations in model space are conducted in preparation for comparison with a model simulation. As part of this technical note, the process of collocation of the observations with grid points in the model space is also provided.

Part I - Pre-processing chain

Conversion of Near Real Time (NRT) BUFR product to internal ECMWF BUFR format

Before being used by ECMWF BUFR software, BUFR data received from the DPGS is transformed into a version compatible with ECMWF software. This process is done automatically by the operations department almost as quick as the data is received. Files received at ECMWF have the following format: ”*miras_YYYYMMDD_HHMMSS_ YYYYMMDD_HHMMSS_smos_$orbit_o_YYYYMMDD_HHMMSS_l1c.bufr*”. The sequences *YYYYMMDD* and *HHMMSS* are numbers corresponding to the year (YYYY), month (MM), day (DD), hour (HH), minute (MM) and second (SS) of the MIRAS instrument integration. The first *YYYYMMDD_HHMMSS* sequence corresponds to the first MIRAS integration time within the generated product, whereas the second *YYYYMMDD_HHMMSS* sequence of numbers is the end time at what the MIRAS instrument made an observation. These conventions are defined more in detail in the SMOS NRT BUFR specification document (version 1.9).

After the conversion of format is conducted, new files compatible with ECMWF BUFR software are available. Their format is the following: *SMOS0001YYYYMMDDHH.DAT*, representing 6 hours worth of data and with *YYYYMMDDHH* the year, month and day of the corresponding data, respectively. Further, these files are currently stored in ECMWF archive system ECFS (ECMWF File Storage system). ECFS is the Centre’s archive/retrieval system for user files. It runs on a series of dedicated IBM machines, and can be accessed from all major platforms by Unix like commands. For monitoring purposes, SMOS pre-processed files will be fetched from ECFS (available in the path *ec:/emos/e/SMOS/*), ready to be used for consistency checks.

Data pre-screening

Once the data has a format compatible with ECMWF BUFR software, each individual observation is checked to be a consistent data register. These checks are the following:

1. Generic checks: files which fail to contain crucial header information are rejected: it is checked that files are encoded in BUFR format, date and time are complete, geographic coordinates are not missing and instrument data corresponds to SMOS data,
2. The validity of data is checked:
   - Individual observations are checked to be in a correct geographical position,
   - Brightness temperatures are checked to be in the range of physically reasonable values, concretely not lower than 50 K and not greater than 350 K. In this way observations affected by hard Radio Frequency Interference (RFI) are also rejected.
3. Data is thinned to reduce the volume of data which input the IFS and to avoid redundancy in the data.

If the MIRAS instrument operates normally, more than 1 Gb of data would be provided for a timeslot of 6h in dual-polarisation mode, whereas this quantity can be doubled if the instrument operates in full-polarisation mode. This amount of data cannot all be introduced in the IFS just for a single satellite instrument, taking into account that many other satellite data are used simultaneously. Thinning is therefore mandatory. Thinning is also a critical step in so far as it selects which data from the original files will be monitored, but also which data has a potential input to correct the soil moisture state and the ocean salinity value through assimilation experiments.

Data thinning can be done in many different ways. Currently thinning of SMOS data follows a very simple approach, filtering only 1 out of 10 subsets in a BUFR message. This is equivalent to thin the volume of the initial data set in about 90%. As reference 10% is an approximate benchmark of the initial volume of SMOS data that the IFS can handle. Although the volume of data is strongly reduced, however the angular signature is maintained for monitoring purposes. In the context of the next contract with ESA, a specific work package will be entirely devoted to investigate a flexible strategy to thin SMOS data. This further study will ensure an approach which will make the best possible use of the multi-angular aspect of the signal and therefore will make the most efficient use of SMOS data both for monitoring and assimilation purposes.

Fig. 1 shows a snapshot of the Supervisor Monitor Scheduler (SMS) analysis family, highlighting the *presmos* job, being part of the *prepare_obs* subfamily, where all the pre-screening tasks have been implemented.

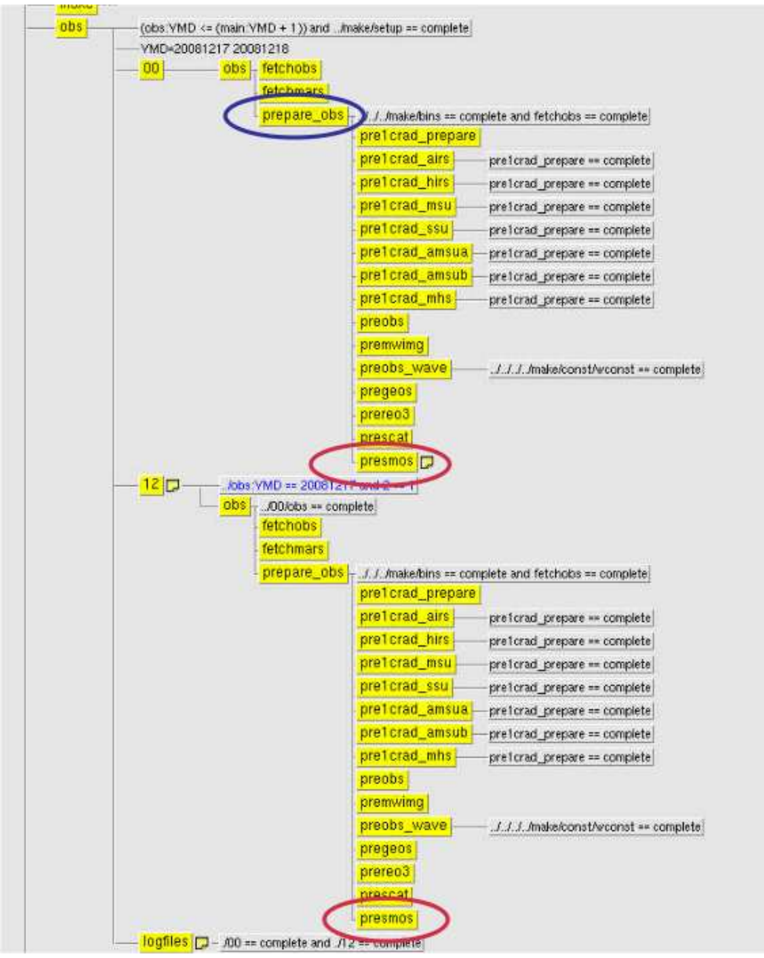


Figure 1: Presmos jobs, being part of the obs family in an experiment using 12h windows for satellite data.

Pre-screening tests

The different quality checks enumerated in section were tested with three different data sets corresponding to data of the two first months of the commissioning phase, where multiples calibration actions took place. The comparison of these data sets at three different times makes it possible to observe a clear evolution on the quality of the data still in a premature phase, as shown in Table 1.

Fig.2 shows the number of individual observations rejected as a function of the first 18000 snapshots for the files

Table 1: Number of observations rejected per snapshot during the early quality checks phase.

Date	snapshots	subsets	rejections	% rejected
28-11-2009	17940	28203176	147185	0.52
20-12-2009	17592	28739029	58967	0.21
16-01-2010	15347	24322415	34386	0.14

including the previous data. This corresponds to the first 6 hours of sensed data for these days. Only snapshots with less than 5000 subsets are shown because they correspond to pure H or V polarisation integrations. Cross-correlated polarisations are not shown here since they were not available for all files. This figure clearly shows how the number of rejected radiances is maximum for the 28*th* of November, when still no calibration was performed, and they are significantly smaller for the 20*th* of December and 16*th* of January. Large peaks in the number of rejected data are attributed to areas contaminated with RFI. In Table 1 a quantitative comparison between the three data sets is presented. It shows how the quality of the data is best in January 2010, with only 0.14% observations rejected after the first group of quality checks.

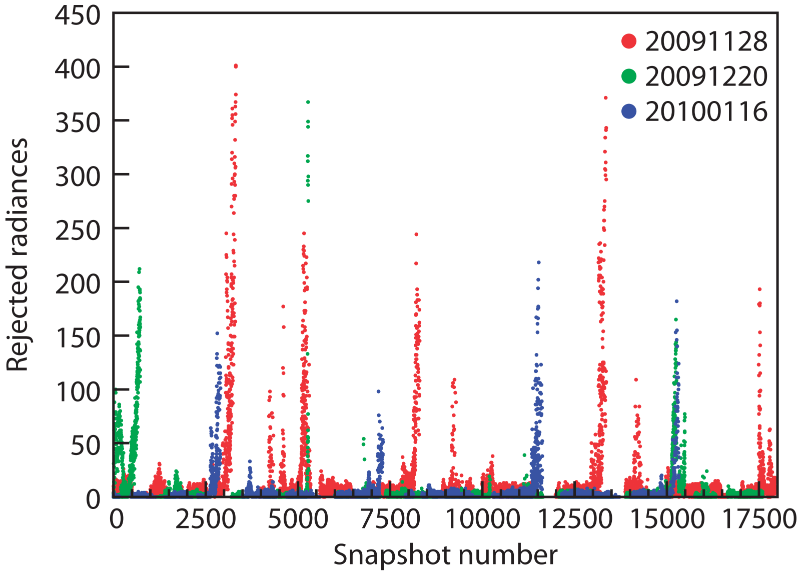


Figure 2: Number of rejected observations as a function of the snapshot number, conducted by pre-screening tasks for 6h of NRT SMOS product.

Part II - Collocation software

All SMOS data that go through the pre-screening jobs are mapped into model space. It is in this space where SMOS data is collocated to the model grid at the required model resolution. The technique used for collocation is the nearest neighbour technique. The routine that performs this job is a Fortran-90 routine called ”*smos_nearest*” (see the Fortran code in the appendix). For each observation available in model space, ”*smos_nearest*” assigns a model grid point which is the closest to the observation, and on top of that it selects which of these observations is the closest to the grid point. Fig.3 shows SMOS observations for a file containing the first 12h of data the 3^{rd} of May 2010 collocated to the ECMWF T255 spectral resolution grid ($\sim$ 80 km). For sake of clarity in the picture, a zoom over the North-East coast of United Kingdom and North Sea is shown in Fig. 4. Black dots are corresponding to grid-points. Values in the figure are the distance of grid points to the nearest SMOS observation (blue dots) expressed in metres. Note that these observations are selected after a rather simplistic thinning method is applied to SMOS data, thus observations which are relatively far to the grid point can still be selected in the pre-processing chain. The number of observations which will be monitored depends on the model grid resolution and the distance limit parameter. At T799 ($\sim$ 25 Km) and T1279 ($\sim$ 16 Km) SMOS observations within a distance limit fixed to 10000 meters are found for all grid points.

The routine *smos_nearest* is executed following a multi-processor approach. A correspondence between the processor that contains a grid point and the processor that contains an observation associated to this grid point is established. This information is stored and propagated by means of global arrays. This is particularly an important point as otherwise the matching between grid points and observations would not be accurate when gathering all the information into a single data base. This correspondence is done through the ”*smos_obs2gp*” routine. Observations are also initially flagged following the same approach through the ”*smos_iobs2gp*” routine. For more details see Section 5 of Milestone 1 Tech Note - Part 2.

The monitoring chain developed in this way resembles that of all-sky radiances for AMSR-E and SSMI data. However, the number of SMOS observations found per timeslot in model space is notably larger than for that of other microwave sensors. This is a major obstacle in the implementation as the previous routines allocate memory with a size proportional to the number of observations and the number of processors required to do this job. Hence these routines consume lot of memory specially when model resolution increases, as the number of processors is also increased. In order to avoid this major obstacle as well as other memory issues related to the internal ODB data base created for SMOS, a substantial re-structuration of the code has been made with the aim of efficiently collocate SMOS observations to model grid points. In the new structure (see Fig.5), it is the last version of the ODB software which distributes the observations per processors, associates a local grid point number to each observation and evaluates the distance to this grid point. This new structure has demonstrated to be very efficient in terms of memory consumption, as not additional global allocated variables are further required. Thus, the collocation with the model grid is efficient at any model resolution and it does not increase significantly memory resources. This new version of the SMOS implementation chain will be included in the next cycle CY36R5/CY37R1 and used for the offline monitoring of SMOS data.

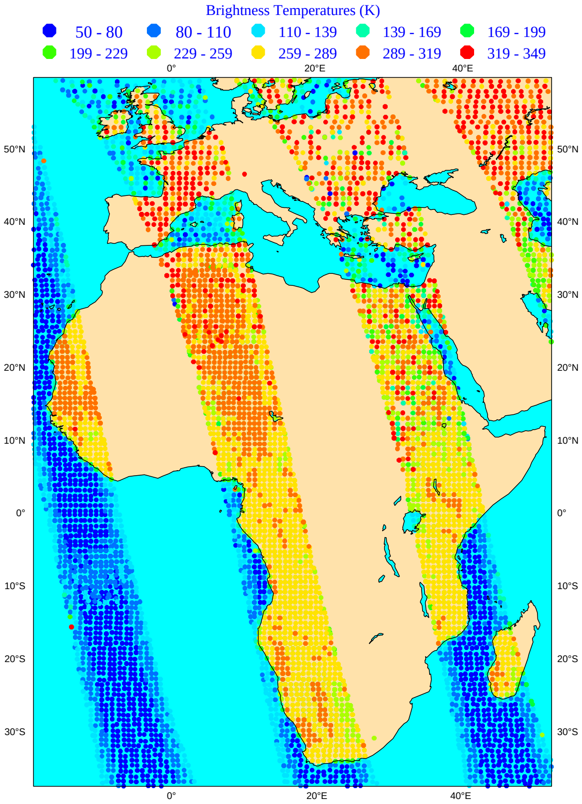


Figure 3: SMOS observations collocated to grid-points in a T255 model grid resolution. Brightness temperatures values are in Kelvin.

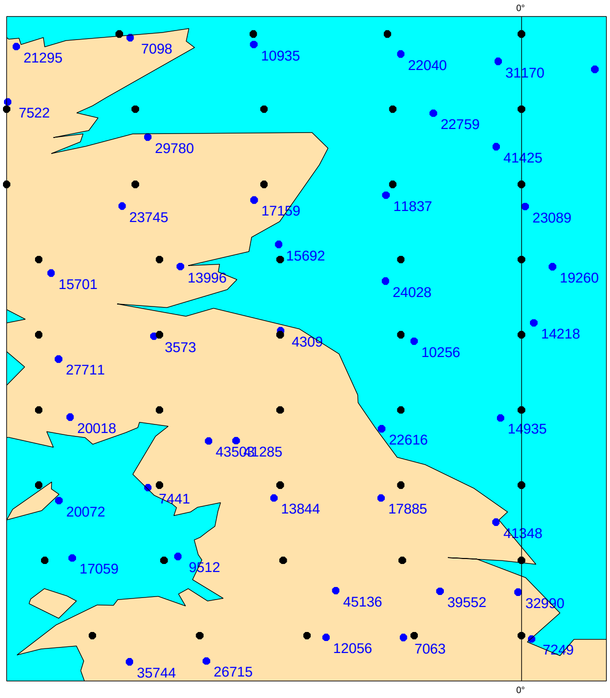


Figure 4: Nearest SMOS observations (blue dots) to grid points (black dots) in a grid of T255 spectral resolution. Values in the figure represent the distance between the model grid point and the nearest SMOS observation point, in meters.

Part III - Offline monitoring suite

Due to memory limitations, SMOS data monitoring is not switched on in the ECMWF operational suite. Instead an experiment in research mode is run which enables the continuous monitoring of SMOS data. The main goal of the SMOS offline monitoring suite is to routinely monitor the data over land and sea as well as to localize the presence of possible temporal or spatial bias which may substantially affect the soil moisture analysis. The offline monitoring suite is composed of the following main steps:

a.- Routine acquisition of data,

b.- Pre-screening of data in BUFR format (see Part I of this report),

c.- Mapping of SMOS BUFR data to internal Observational Data Base (ODB) format for use in the IFS (see section 4 of Milestone 1 Tech Note - Part 2),

d.- Collocation of observations with model fields (see Part II of this report) and computations in model space (forward modelling, first-guess departures),

e.- Production of global maps with statistics (see Global Statistics section of Part III of this report) and publication at the SMOS offline monitoring webpage (http://www.ecmwf.int/research/ESA_projects/SMOS/monitoring/smos_monitor.html)

SMOS brightness temperatures are currently being monitored for a specific number of incidence angles and for the XX and YY polarisations (polarisations in the satellite antenna reference frame). For each grid point, only for those observations which are flagged as active (and then with the distance closest to the grid point) the radiative transfer code CMEM (the Community Microwave Emission Model, see Milestone 1 Tech Note, Part 1) is activated and a model equivalent of this observation is computed. The computed modelled value is then compared to the closest observation value and the first-guess departure (difference between both data values) is stored in the Operational Data Base for further use.

Fig. 5 shows an organigram of the complete SMOS offline monitoring chain with the new structure. Compared to Fig.5 of Milestone 1 Tech Note - Part 2, two big changes can be observed:
Firstly, pre-screening is now carried out simultaneously in multi processors. The reason for this is that ’*presmos*’ jobs in Fig. 1 were taking too long to be incorporated in an operational chain. By distributing the data in several processors and by conducting parallel similar pre-screening steps, the computing time required to run ’*presmos*’ jobs was considerably reduced as to be comparable to that conducted for other satellite data, and thus being compatible with the operational suite. Also, the thinning scheme has been modified in order to have a better control and add more flexibility over the incidence angle of the observations which are monitored (to be addressed in the next contract, -in Technical Note of Phase II, WP1200 at K0+7- currently under evolution).
Secondly, computations in model space are greatly simplified. In particular, global variables allocating a large amount of memory dissapear in this version and the distribution of the observations is now done at the time of mapping observations to model space. In particular, this approach has demonstrated to be an efficient way to collocate observations with model grid points, mainly at high resolution.

All SMOS observations in ODB are screened and a flag is attributed individually to each of them (land, sea, active, passive, etc). Forward modelling is then computed only for those observations which are closest to the grid point. The transition back to observation space is mainly carried out by *smos_update* (see right box of Fig. 5). Finally departures are computed by the ”*hop*” routine and stored in ODB. The resulting ODB database is connected with a statistical package (adapted also for the SMOS requirements) which is able to produce long term statistics about the data and the model simulations in NRT.

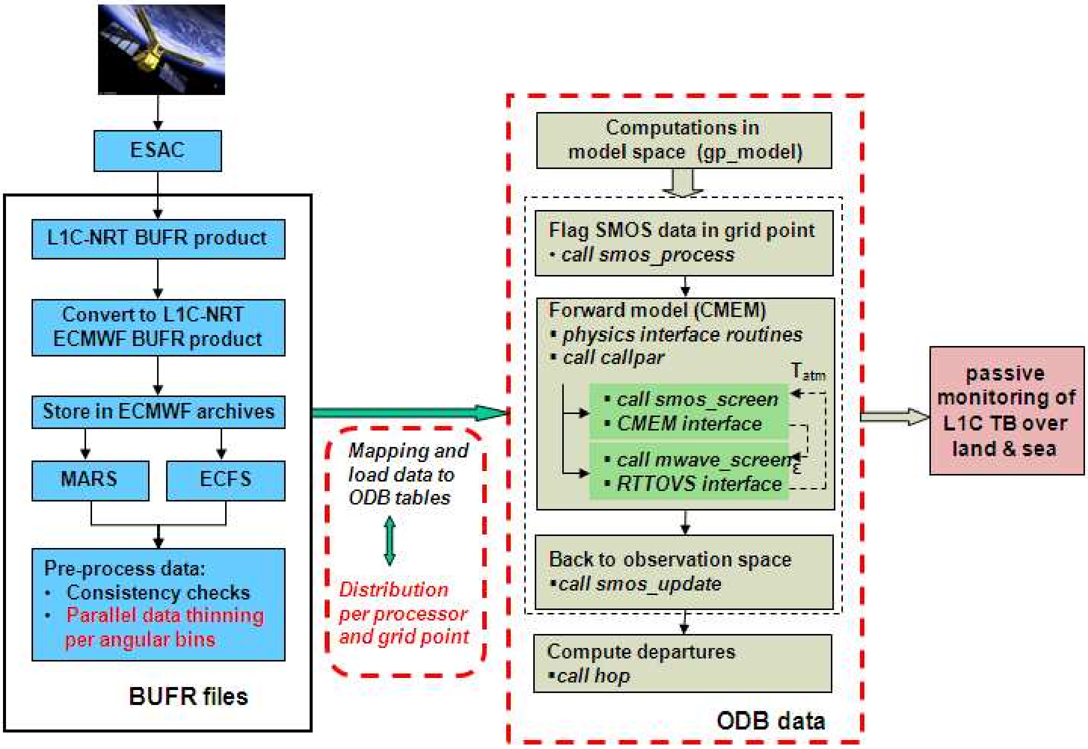


Figure 5: Organigram of the SMOS offline monitoring chain developed at ECMWF.

SMOS offline monitoring webpage

As part of the SMOS monitoring support activities, ECMWF is maintaining and regularly updating an offline webpage where global daily maps of NRT brightness temperatures are shown, sorted by date,incidence angle and polarisation mode, respectively. The web address is the following: ”*http://www.ecmwf.int/research/ESA_projects/SMOS/monitoring/smos_monitor.html*.”
The multi-angular global maps of brightness temperatures are a very interesting product of the SMOS data monitoring implementation, as a simple inspection of time series of this data at global scale makes it possible to observe an evolution and localize possible spatial patterns or angular effects in the data. As example, figures 8, 9 and 10 show the brightness temperatures at 30, 40 and 50 degrees incidence angle, respectively. Left panels correspond to H polarisation, whereas right panels show the V polarisation. Figures are shown for the following days: 1) 28th November 2009 (top figures), 2) 20th December 2009 (middle figures) and 3) 16th January 2010 (bottom figures). Thus, approximately 1 month of difference between each data set is permitted.
Firstly, each figure shows a clear evolution on the quality of the data, from day 1) (top) to day 3) (bottom). The day in November is presented as to be very noisy. This is data received within the two first weeks of the instrument switch on phase. At this stage no calibration was carried out yet, radio-frequency interference was present in many areas, geolocalisation was not accurate, the data processor was not fully operational, etc. In December a major calibration event took place and the difference in the product is quite significant when comparing top with middle figures. Improvements are present almost everywhere. The data is even better the 16*th* of January, although this needs of a closer look-up and it needs of quantitative results to confirm it.
Secondly, since in SMOS a new technology is tested, it is important to check the correct functioning of the

instrument. Days in December and January have an expected behaviour for both polarisations: brightness temperatures values getting colder with the incidence angle for H polarisation and an opposite behaviour for the V polarisation, with both displaying values within an acceptable physical range, as confirmed in the pre-screening phase. This confirms that the novel technique used in SMOS is running relatively well.
Finally, it is also an objective of data monitoring activities reporting on possible spatial or temporal effects on the data: 1/ For both polarisations, at 50 degrees, it is observed a thin stripe away of the main satellite track; 2/ For V polarisation, over oceans, the edges of the satellite track look colder than the inner part, mainly visible at 30 and 40 degrees. These last two apparent problems are due to the fact of including the extended alias free field of view in the data, which is of less quality that the alias free area. 3/ There is still residual RFI over Europe, Middle-East and Asia, which is particularly straightforward to spot when the data looks very ”red” and noisy. However, in January the data look apparently of good quality over the whole America, Australia and South of Africa. As example, Fig.11 uses the data set 3) to reconstruct a whole image of brightness temperatures over South America, where several land features are easy to identify, as is the case of the Amazon and Tocantins rivers and the Andes mountains close to the West coast which present colder brightness temperatures.

Global statistics

As stated in the previous section, the main goal of the SMOS offline monitoring suite is to routinely monitor the data over land and sea. A statistical method and very robust way to identify systematic differences between modelled values and observations is by producing statistics at global scale accounting for several weeks of data. This is routinely being done for many other satellite data at ECMWF and this is the ultimate goal of the SMOS offline monitoring chain. Fig. 6 presents an overview of the sort of statistical products which will be obtained in NRT during the SMOS lifetime with the offline monitoring chain. These statistics are distributed in three main groups:

1. **Time series of area averages**: their purpose is to show time series (at least for the last month of data) of:

    - first guess departures (observations minus first guess),
    - standard deviation of first guess departures,
    - observed brightness temperatures vs. modelled brightness temperatures (first-guess),
    - number of observations used in the monitoring.

    Values shown represent one mean value per 12h cycle averaged at global scale. These figures are shown separately for different incidence angles of the observations and for the XX and YY polarisation modes of the satellite antenna reference frame. By default these plots will produce statistics for the last month of data.

2. **Time-averaged geographical field means**: these plots provide a spatial global perspective of:

    - mean value of all the brightness temperatures observed per grid box (by default the spatial resolution of the grid box is 0.25 degrees for land surfaces and 1 degree for oceans),
    - standard deviation of all the brightness temperatures observed per grid box,
    - mean value of the first-guess departures collected per grid box,
    - standard deviation of the first-guess departures collected per grid box,

These plots average all the values contained from the beginning of a month. They are quite useful to localize areas where RFI is strong but also they show significant correlations with physical land characteristics. More details will be given in Continuous Monitoring Reports. They are also provided separately for different incidence angles of the observations, and for land and sea. Fig 7 shows an example of these plots for the whole month of June 2010. It corresponds to mean averaged values of the observations at 50 degrees incidence angle and at horizontal polarisation.

3. **Hovmoeller zonal mean fields**: these fields have as objective to plot one mean value averaged per pre-defined bands of latitude as a function of time. They make it possible to analyse the geographical-averaged temporal evolution of statistical variables. Thus, punctual problems in the data that could be unnoticed in geographical area average plots, can be easy localized in these ones. Hovmoeller plots are available for the following variables:
   - mean value of observations (by default three bands of longitude of 120 degrees and bands of 2.5 degrees of latitude are used to obtain statistics),
   - number of observations per box,
   - first guess departures per box,
   - standard deviation of all the brightness temperatures observed per box,
   - standard deviation of the the first-guess departures collected per box,

Currently these statistics are being produced and published during a test period before routine production. A following update will produce the same statistics independently per hemispheres, continents and selected areas of interest for cal/val teams. On top of that a new statistical product plotting first-guess departures as a function of the incidence angle, averaged per grid box and containing several weeks of data will be included too.

Acknowledgements

This work is funded under the ESA-ESRIN contract number 20244/07/I-LG and are Part 1,2 and 3 of Milestone-2 Technical Note (MS2TN-P1/2/3). The authors would like to thank Lars Isaksen for his useful advice, Milan Dragosavac and Ioannis Mallas for the operational acquisition and contribution to the processing of the data, Mohamed Dahoui for his help in the generation of long-term statistics with SMOS data and Matthias Drusch for his contribution in the project definition.

SMOS

Time series of area averages

Product updated the 24th of last month

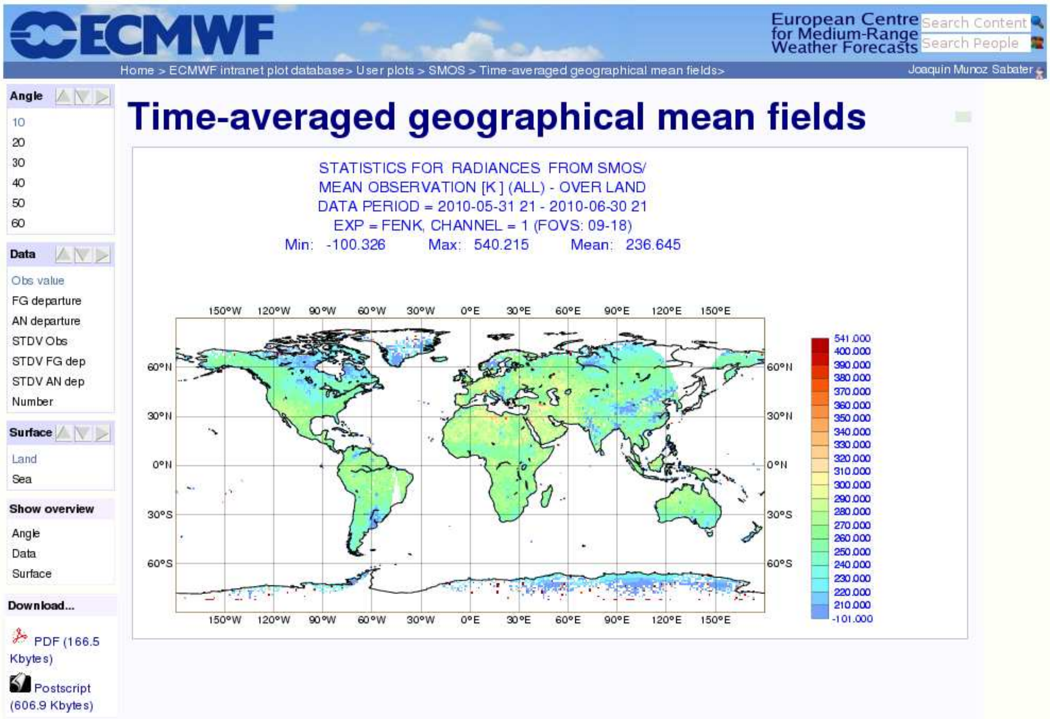
Hovmoeller zonal mean fields

Product updated the 25th of last month

Time-averaged geographical mean fields

Product updated the 25th of last month

Figure 6: Global overview of SMOS statistical products obtained under the offline monitoring suite.



Time-averaged geographical mean fields

STATISTICS FOR RADIANCES FROM SMOS/
MEAN OBSERVATION [K] (ALL) - OVER LAND
DATA PERIOD = 2010-05-31 21 - 2010-06-30 21
EXP = FENK, CHANNEL = 1 (FOVS: 09-18)
Min: -100.326 Max: 540.215 Mean: 236.645

Figure 7: Spatial distribution of time averaged SMOS brightness temperatures for June 2010 at vertical polarisation. Only observations with at an incidence angle around 50 degrees are used in this figure.

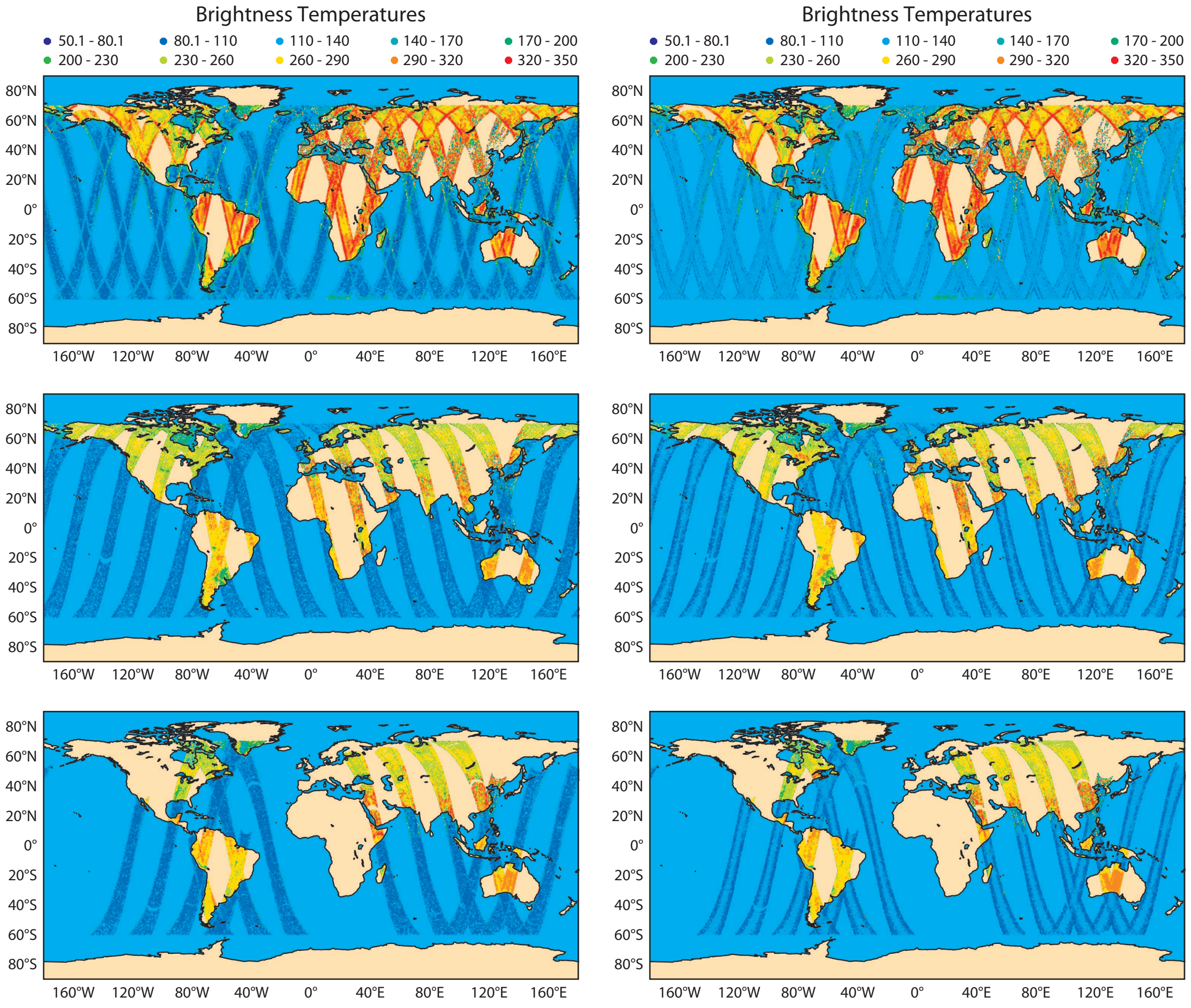


Figure 8: Brightness temperatures for real NRT SMOS data at 30 degrees incidence angle. The left panel corresponds to H polarisation whereas the right panel is V polarisation. Figures on top are for the 28th *November 2009, middle figures correspond to the 20*th *of December 2009 and bottom figures to the 16*th *of January 2010.*

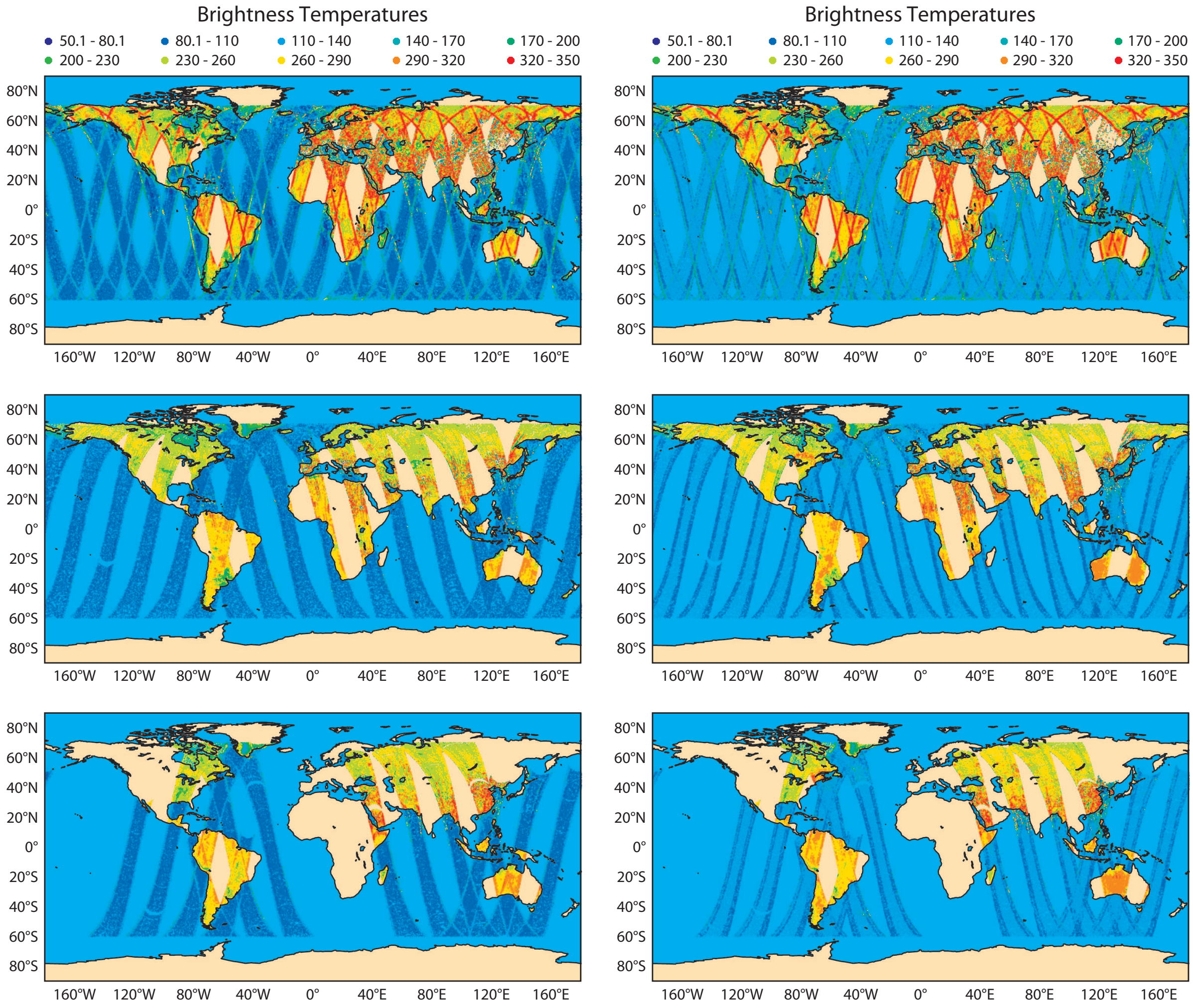


Figure 9: Brightness temperatures for real NRT SMOS data at 40 degrees incidence angle. The left panel corresponds to H polarisation whereas the right panel is V polarisation. Figures on top are for the 28th November 2009, middle figures correspond to the 20th of December 2009 and bottom figures to the 16th of January 2010.

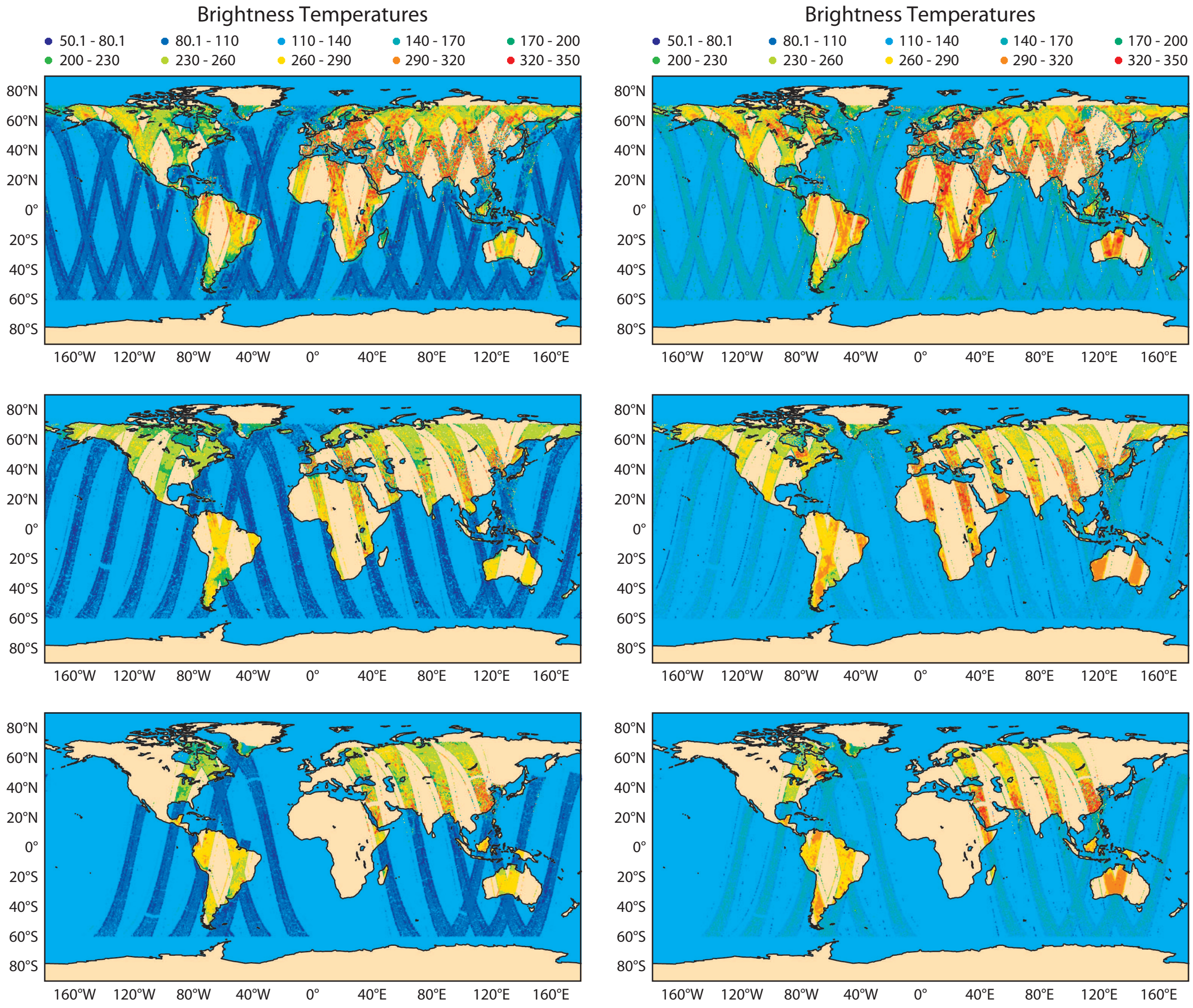


Figure 10: Brightness temperatures for real NRT SMOS data at 50 degrees incidence angle. The left panel corresponds to H polarisation whereas the right panel is V polarisation. Figures on top are for the 28th November 2009, middle figures correspond to the 20th of December 2009 and bottom figures to the 16th of January 2010.

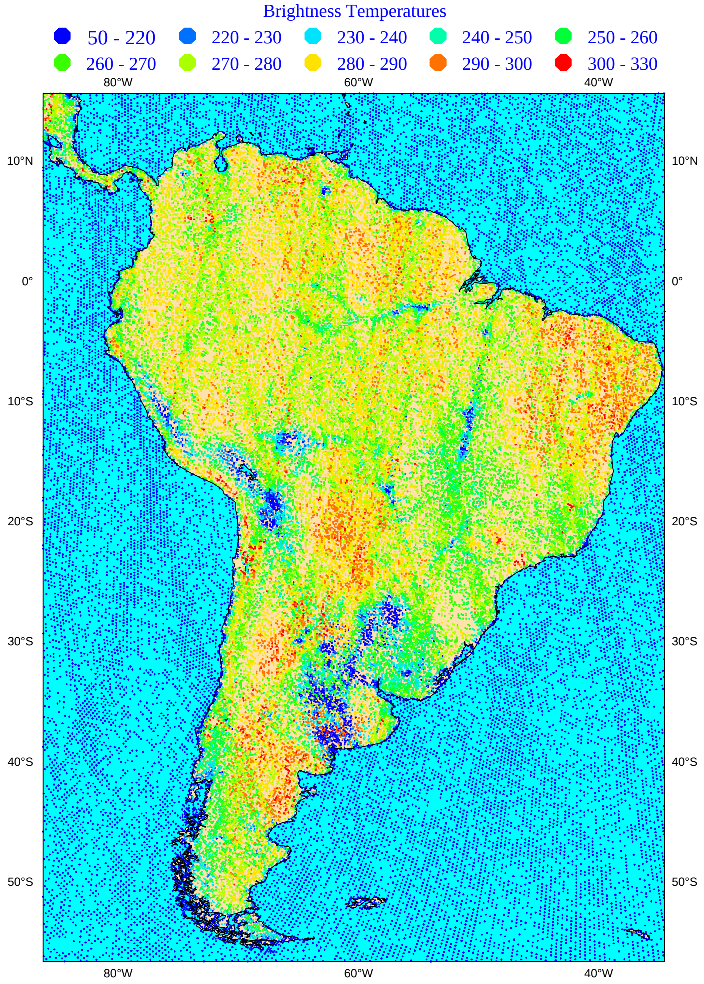
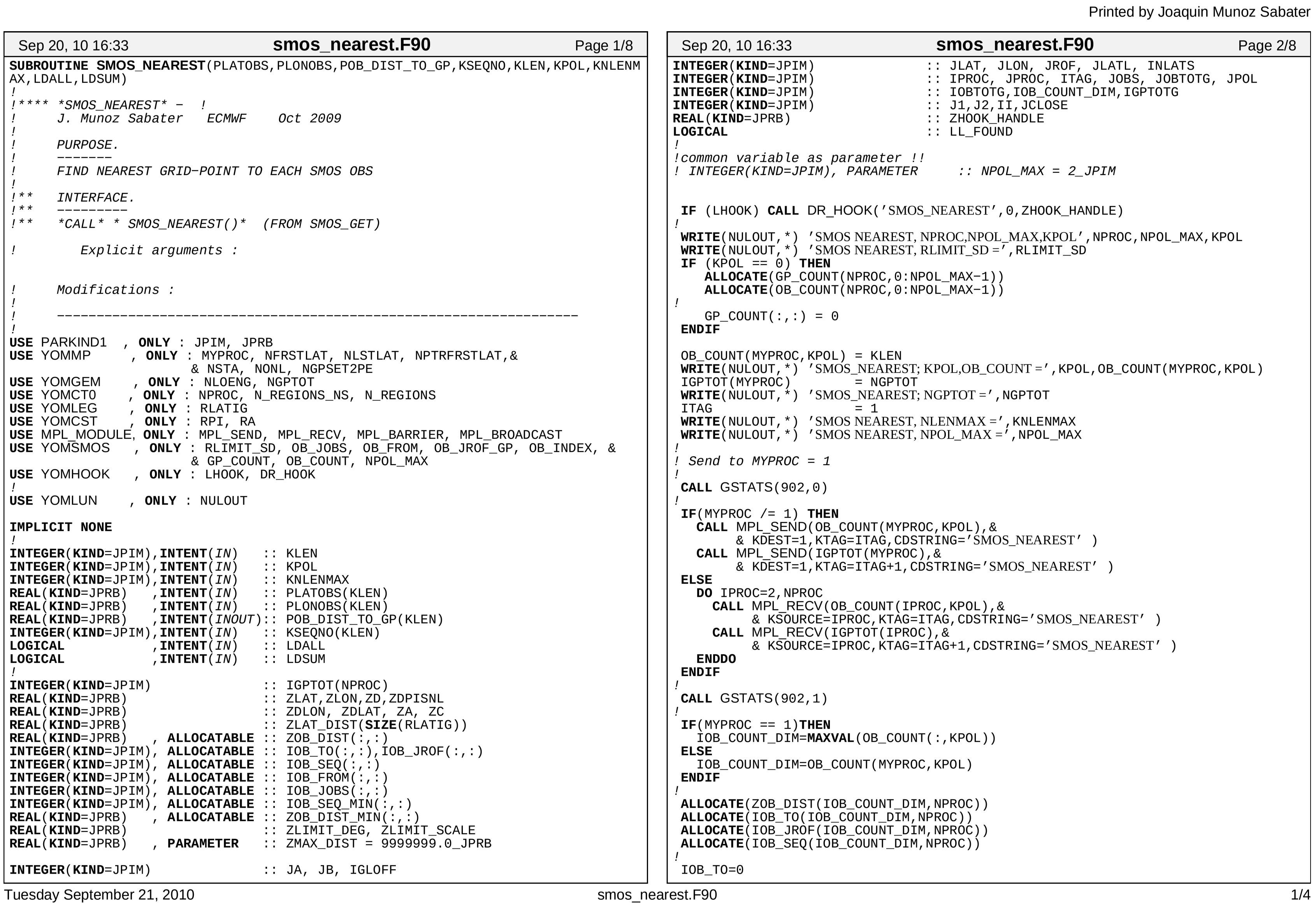


Figure 11: Brightness temperatures over South-America at 40 incidence angle and V polarisation. This figure collects three days of NRT SMOS data, from 18th *to 20*th *December 2009.*

Appendix



Printed by Joaquin Munoz Sabater

Sep 20, 10 16:33 **smos_nearest.F90** Page 1/8

```
SUBROUTINE SMOS_NEAREST(PLATOBS,PLONOBS,POB_DIST_TO_GP,KSEQNO,KLEN,KPOL,KNLENM
AX,LDALL,LDSUM)
!
!**** *SMOS_NEAREST* -  !
!      J. Munoz Sabater   ECMWF    Oct 2009
!
!      PURPOSE.
!      -------
!      FIND NEAREST GRID-POINT TO EACH SMOS OBS
!
!**    INTERFACE.
!**    ---------
!**    *CALL* * SMOS_NEAREST()*  (FROM SMOS_GET)

!         Explicit arguments :


!      Modifications :
!
!      -----------------------------------------------------------------
!
USE PARKIND1  , ONLY : JPIM, JPRB
USE YOMMP     , ONLY : MYPROC, NFRSTLAT, NLSTLAT, NPTRFRSTLAT,&
                     & NSTA, NONL, NGPSET2PE
USE YOMGEM    , ONLY : NLOENG, NGPTOT
USE YOMCT0    , ONLY : NPROC, N_REGIONS_NS, N_REGIONS
USE YOMLEG    , ONLY : RLATIG
USE YOMCST    , ONLY : RPI, RA
USE MPL_MODULE, ONLY : MPL_SEND, MPL_RECV, MPL_BARRIER, MPL_BROADCAST
USE YOMSMOS   , ONLY : RLIMIT_SD, OB_JOBS, OB_FROM, OB_JROF_GP, OB_INDEX, &
                     & GP_COUNT, OB_COUNT, NPOL_MAX
USE YOMHOOK   , ONLY : LHOOK, DR_HOOK
!
USE YOMLUN    , ONLY : NULOUT

IMPLICIT NONE
!
INTEGER(KIND=JPIM),INTENT(IN)   :: KLEN
INTEGER(KIND=JPIM),INTENT(IN)   :: KPOL
INTEGER(KIND=JPIM),INTENT(IN)   :: KNLENMAX
REAL(KIND=JPRB)   ,INTENT(IN)   :: PLATOBS(KLEN)
REAL(KIND=JPRB)   ,INTENT(IN)   :: PLONOBS(KLEN)
REAL(KIND=JPRB)   ,INTENT(INOUT):: POB_DIST_TO_GP(KLEN)
INTEGER(KIND=JPIM),INTENT(IN)   :: KSEQNO(KLEN)
LOGICAL           ,INTENT(IN)   :: LDALL
LOGICAL           ,INTENT(IN)   :: LDSUM
!
INTEGER(KIND=JPIM)              :: IGPTOT(NPROC)
REAL(KIND=JPRB)                 :: ZLAT,ZLON,ZD,ZDPISNL
REAL(KIND=JPRB)                 :: ZDLON, ZDLAT, ZA, ZC
REAL(KIND=JPRB)                 :: ZLAT_DIST(SIZE(RLATIG))
REAL(KIND=JPRB)   , ALLOCATABLE :: ZOB_DIST(:,:)
INTEGER(KIND=JPIM), ALLOCATABLE :: IOB_TO(:,:),IOB_JROF(:,:)
INTEGER(KIND=JPIM), ALLOCATABLE :: IOB_SEQ(:,:)
INTEGER(KIND=JPIM), ALLOCATABLE :: IOB_FROM(:,:)
INTEGER(KIND=JPIM), ALLOCATABLE :: IOB_JOBS(:,:)
INTEGER(KIND=JPIM), ALLOCATABLE :: IOB_SEQ_MIN(:,:)
REAL(KIND=JPRB)   , ALLOCATABLE :: ZOB_DIST_MIN(:,:)
REAL(KIND=JPRB)                 :: ZLIMIT_DEG, ZLIMIT_SCALE
REAL(KIND=JPRB)   , PARAMETER   :: ZMAX_DIST = 9999999.0_JPRB

INTEGER(KIND=JPIM)              :: JA, JB, IGLOFF
```

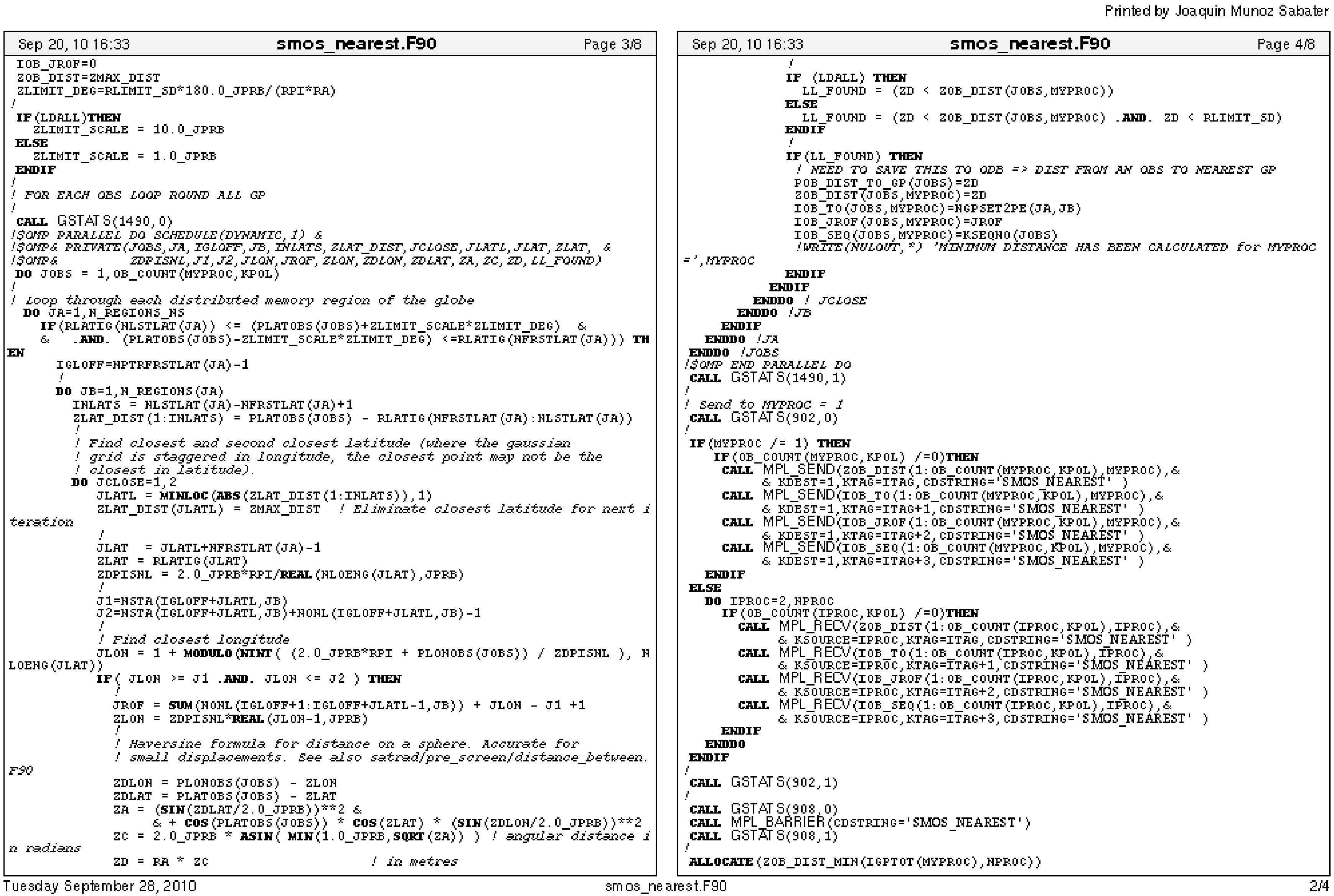
Sep 20, 10 16:33 **smos_nearest.F90** Page 2/8

```
INTEGER(KIND=JPIM)                :: JLAT, JLON, JROF, JLATL, INLATS
INTEGER(KIND=JPIM)                :: IPROC, JPROC, ITAG, JOBS, JOBTOTG, JPOL
INTEGER(KIND=JPIM)                :: IOBTOTG,IOB_COUNT_DIM,IGPTOTG
INTEGER(KIND=JPIM)                :: J1,J2,II,JCLOSE
REAL(KIND=JPRB)                   :: ZHOOK_HANDLE
LOGICAL                           :: LL_FOUND
!
!common variable as parameter !!
! INTEGER(KIND=JPIM), PARAMETER     :: NPOL_MAX = 2_JPIM


 IF (LHOOK) CALL DR_HOOK('SMOS_NEAREST',0,ZHOOK_HANDLE)
!
 WRITE(NULOUT,*) 'SMOS NEAREST, NPROC,NPOL_MAX,KPOL',NPROC,NPOL_MAX,KPOL
 WRITE(NULOUT,*) 'SMOS NEAREST, RLIMIT_SD =',RLIMIT_SD
 IF (KPOL == 0) THEN
    ALLOCATE(GP_COUNT(NPROC,0:NPOL_MAX-1))
    ALLOCATE(OB_COUNT(NPROC,0:NPOL_MAX-1))
!
    GP_COUNT(:,:) = 0
 ENDIF

 OB_COUNT(MYPROC,KPOL) = KLEN
 WRITE(NULOUT,*) 'SMOS_NEAREST; KPOL,OB_COUNT =',KPOL,OB_COUNT(MYPROC,KPOL)
 IGPTOT(MYPROC)         = NGPTOT
 WRITE(NULOUT,*) 'SMOS_NEAREST; NGPTOT =',NGPTOT
 ITAG                   = 1
 WRITE(NULOUT,*) 'SMOS NEAREST, NLENMAX =',KNLENMAX
 WRITE(NULOUT,*) 'SMOS NEAREST, NPOL_MAX =',NPOL_MAX
!
! Send to MYPROC = 1
!
 CALL GSTATS(902,0)
!
 IF(MYPROC /= 1) THEN
   CALL MPL_SEND(OB_COUNT(MYPROC,KPOL),&
        & KDEST=1,KTAG=ITAG,CDSTRING='SMOS_NEAREST' )
   CALL MPL_SEND(IGPTOT(MYPROC),&
        & KDEST=1,KTAG=ITAG+1,CDSTRING='SMOS_NEAREST' )
 ELSE
   DO IPROC=2,NPROC
     CALL MPL_RECV(OB_COUNT(IPROC,KPOL),&
          & KSOURCE=IPROC,KTAG=ITAG,CDSTRING='SMOS_NEAREST' )
     CALL MPL_RECV(IGPTOT(IPROC),&
          & KSOURCE=IPROC,KTAG=ITAG+1,CDSTRING='SMOS_NEAREST' )
   ENDDO
 ENDIF
!
 CALL GSTATS(902,1)
!
 IF(MYPROC == 1)THEN
   IOB_COUNT_DIM=MAXVAL(OB_COUNT(:,KPOL))
 ELSE
   IOB_COUNT_DIM=OB_COUNT(MYPROC,KPOL)
 ENDIF
!
 ALLOCATE(ZOB_DIST(IOB_COUNT_DIM,NPROC))
 ALLOCATE(IOB_TO(IOB_COUNT_DIM,NPROC))
 ALLOCATE(IOB_JROF(IOB_COUNT_DIM,NPROC))
 ALLOCATE(IOB_SEQ(IOB_COUNT_DIM,NPROC))
!
 IOB_TO=0
```

Tuesday September 21, 2010 smos_nearest.F90 1/4



Printed by Joaquin Munoz Sabater

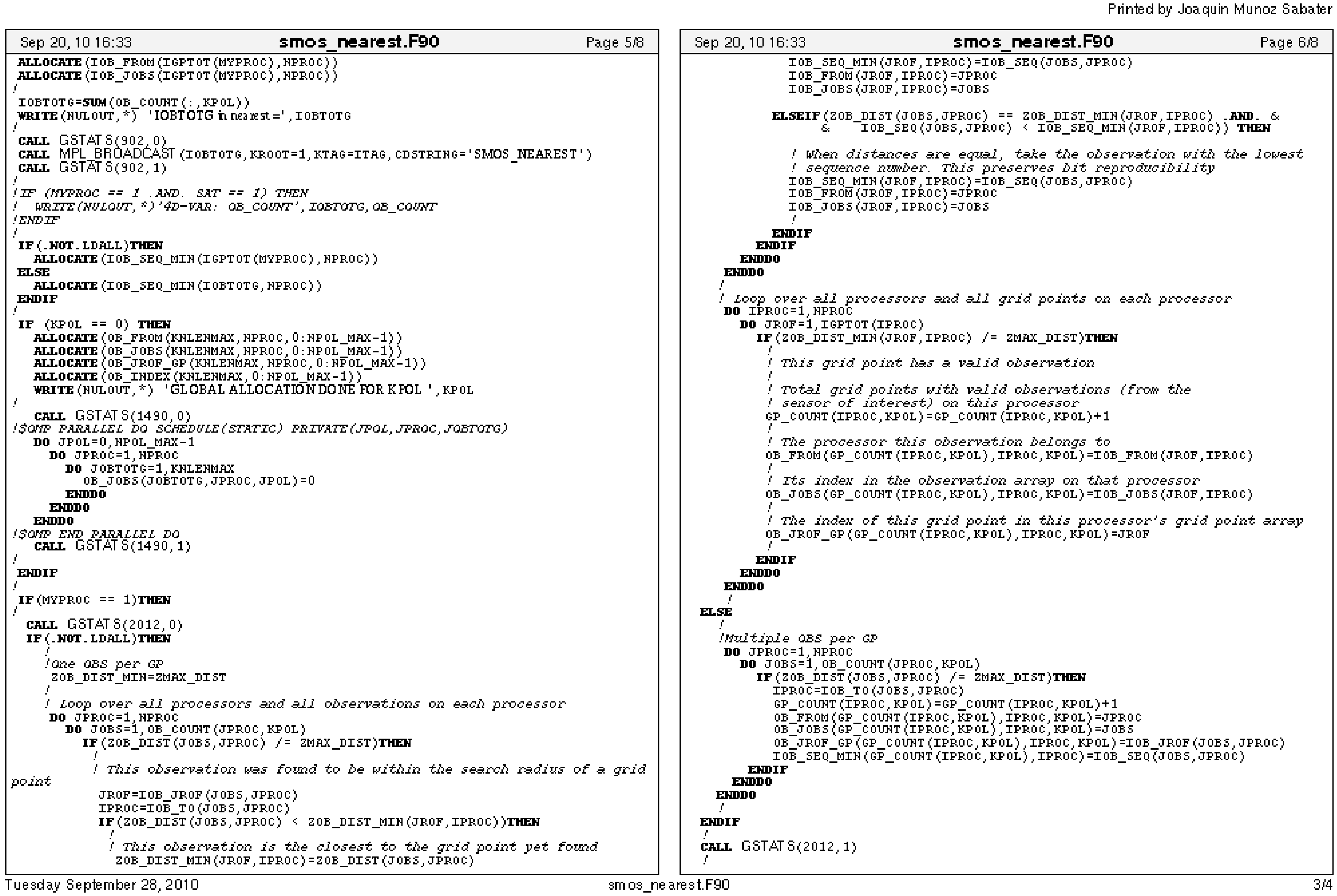
Sep 20, 10 16:33 **smos_nearest.F90** Page 3/8

```
 IOB_JROF=0
 ZOB_DIST=ZMAX_DIST
 ZLIMIT_DEG=RLIMIT_SD*180.0_JPRB/(RPI*RA)
!
 IF(LDALL)THEN
   ZLIMIT_SCALE = 10.0_JPRB
 ELSE
   ZLIMIT_SCALE = 1.0_JPRB
 ENDIF
!
! FOR EACH OBS LOOP ROUND ALL GP
!
 CALL GSTATS(1490,0)
!$OMP PARALLEL DO SCHEDULE(DYNAMIC,1) &
!$OMP& PRIVATE(JOBS,JA,IGLOFF,JB,INLATS,ZLAT_DIST,JCLOSE,JLATL,JLAT,ZLAT,  &
!$OMP&           ZDPISNL,J1,J2,JLON,JROF,ZLON,ZDLON,ZDLAT,ZA,ZC,ZD,LL_FOUND)
 DO JOBS = 1,OB_COUNT(MYPROC,KPOL)
!
! Loop through each distributed memory region of the globe
  DO JA=1,N_REGIONS_NS
    IF(RLATIG(NLSTLAT(JA)) <= (PLATOBS(JOBS)+ZLIMIT_SCALE*ZLIMIT_DEG)  &
    &   .AND. (PLATOBS(JOBS)-ZLIMIT_SCALE*ZLIMIT_DEG) <=RLATIG(NFRSTLAT(JA))) TH
EN
      IGLOFF=NPTRFRSTLAT(JA)-1
      !
      DO JB=1,N_REGIONS(JA)
        INLATS = NLSTLAT(JA)-NFRSTLAT(JA)+1
        ZLAT_DIST(1:INLATS) = PLATOBS(JOBS) - RLATIG(NFRSTLAT(JA):NLSTLAT(JA))
        !
        ! Find closest and second closest latitude (where the gaussian
        ! grid is staggered in longitude, the closest point may not be the
        ! closest in latitude).
        DO JCLOSE=1,2
           JLATL = MINLOC(ABS(ZLAT_DIST(1:INLATS)),1)
           ZLAT_DIST(JLATL) = ZMAX_DIST  ! Eliminate closest latitude for next i
teration
           !
           JLAT  = JLATL+NFRSTLAT(JA)-1
           ZLAT = RLATIG(JLAT)
           ZDPISNL = 2.0_JPRB*RPI/REAL(NLOENG(JLAT),JPRB)
           !
           J1=NSTA(IGLOFF+JLATL,JB)
           J2=NSTA(IGLOFF+JLATL,JB)+NONL(IGLOFF+JLATL,JB)-1
           !
           ! Find closest longitude
           JLON = 1 + MODULO(NINT( (2.0_JPRB*RPI + PLONOBS(JOBS)) / ZDPISNL ), N
LOENG(JLAT))
           IF( JLON >= J1 .AND. JLON <= J2 ) THEN
             !
             JROF = SUM(NONL(IGLOFF+1:IGLOFF+JLATL-1,JB)) + JLON - J1 +1
             ZLON = ZDPISNL*REAL(JLON-1,JPRB)
             !
             ! Haversine formula for distance on a sphere. Accurate for
             ! small displacements. See also satrad/pre_screen/distance_between.
F90
             ZDLON = PLONOBS(JOBS) - ZLON
             ZDLAT = PLATOBS(JOBS) - ZLAT
             ZA = (SIN(ZDLAT/2.0_JPRB))**2 &
                  & + COS(PLATOBS(JOBS)) * COS(ZLAT) * (SIN(ZDLON/2.0_JPRB))**2
             ZC = 2.0_JPRB * ASIN( MIN(1.0_JPRB,SQRT(ZA)) ) ! angular distance i
n radians
             ZD = RA * ZC                           ! in metres
```

Tuesday September 28, 2010 smos_nearest.F90 2/4

Sep 20, 10 16:33 **smos_nearest.F90** Page 4/8

```
             !
             IF (LDALL) THEN
               LL_FOUND = (ZD < ZOB_DIST(JOBS,MYPROC))
             ELSE
               LL_FOUND = (ZD < ZOB_DIST(JOBS,MYPROC) .AND. ZD < RLIMIT_SD)
             ENDIF
             !
             IF(LL_FOUND) THEN
              ! NEED TO SAVE THIS TO ODB => DIST FROM AN OBS TO NEAREST GP
              POB_DIST_TO_GP(JOBS)=ZD
              ZOB_DIST(JOBS,MYPROC)=ZD
              IOB_TO(JOBS,MYPROC)=NGPSET2PE(JA,JB)
              IOB_JROF(JOBS,MYPROC)=JROF
              IOB_SEQ(JOBS,MYPROC)=KSEQNO(JOBS)
              !WRITE(NULOUT,*) 'MINIMUM DISTANCE HAS BEEN CALCULATED for MYPROC
=',MYPROC
             ENDIF
           ENDIF
         ENDDO ! JCLOSE
       ENDDO !JB
     ENDIF
   ENDDO !JA
 ENDDO !JOBS
!$OMP END PARALLEL DO
 CALL GSTATS(1490,1)
!
! Send to MYPROC = 1
 CALL GSTATS(902,0)
!
 IF(MYPROC /= 1) THEN
    IF(OB_COUNT(MYPROC,KPOL) /=0)THEN
     CALL MPL_SEND(ZOB_DIST(1:OB_COUNT(MYPROC,KPOL),MYPROC),&
          & KDEST=1,KTAG=ITAG,CDSTRING='SMOS_NEAREST' )
     CALL MPL_SEND(IOB_TO(1:OB_COUNT(MYPROC,KPOL),MYPROC),&
          & KDEST=1,KTAG=ITAG+1,CDSTRING='SMOS_NEAREST' )
     CALL MPL_SEND(IOB_JROF(1:OB_COUNT(MYPROC,KPOL),MYPROC),&
          & KDEST=1,KTAG=ITAG+2,CDSTRING='SMOS_NEAREST' )
     CALL MPL_SEND(IOB_SEQ(1:OB_COUNT(MYPROC,KPOL),MYPROC),&
          & KDEST=1,KTAG=ITAG+3,CDSTRING='SMOS_NEAREST' )
   ENDIF
 ELSE
   DO IPROC=2,NPROC
     IF(OB_COUNT(IPROC,KPOL) /=0)THEN
       CALL MPL_RECV(ZOB_DIST(1:OB_COUNT(IPROC,KPOL),IPROC),&
            & KSOURCE=IPROC,KTAG=ITAG,CDSTRING='SMOS_NEAREST' )
       CALL MPL_RECV(IOB_TO(1:OB_COUNT(IPROC,KPOL),IPROC),&
            & KSOURCE=IPROC,KTAG=ITAG+1,CDSTRING='SMOS_NEAREST' )
       CALL MPL_RECV(IOB_JROF(1:OB_COUNT(IPROC,KPOL),IPROC),&
            & KSOURCE=IPROC,KTAG=ITAG+2,CDSTRING='SMOS_NEAREST' )
       CALL MPL_RECV(IOB_SEQ(1:OB_COUNT(IPROC,KPOL),IPROC),&
            & KSOURCE=IPROC,KTAG=ITAG+3,CDSTRING='SMOS_NEAREST' )
     ENDIF
   ENDDO
 ENDIF
!
 CALL GSTATS(902,1)
!
 CALL GSTATS(908,0)
 CALL MPL_BARRIER(CDSTRING='SMOS_NEAREST')
 CALL GSTATS(908,1)
!
 ALLOCATE(ZOB_DIST_MIN(IGPTOT(MYPROC),NPROC))
```



Printed by Joaquin Munoz Sabater

Sep 20, 10 16:33 **smos_nearest.F90** Page 5/8

```
 ALLOCATE(IOB_FROM(IGPTOT(MYPROC),NPROC))
 ALLOCATE(IOB_JOBS(IGPTOT(MYPROC),NPROC))
!
 IOBTOTG=SUM(OB_COUNT(:,KPOL))
 WRITE(NULOUT,*) 'IOBTOTG in nearest=',IOBTOTG
!
 CALL GSTATS(902,0)
 CALL MPL_BROADCAST(IOBTOTG,KROOT=1,KTAG=ITAG,CDSTRING='SMOS_NEAREST')
 CALL GSTATS(902,1)
!
!IF (MYPROC == 1 .AND. SAT == 1) THEN
!  WRITE(NULOUT,*)'4D-VAR: OB_COUNT',IOBTOTG,OB_COUNT
!ENDIF
!
 IF(.NOT.LDALL)THEN
   ALLOCATE(IOB_SEQ_MIN(IGPTOT(MYPROC),NPROC))
 ELSE
   ALLOCATE(IOB_SEQ_MIN(IOBTOTG,NPROC))
 ENDIF
!
 IF (KPOL == 0) THEN
   ALLOCATE(OB_FROM(KNLENMAX,NPROC,0:NPOL_MAX-1))
   ALLOCATE(OB_JOBS(KNLENMAX,NPROC,0:NPOL_MAX-1))
   ALLOCATE(OB_JROF_GP(KNLENMAX,NPROC,0:NPOL_MAX-1))
   ALLOCATE(OB_INDEX(KNLENMAX,0:NPOL_MAX-1))
   WRITE(NULOUT,*) 'GLOBAL ALLOCATION DONE FOR KPOL ',KPOL
!
   CALL GSTATS(1490,0)
!$OMP PARALLEL DO SCHEDULE(STATIC) PRIVATE(JPOL,JPROC,JOBTOTG)
   DO JPOL=0,NPOL_MAX-1
     DO JPROC=1,NPROC
       DO JOBTOTG=1,KNLENMAX
         OB_JOBS(JOBTOTG,JPROC,JPOL)=0
       ENDDO
     ENDDO
   ENDDO
!$OMP END PARALLEL DO
   CALL GSTATS(1490,1)
!
 ENDIF
!
 IF(MYPROC == 1)THEN
!
  CALL GSTATS(2012,0)
  IF(.NOT.LDALL)THEN
    !
    !One OBS per GP
     ZOB_DIST_MIN=ZMAX_DIST
    !
    ! Loop over all processors and all observations on each processor
     DO JPROC=1,NPROC
       DO JOBS=1,OB_COUNT(JPROC,KPOL)
         IF(ZOB_DIST(JOBS,JPROC) /= ZMAX_DIST)THEN
           !
           ! This observation was found to be within the search radius of a grid
point
           JROF=IOB_JROF(JOBS,JPROC)
           IPROC=IOB_TO(JOBS,JPROC)
           IF(ZOB_DIST(JOBS,JPROC) < ZOB_DIST_MIN(JROF,IPROC))THEN
             !
             ! This observation is the closest to the grid point yet found
              ZOB_DIST_MIN(JROF,IPROC)=ZOB_DIST(JOBS,JPROC)
```

Tuesday September 28, 2010

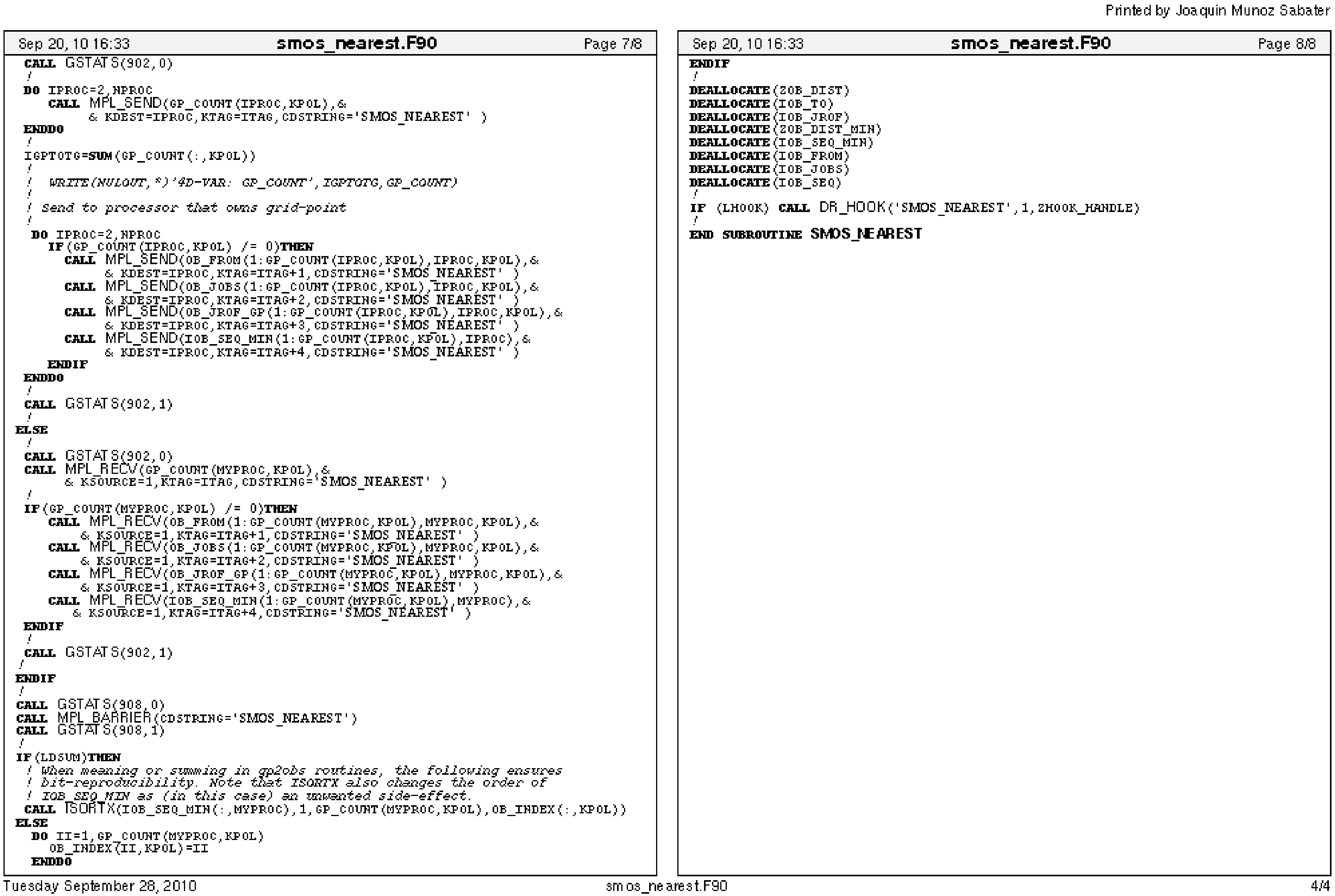
Sep 20, 10 16:33 **smos_nearest.F90** Page 6/8

```
             IOB_SEQ_MIN(JROF,IPROC)=IOB_SEQ(JOBS,JPROC)
             IOB_FROM(JROF,IPROC)=JPROC
             IOB_JOBS(JROF,IPROC)=JOBS

           ELSEIF(ZOB_DIST(JOBS,JPROC) == ZOB_DIST_MIN(JROF,IPROC) .AND. &
              &    IOB_SEQ(JOBS,JPROC) < IOB_SEQ_MIN(JROF,IPROC)) THEN

             ! When distances are equal, take the observation with the lowest
             ! sequence number. This preserves bit reproducibility
             IOB_SEQ_MIN(JROF,IPROC)=IOB_SEQ(JOBS,JPROC)
             IOB_FROM(JROF,IPROC)=JPROC
             IOB_JOBS(JROF,IPROC)=JOBS
             !
           ENDIF
         ENDIF
       ENDDO
     ENDDO
    !
    ! Loop over all processors and all grid points on each processor
     DO IPROC=1,NPROC
       DO JROF=1,IGPTOT(IPROC)
         IF(ZOB_DIST_MIN(JROF,IPROC) /= ZMAX_DIST)THEN
          !
          ! This grid point has a valid observation
          !
          ! Total grid points with valid observations (from the
          ! sensor of interest) on this processor
          GP_COUNT(IPROC,KPOL)=GP_COUNT(IPROC,KPOL)+1
          !
          ! The processor this observation belongs to
          OB_FROM(GP_COUNT(IPROC,KPOL),IPROC,KPOL)=IOB_FROM(JROF,IPROC)
          !
          ! Its index in the observation array on that processor
          OB_JOBS(GP_COUNT(IPROC,KPOL),IPROC,KPOL)=IOB_JOBS(JROF,IPROC)
          !
          ! The index of this grid point in this processor's grid point array
          OB_JROF_GP(GP_COUNT(IPROC,KPOL),IPROC,KPOL)=JROF
          !
         ENDIF
       ENDDO
     ENDDO
      !
  ELSE
    !
    !Multiple OBS per GP
     DO JPROC=1,NPROC
       DO JOBS=1,OB_COUNT(JPROC,KPOL)
         IF(ZOB_DIST(JOBS,JPROC) /= ZMAX_DIST)THEN
           IPROC=IOB_TO(JOBS,JPROC)
           GP_COUNT(IPROC,KPOL)=GP_COUNT(IPROC,KPOL)+1
           OB_FROM(GP_COUNT(IPROC,KPOL),IPROC,KPOL)=JPROC
           OB_JOBS(GP_COUNT(IPROC,KPOL),IPROC,KPOL)=JOBS
           OB_JROF_GP(GP_COUNT(IPROC,KPOL),IPROC,KPOL)=IOB_JROF(JOBS,JPROC)
           IOB_SEQ_MIN(GP_COUNT(IPROC,KPOL),IPROC)=IOB_SEQ(JOBS,JPROC)
        ENDIF
       ENDDO
     ENDDO
    !
  ENDIF
  !
  CALL GSTATS(2012,1)
  !
```

smos_nearest.F90 3/4



Printed by Joaquin Munoz Sabater

Sep 20, 10 16:33 **smos_nearest.F90** Page 7/8

```
  CALL GSTATS(902,0)
  !
  DO IPROC=2,NPROC
     CALL MPL_SEND(GP_COUNT(IPROC,KPOL),&
          & KDEST=IPROC,KTAG=ITAG,CDSTRING='SMOS_NEAREST' )
  ENDDO
  !
  IGPTOTG=SUM(GP_COUNT(:,KPOL))
  !
  !  WRITE(NULOUT,*)'4D-VAR: GP_COUNT',IGPTOTG,GP_COUNT)
  !
  ! Send to processor that owns grid-point
  !
   DO IPROC=2,NPROC
     IF(GP_COUNT(IPROC,KPOL) /= 0)THEN
       CALL MPL_SEND(OB_FROM(1:GP_COUNT(IPROC,KPOL),IPROC,KPOL),&
            & KDEST=IPROC,KTAG=ITAG+1,CDSTRING='SMOS_NEAREST' )
       CALL MPL_SEND(OB_JOBS(1:GP_COUNT(IPROC,KPOL),IPROC,KPOL),&
            & KDEST=IPROC,KTAG=ITAG+2,CDSTRING='SMOS_NEAREST' )
       CALL MPL_SEND(OB_JROF_GP(1:GP_COUNT(IPROC,KPOL),IPROC,KPOL),&
            & KDEST=IPROC,KTAG=ITAG+3,CDSTRING='SMOS_NEAREST' )
       CALL MPL_SEND(IOB_SEQ_MIN(1:GP_COUNT(IPROC,KPOL),IPROC),&
            & KDEST=IPROC,KTAG=ITAG+4,CDSTRING='SMOS_NEAREST' )
     ENDIF
  ENDDO
  !
  CALL GSTATS(902,1)
  !
ELSE
  !
  CALL GSTATS(902,0)
  CALL MPL_RECV(GP_COUNT(MYPROC,KPOL),&
       & KSOURCE=1,KTAG=ITAG,CDSTRING='SMOS_NEAREST' )
  !
  IF(GP_COUNT(MYPROC,KPOL) /= 0)THEN
     CALL MPL_RECV(OB_FROM(1:GP_COUNT(MYPROC,KPOL),MYPROC,KPOL),&
          & KSOURCE=1,KTAG=ITAG+1,CDSTRING='SMOS_NEAREST' )
     CALL MPL_RECV(OB_JOBS(1:GP_COUNT(MYPROC,KPOL),MYPROC,KPOL),&
          & KSOURCE=1,KTAG=ITAG+2,CDSTRING='SMOS_NEAREST' )
     CALL MPL_RECV(OB_JROF_GP(1:GP_COUNT(MYPROC,KPOL),MYPROC,KPOL),&
          & KSOURCE=1,KTAG=ITAG+3,CDSTRING='SMOS_NEAREST' )
     CALL MPL_RECV(IOB_SEQ_MIN(1:GP_COUNT(MYPROC,KPOL),MYPROC),&
         & KSOURCE=1,KTAG=ITAG+4,CDSTRING='SMOS_NEAREST' )
  ENDIF
  !
  CALL GSTATS(902,1)
 !
ENDIF
!
CALL GSTATS(908,0)
CALL MPL_BARRIER(CDSTRING='SMOS_NEAREST')
CALL GSTATS(908,1)
!
IF(LDSUM)THEN
  ! When meaning or summing in gp2obs routines, the following ensures
  ! bit-reproducibility. Note that ISORTX also changes the order of
  ! IOB_SEQ_MIN as (in this case) an unwanted side-effect.
  CALL ISORTX(IOB_SEQ_MIN(:,MYPROC),1,GP_COUNT(MYPROC,KPOL),OB_INDEX(:,KPOL))
ELSE
   DO II=1,GP_COUNT(MYPROC,KPOL)
     OB_INDEX(II,KPOL)=II
   ENDDO
```

Tuesday September 28, 2010 smos_nearest.F90

Sep 20, 10 16:33 **smos_nearest.F90** Page 8/8

```
ENDIF
!
DEALLOCATE(ZOB_DIST)
DEALLOCATE(IOB_TO)
DEALLOCATE(IOB_JROF)
DEALLOCATE(ZOB_DIST_MIN)
DEALLOCATE(IOB_SEQ_MIN)
DEALLOCATE(IOB_FROM)
DEALLOCATE(IOB_JOBS)
DEALLOCATE(IOB_SEQ)
!
IF (LHOOK) CALL DR_HOOK('SMOS_NEAREST',1,ZHOOK_HANDLE)
!
END SUBROUTINE SMOS_NEAREST
```

4/4