Document NWPSAF-KN-TV-008
Version 4.0.02
August 2022

PenWP Test Plan and Test Report

OSI-501-c

Anton Verhoef, Jur Vogelzang, Jeroen Verspeek and Ad Stoffelen

KNMI, De Bilt, the Netherlands

PenWP Test Plan and Test Report

KNMI, De Bilt, the Netherlands

This documentation was developed within the context of the EUMETSAT Ocean and Sea Ice Satellite Application Facility (OSI SAF).

Change record			
Version	**Date**	**Author / changed by**	**Remarks**
1.9	May 2015	Anton Verhoef	First version for PenWP beta release
2.0	Oct 2015	Anton Verhoef	Version for first public PenWP release
2.0.01	Nov 2015	Anton Verhoef	Modified according to DRI comments
2.1	Feb 2017	Anton Verhoef	Version for PenWP 2.1
2.2	May 2018	Anton Verhoef	Version for PenWP 2.2
4.0	Mar 2022	Anton Verhoef	Version for PenWP 4.0 DRR
4.0.01	June 2022	Anton Verhoef	Modified according to DRR comments
4.0.02	Aug 2022	Anton Verhoef	Added HY-2B/C/D integration tests

Contents

1 Introduction

This document is the test plan and test report for the Pencil beam Wind Processor (PenWP) software package. It is set up according to the guidelines of the NWP SAF; see the NWP SAF Development Procedures for Software Deliverables [4]. Parts of the PenWP developments are in fact genscat developments. The tests for genscat modules are also included in this document. Part of the test plan is a traceability matrix to show how requirements as described in the Product Specification [2] are related to the tests in this document.

Most of the module tests described in this document have been developed and performed for OWDP (the OSCAT Wind Data Processor), AWDP (the ASCAT Wind Data Processor) and SDP (the SeaWinds Data Processor) a large part of the code in genscat is shared between PenWP and other OSI SAF wind processors. For this new PenWP version, all module tests have been repeated.

1.1 Aims and scope

The Pencil Beam Wind Processor (PenWP) is a software package written mainly in Fortran 90. The parts and libraries for handling HDF5, NetCDF, and BUFR format data have been partly written in C. The processor can handle data from the SeaWinds (on QuikSCAT or ADEOS-II), OSCAT (on Oceansat-2 or ScatSat-1), HSCAT (on HY-2A/B/C/D) and RapidScat (on the International Space Station) scatterometer instruments. Details of these instruments can be found in [5] and [6], respectively, and on several web sites, see e.g. information on the NASA and ISRO web sites. PenWP is intended to be a generic wind processor for Ku band pencil beam scatterometer data. It will be adapted to handle data from future instruments once they become available.

PenWP generates surface winds based on pencil beam radar backscatter data. It allows performing the ambiguity removal with the Two-dimensional Variational Ambiguity Removal (2DVAR) method and it supports the Multiple Solution Scheme (MSS). The output of PenWP consists of wind vectors which represent surface winds within the ground swath of the scatterometer. Input of PenWP is Normalized Radar Cross Section (NRCS, σ^0) data. These data may be near real-time. The input files of PenWP are in BUFR. Conversion programs are included in the package to convert Hierarchical Data Format (HDF5) data from various instruments to BUFR. Output is written using the SeaWinds BUFR template or the KNMI BUFR template with generic wind section.

Depending on the grid spacing of the BUFR product, PenWP will process the data on 25 km, 50 km or 100 km grid spacing. The SeaWinds/RapidScat HDF5 to BUFR converters can create BUFR data on 25, 50 or 100 km grid spacing by averaging the backscatter data in the level 1b or level 2a input file to the requested gridding. The OSCAT HDF5 to BUFR converters will create BUFR data on 25 or 50 km grid spacing from level 1b data, or on 50 km from level 2a data. HSCAT level 2a input data are currently available on 25 km grid spacing but can also be averaged to a 50 km product. HSCAT level 1b data can also be averaged to a 25 km or 50 km product.

Apart from the scatterometer input data, PenWP needs Numerical Weather Prediction (NWP) model

winds as a first guess for the Ambiguity Removal step. These data need to be provided in GRIB edition 1 or 2 format.

Note that this Test Report only covers the technical and functional software tests and not the validation of scientific improvements in this new PenWP release, like:

- Capability to process data from the Haiyang-2B/C/D satellites.
- The new NSCAT-4DS Geophysical Model Function is included for improved consistency between Ku-band wind retrievals and C-band wind retrievals and buoy winds.
- Sea Surface Temperature dependent adjustments to the backscatter can be applied to reduce wind speed biases.
- Improved and New Quality Control flags have been introduced.
- The backscatter calibrations have been tuned for the generation of climate data records from QuikSCAT, RapidScat and Oceansat-2

The scientific improvements are described and validated in scientific validation reports and journal papers [9], [12], [13], [14], [15].

1.2 Development of PenWP

PenWP is developed within the Ocean and Sea Ice Satellite Application Facility (OSI SAF) project. Originally the wind software packages were Numerical Weather Prediction Satellite Application Facility (NWP SAF) products but from the beginning of the CDOP3 SAF phase, they are OSI SAF products. The packages are still distributed through the NWP SAF web site. The coding is mainly in Fortran 90 and has followed the procedures specified for the NWP SAF. Special attention has been paid on robustness and readability. PenWP will run on Unix or Linux platforms. It is also possible to run on a Windows machine if a Linux environment like the Windows Installer for Ubuntu (Wubi) is installed. Details on the PenWP package and its system requirements can be found in [1], [2] and [3].

1.3 Testing PenWP, traceability matrix

This section describes the Test Plan of the PenWP deliverable. Tests have been carried out in all stages of the development of PenWP. The inversion module is not tested for the PenWP package, because such a test has already been made for the QuikSCAT Data Processor (QDP) development. PenWP contains several methods for Ambiguity Removal within module *ambrem* and its sub modules. Only modules needed for the KNMI 2DVAR scheme for Ambiguity Removal are tested within this project.

Compilation is done on several platforms (operating systems) and with different Fortran 90 compilers. The integration and validation tests were done on both a Linux work station and a Linux server environment.

Section 2 contains the tests for a number of individual modules. In general, modules are tested with the associated test programs. The test programs are located in the folder containing the module under consideration. The output of the test programs is always the standard output (screen) which may be redirected to any test log file or to some output files which are stored in the associated folders. Section 3 describes the PenWP integration test. A test folder containing some sample data is

provided with PenWP and some of the resulting wind fields from these data are shown. Section 4 discusses the validation tests. PenWP winds have been compared with ECMWF model winds in the scope of this report, buoy validations are or will be performed in the scope of the OSI SAF. Section 4 also contains a technical check of the ice screening algorithm. Section 5 describes the portability tests. It contains an overview of platform/operating systems and Fortran and C compilers for which PenWP is supported. Finally, section 6 is devoted to testing the user documentation.

The table below is the traceability matrix. It shows the requirements in the Product Specification (PS, [2]) or Top Level Design (TLD, [3]), how they are tested and where in this report these tests are described.

Require-ment		**Section of PS / TLD**	**Testing method**	**Test plan reference (section)**	**Comment**	**Passed?**
PenWP-001	PenWP generates surface winds	PS 2.1, 3.5, 3.7	Process L2A file in penwp/test folder and inspect output	3.1		Passed
PenWP-002	PenWP generates BUFR output in NOAA format and in KNMI format	PS 2.1, 3.1	Process L2A file in penwp/test folder and inspect output	3.1		Passed
PenWP-003	PenWP generates output in the same WVC spacing as the input data	PS 2.2, 3.2	Process L2A file in penwp/test folder and inspect output	3.1		Passed
PenWP-004	PenWP output contains latitude, longitude and other parameters	PS 2.2	Process L2A file in penwp/test folder and inspect output	3.1		Passed
PenWP-005	PenWP can use either L2A HDF5 data or BUFR data as input (HDF5 after conversion to BUFR)	PS 2.3	Process L2A HDF5 data in penwp/test folder and subsequently reprocess BUFR output	3.1		Passed
PenWP-006	PenWP reads GRIB data containing LSM, SST and forecast winds	PS 2.3	Process L2A file in penwp/test folder and check that a consistent wind field is obtained	3.1		Passed
PenWP-007	PenWP will compile and run on different Linux and Unix platforms	PS 2.4	Compile and run PenWP on different platforms	5		Passed
PenWP-008	L2A backscatter slices are	PS 3.2	Process a few orbits of data,	4.1	When averaging is not done well, a	Passed

Require-ment		Section of PS / TLD	Testing method	Test plan reference (section)	Comment	Passed?
	averaged into a regular WVC swath grid.		check WVC swath grid, and compare output winds to ECMWF background winds.		noisy or inconsistent wind field is obtained. This is reflected in the statistics of scatterometer winds vs. ECMWF.	
PenWP-009	Atmospheric attenuations are computed and stored in output	PS 3.3	Process L2A file in penwp/test folder and inspect output	3.1	Atmospheric attenuations should be in the order of 0.2 to 0.3 dB	Passed
PenWP-010	WVCs with high MLEs must be rejected by Quality Control	PS 3.4	Process L2A file in penwp/test folder and check if QC flag is set for high MLE values	3.1		Passed
PenWP-011	Bayesian ice screening is implemented	PS 3.6	Process a few orbits of L2A data and inspect ice maps	4.2		Passed
PenWP-012	A product monitoring flag is implemented	PS 3.8	Not tested since there are no data with anomalous instrument performance available	-		-
PenWP-013	PenWP can process one orbit within 5 minutes wall clock time.	PS 3.9	Process L2A file in penwp/test folder and check processing time.	3.1		Passed
PenWP-014	Wind accuracy better than 2 m/s in wind component std. dev. with a bias of less than 0.5 m/s in wind speed	TLD 1.1	Process at least one day of data and compare output winds to ECMWF background.	4.1		Passed

Table 1.1 Traceability matrix.

1.4 Test folders

The Test folder of the PenWP software package is located in subdirectory `penwp/tests`. This subdirectory contains several input files for PenWP that are discussed in more detail in section 3. The scripts for executing these tests are located in directory `penwp/execs`. It is recommended to use these scripts (or a modified version) also for normal PenWP operation, as the environment variables needed by PenWP are set in these scripts.

As stated before, most test programs are located in the same directory as the module to be tested. See section 2 for detailed information.

1.5 Conventions

Names of physical quantities (e.g., wind speed components *u* and *v*), modules (e.g. *bufrio_module*), subroutines and identifiers are printed italic.

Names of directories and subdirectories (e.g. `penwp/src)`, files (e.g. `penwp.F90`), and commands (e.g. `penwp -f input`) are printed in Courier. Software systems in general are addressed using the normal font (e.g. PenWP, genscat).

Hyperlinks are printed in blue and underlined (e.g. https://scatterometer.knmi.nl/).

2 Module tests

In this section the various tests to individual modules within PenWP are presented. The tests are listed alphabetically in the module name. Table 2.1 gives an overview of the modules tested, their location and the name of the associated test programs.

Module tests have been included in PenWP if the following conditions were satisfied:

1. The test does not require additional software.
2. The output of the test program is self-explanatory enough to judge the outcome of the test.

Note that in the test program outputs in this section, the number of digits displayed after the decimal point in real values may differ for different compilers. Also the real values may differ slightly due to rounding differences.

Module name	Location	Test program
BFGSMod	`genscat/support/BFGS`	*Test_BFGS*
bufrio_module	`genscat/support/eccodes`	*test_read_BUFR1*
convert	`genscat/support/convert`	*test_convert*
CostFunction	`genscat/ambrem/twodvar`	*Test_SOS*
StrucFunc	`genscat/ambrem/twodvar`	*Test_SOS*
DateTimeMod	`genscat/support/datetime`	*TestDateTimeMod*
ErrorHandler	`genscat/support/ErrorHandler`	*TestErrorHandler*
gribio_module	`genscat/support/eccodes`	*test_read_GRIB1, test_read_GRIB2, test_read_GRIB3*
HDF5Mod	`genscat/support/hdf5`	*TestHDF5*
LunManager	`genscat/support/file`	*TestLunManager*
numerics	`genscat/support/num`	*test_numerics*
SingletonFFT	`genscat/support/singletonfft`	*TestSingleton*
SortMod	`genscat/support/sort`	*SortModTest*

Table 2.1 Overview of module tests.

2.1 Module *BFGSMod*

Directory `genscat/support/BFGS` contains program `Test_BFGS`. This program tests the minimization routine LBFGS and its associated routines in module *BFGSMod*. The routines in *BFGSMod* are slightly modified versions of the freeware routine `LBFGS` and its subroutines. `LBFGS` was written by J. Nocedal, see [7].

Program `Test_BFGS` finds the minimum of the function

$$f(x) = \sum_{i=1}^{100000} (x - i)^4$$

The minimum is the point (1, 2, …, 100000). The search starts at the origin. The typical output is

shown in table 2.2.

```
Program Test_BFGS testing routine LBFGS

 Behavour of cost function:
 Iter         Cost
 -----------------
    0  0.20001E+25
    1  0.19527E+25
    2  0.17724E+25
…
   84  0.29492E-15
   85  0.95608E-16
   86  0.30995E-16

 Routine LBFGS completed succesfully
   Number of iterations                  :      87
   Dimension of problem                  :  100000
   Number of corrections in BFGS update  :       5
   Cost function at start                :  0.20001D+25
   Cost function at end                  :  0.30995D-16
   Precision required                    :  0.10D-19
   Norm of final X                       :  0.18258D+08
   Norm of final G                       :  0.97625D-13
   Minimum and Maximum error in solution :  0.000003  0.000005
   Time needed                           :  0.248  seconds
 Program Test_BFGS completed succesfully.
```

Table 2.2 Output of program Test_BFGS.

2.2 Module *bufrio_module*

Directory `genscat/support/eccodes` contains program *test_read_BUFR1*. This program will read in a small BUFR test file, decode it, encode the data again and write them to an output BUFR file named `testwrite.bufr`. Hence, the program can be used to check the BUFR library. Table 2.3 shows the output generated by *test_read_BUFR1*.

```
open BUFR file
 file name = ./testfile.bufr
 number of BUFR messages in file   =    1
 number of subsets in BUFR message =  361
 latitude  position =   25
 longitude position =   26
 latitude  range:   -3.630   1.260
 longitude range:    2.850   7.690
 open BUFR file for writing
 file name = ./testwrite.bufr
 BUFR file was written successfully
```

Table 2.3 Output of program *test_read_BUFR1*.

2.3 Module *convert*

Directory `genscat/support/convert` contains module *convert.F90*, a number of routines for the conversion of meteorological and geographical quantities. Its associated test program is *test_convert*, and part of its output is listed in table 2.4. Program *test_convert* produces quite a lot of output.

It starts with checking some conversions between different wind vector representations and transformations between different geographical coordinate systems, followed by a check of the

transformation from orbit angles (p,a,rot(z)) to three-dimensional position (x,y,z).

Only the results for p = 0° and 90° are (partly) shown in table 2.4; those for p = 10°, 45°, and 70° are omitted. Program *test_convert* ends with some trigonometric calculations on a sphere.

```
==========================================
u =     5.000000      v =    -7.000000
uv_to_speed, uv_to_dir ====> sp =     8.602325      dir =     324.4623
==========================================
sp =     8.602325      dir =     324.4623
speeddir_to_u, speeddir_to_v ====> u =     5.000002      v =    -6.999999
==========================================
met2uv: sp =     10.00000      dir =     135.0000
met2uv: ====> u =    -7.071068      v =     7.071068
uv2met: u =    -7.071068      v =     7.071068
uv2met: ====> sp =     10.00000      dir =     135.0000
==========================================
lat,lon =     55.00000        5.000000
latlon2xyz: ====> x,y,z =    0.5713938       4.9990479E-02   0.8191521
x,y,z =    0.5713938       4.9990479E-02   0.8191521
xyz2latlon: ====>lat,lon =     55.00000        5.000000
==========================================
        p          a      rot_z          x          y          z         a1     rot_z1         a2     rot_z2
  0.00000  -90.00000    0.00000    0.00000    0.00000   -1.00000  -90.00000  106.16298  270.00000    0.00000
  0.00000  -90.00000   15.00000    0.00000    0.00000   -1.00000  -90.00000  105.59795  270.00000    9.72975
  0.00000  -90.00000   30.00000    0.00000    0.00000   -1.00000  -90.00000  103.95005  270.00000   27.91061
  0.00000  -90.00000   45.00000    0.00000    0.00000   -1.00000  -90.00000  101.35209  270.00000   43.81981
  0.00000  -90.00000   60.00000    0.00000    0.00000   -1.00000  -90.00000   98.00070  270.00000   59.32336
  0.00000  -10.00000    0.00000    0.98481    0.00000   -0.17365  -10.00000    0.00000  190.00000  180.00000
  0.00000  -10.00000   15.00000    0.95125    0.25489   -0.17365  -10.00000   15.00000  190.00000 -164.99998
  0.00000  -10.00000   30.00000    0.85287    0.49240   -0.17365  -10.00000   30.00000  190.00000 -149.99998
…
 90.00000   45.00000   30.00000    0.25882    0.96593    0.00000   74.99999    0.00000  105.00000    0.00000
 90.00000   45.00000   45.00000    0.00000    1.00000    0.00000   90.00000    0.00000   90.00000    0.00000
 90.00000   45.00000   60.00000   -0.25882    0.96593    0.00000   74.99999    0.00000  105.00000    0.00000
 90.00000   90.00000    0.00000    0.00000    1.00000    0.00000   90.00000    0.00000   90.00000    0.00000
 90.00000   90.00000   15.00000   -0.25882    0.96593    0.00000   74.99999    0.00000  105.00000    0.00000
 90.00000   90.00000   30.00000   -0.50000    0.86603    0.00000   59.99999    0.00000  120.00000    0.00000
 90.00000   90.00000   45.00000   -0.70711    0.70711    0.00000   45.00000    0.00000  135.00000    0.00000
 90.00000   90.00000   60.00000   -0.86603    0.50000    0.00000   30.00000    0.00000  149.99998    0.00000
==========================================
latlon1 =     5.000000        5.000000     latlon2 =     6.000000    5.000000
angle distance =    0.9999999       (old function gives     1.000000     )
km distance    =     111.3188
latlon1 =     55.00000        5.000000     latlon2 =     56.00000    5.000000
angle distance =    0.9999999       (old function gives     1.000000     )
km distance    =     111.3188
latlon1 =     85.00000        5.000000     latlon2 =     86.00000    5.000000
angle distance =    0.9999999       (old function gives     1.000000     )
km distance    =     111.3188
==========================================
latlon1 =     5.000000        5.000000     latlon2 =     5.000000    6.000000
angle distance =    0.9961945       (old function gives    0.9961947     )
km distance    =     110.8952
latlon1 =     55.00000        5.000000     latlon2 =     55.00000    6.000000
angle distance =    0.5735716       (old function gives    0.5735765     )
km distance    =     63.84933
latlon1 =     85.00000        5.000000     latlon2 =     85.00000    6.000000
angle distance =    8.7154694E-02   (old function gives    8.7155804E-02 )
km distance    =     9.701961
latlon1 =    -88.50000       -5.000000     latlon2 =    -80.00000    6.000000       SOUTH POLE EXAMPLE
angle distance =     8.532312       (old function gives     8.711980     )
km distance    =     949.8073
==========================================
```

```
Test WVC_Orientation
  WVC1 coordinates (Lam1,Phi1) =   -115.2000       -18.61000
  WVC2 coordinates (Lam2,Phi2) =   -123.6500       -17.52000
  WVC1 orientation Alfa1 =    173.5995        (Should equal 173.5994720)
  WVC2 orientation Alfa2 =    170.9747        (Should equal 170.9747467)
==========================================
angle avg: 10,20:     15.00000
angle avg: -10,20:     5.000000
angle avg: -10,20:     182.0000
angle avg: 12,340:     356.0000
angle avg: 1,359:     0.000000
angle avg: 3,359:     1.000000
angle avg: 1,357:     359.0000
==========================================
ENDREALACC
```

Table 2.4 Output of program *test_convert*

2.4 Modules *CostFunction* and *StrucFunc*

Module *CostFunc.F90* in directory `genscat/ambrem/twodvar` contains the cost function definition of the 2DVAR method. Module *StrucFunc* in the same directory contains the error covariance model of the background field. Large parts of these modules are tested in the single observation solution test implemented in program *Test_SOS*. Table 2.5 lists its output.

The main idea behind this test is that the 2DVAR analysis increment can be calculated analytically in case of one single observation with unit probability. Starting with zero background increment and an observation increment (t_o,l_o) on the 2DVAR grid at the position with indices (1,1), the initial total cost function equals

$$J_t^{init} = \frac{t_o^2 + l_o^2}{\varepsilon^2}$$

where ε stands for the standard deviation of the observation error, which is set to 1.8 in *Test_SOS*. The 2DVAR problem now reduces to a simple optimal interpolation problem. If the standard deviation of the background error is set to the same value as that of the observation error, the final solution has $J_t^{fin} = J_o^{fin} + J_b^{fin} = ½\, J_t^{init}$ with $J_b^{fin} = J_o^{fin}$. This allows construction of the final solution and its gradient, see [8] for more detailed information and a complete description of the 2DVAR method.

Program *Test_SOS* reads the observation increment and the structure function parameters from an input file with default name *Test_SOS.inp,* see below. There are two modes for calculating the Helmholz transformation coefficients, controlled by the variable *Mode* in routine *Set_HelmholzCoefficients* in module *CostFunc.F90*. Mode is a character variable of length 2. Its default value is '*JV*' which stands for sampled continuum (the other value is '*HB*' which stands for periodic boundary conditions but these do not reproduce the correct scaling, see [8] for more details). The program copies the structure function parameters into the *SF*-struct, and the observation increments in the *TwoDvarObs*-struct. The structure function parameters are printed by routine *PrnStrucFuncPars*.

The error covariances are calculated numerically in module *StrucFunc*. For Gaussian structure functions, they can also be calculated analytically. The two methods are compared and the relative precision is printed. In table 2.5 it is 0.00345 for the stream function ψ and 0.0 for the velocity

potential χ, since the latter quantity is identically zero in this example. The precision of the covariances depends on the correlation lengths R_ψ and R_χ.

The total cost function and its gradient is evaluated by routine *JoScat* in module *CostFunction*. From this the cost function components and gradients at the final solution are calculated and checked against their analytical value. The (absolute) precision is printed. Finally, *Test_SOS* checks the packing and unpacking routines of the control vector in both directions.

As stated before, program *Test_SOS* reads its input from an input file. The name (and path) of that file must be given as command line argument of *Test_SOS*. When omitted, the program assumes `Test_SOS.inp` as input file. Table 2.6 gives the structure and contents of the input file, which is in free format. The last decimals of the output values may depend on machine precision.

```
=================================================================
PROGRAM Test_SOS - Single Observation Soluton Check
=================================================================


Input read from file        : Test_SOS.inp
Helmholz coefficients type  : JV
2DVAR:
2DVAR: Parameters inside the StructFunc module:
2DVAR:  Grid size in position domain      :    100000.0       m
2DVAR:  Grid dimensions                   :           32  by           32
2DVAR:  Free edge size                    :            5  points
2DVAR:  Structure function type           : Gaus
2DVAR:  Northern hemisphere:
2DVAR:    Error standard deviation in psi :    1.800000      m/s
2DVAR:    Error standard deviation in chi :    1.800000      m/s
2DVAR:    Rotation/divergence ratio       :    1.000000
2DVAR:    Range parameter for psi         :    300000.0
2DVAR:    Range parameter for chi         :    300000.0
2DVAR:  Tropics:
2DVAR:    Error standard deviation in psi :    2.000000      m/s
2DVAR:    Error standard deviation in chi :    2.000000      m/s
2DVAR:    Rotation/divergence ratio       :   0.1000000
2DVAR:    Range parameter for psi         :    300000.0
2DVAR:    Range parameter for chi         :    300000.0
2DVAR:  Southern hemisphere:
2DVAR:    Error standard deviation in psi :   -1.000000      m/s
2DVAR:    Error standard deviation in chi :    76.00000      m/s
2DVAR:    Rotation/divergence ratio       :    0.000000
2DVAR:    Range parameter for psi         :    1.800000
2DVAR:    Range parameter for chi         :    1.800000

CheckCovMat - checking precision of Covariances
  Relative precision in covariances of psi:   3.3184644E-04
  Relative precision in covariances of chi:   2.7596165E-04

Number of observations      :            1
Number of control variables :         2046

Obs2dvar after initialization:
 i  j  Namb    u    v           Jo           gu           gv
-----------------------------------------------------------
 1  1     1  1.0  0.0  0.00000E+00  0.00000E+00  0.00000E+00

The gradient velocity fields duo and dvo (nonzero components only):
 i  j          duo          dvo
-------------------------------

The cost function of the solution:
  Observation part :    0.000000
  Background part  :    0.000000            precision    0.000000

The background velocity field:
```

```
  u(1,1)          :    0.000000
  Expected value :   0.5000000                 precision   0.5000000
  v(1,1)          :    0.000000
  Expected value :    0.000000                 precision    0.000000

Check background cost function
  Direct calculation from psi and chi   :    0.000000
  Calculation by Jb from control vector :    0.000000            precision     0.000000

Check observation cost function
  Expected value                        :    0.000000
  Calculation by Jo from control vector :    0.000000            precision     0.000000
  Precision in gradients better than    1.9753901E-10

Check packing/unpacking:
  Precision in packing/unpacking of xi    0.000000
  Precision in packing/unpacking of psi    0.000000
  Precision in packing/unpacking of chi    0.000000

Program Test_SOS completed.
================================================================
```

Table 2.5 Output of the single observation solution test.

Record	Item nr.	Name	Meaning
1	1	u0_ini	Initial observation increment in transversal direction (m/s)
1	2	v0_ini	Initial observation increment in longitudinal direction (m/s)
2	1	lparameter	Logical parameter indicating if 2DVAR parameters should be read from file
3	1	TDVParameterFile	Name of 2DVAR parameter file

Table 2.6 Input file for *Test_SOS*.

2.5 Module *DateTimeMod*

Module *DateTimeMod.F90* in directory `genscat/support/datetime` contains general purpose date and time help functions. These are tested by program *TestDateTimeMod*, the output of which is listed in table 2.7.

```
 time-tests
time: 14:22:03.70
time_real        = 51723.70
time_real + 77.2 = 51800.90
time: 14:23:20.90
 time2 is valid
 time1 =
time: 14:22:03.70
 time2 =
time: 14:23:20.90
 time 1 .ne. time2
 date-tests
date: 15-12-1999
 date_int =      19991215
 date_int + 1 =      19991216
date: 16-12-1999
 date2 is valid
 date1 =
date: 15-12-1999
 date2 =
date: 16-12-1999
 date 1 .ne. date2
 date-stepping-tests
 ERROR: The date      21000101  is outside the range
 19000101...20991231, this is not implemented at this time
 ERROR: Julian routines differ from my own routines
date: 31-12-2099
```

```
 next_date_int =    2147483647
date: 01-01-2100
 next_julian_date_int =      21000101
 all OK
 before:
time: 23:59:57.70
date: 31-12-1999
after incrementing by:  5.22 seconds
time: 00:00:02.92
date: 01-01-2000
 valid time
 test of function date2string: 19991231
 test of function date2string_sep: 1999-12-31
 test of function time2string: 235957
 test of function time2string_sep: 23:59:57
 before convert_to_derived_datetime:
date: 28-02-2005
time: 52:00:00.00
 after convert_to_derived_datetime:
date: 02-03-2005
time: 04:00:00.00
 Current date and time:
date: 26-10-2021
time: 09:56:52.60
```

Table 2.7 Output of program *TestDateTimeMod*.

2.6 Module *ErrorHandler*

Module *ErrorHandler.F90* in directory `genscat/support/ErrorHandler` contains routines for handling errors during program execution. The module is tested by program *TestErrorHandler*, the output of which is listed in table 2.8.

```
The Error Handler program_abort routine is set to
return after each error,
in order to try and resume the program...
testing: report_error
an error was reported from within subroutine: dummy_module_name1
error while allocating memory
testing: program_abort (with abort_on_error = .false.)
an error was reported from within subroutine: dummy_module_name2
error while allocating memory
==> trying to resume the program ...
The Error Handler program_abort routine is set to
abort on first error...
testing: program_abort (with abort_on_error = .true.)
an error was reported from within subroutine: dummy_module_name2
error while allocating memory
```

Table 2.8 Output of program *TestErrorHandler*.

2.7 Module *gribio_module*

Module *gribio_module.F90* in directory `genscat/support/eccodes` contains routines for reading and decoding GRIB files. The module is tested by programs *test_read_GRIB1*, *test_read_GRIB2* and *test_read_GRIB3*, the output of which is listed in tables 2.9 to 2.11. The test programs read in two small GRIB files (`testfile.grib` in GRIB edition 1 format and `testfile.grib2` in GRIB edition 2 format) present in this directory and print some of their contents to the standard output. The environment variable `$GRIB_DEFINITION_PATH` needs to be set and has to point to the directory containing GRIB definition tables. These are available in `(...)/genscat/support/eccodes/definitions`.

```
open GRIB edition 1 file
file name = ./testfile.grib
date of grib field =            20031111
time of grib field =                  24
 REALACC(4)
derived date of grib field =    20031112
derived time of grib field =           0

    lat    lon      10u      10v    speed
  54.00   4.00   -4.576    8.006    9.221
  54.00   4.50   -5.143    7.764    9.313
  54.00   5.00   -5.034    7.520    9.050
  54.00   5.50   -4.925    7.276    8.786
  54.50   4.00   -4.849    8.455    9.747
  54.50   4.50   -5.139    8.315    9.775
  54.50   5.00   -5.200    8.426    9.902
  54.50   5.50   -5.261    8.537   10.028
  55.00   4.00   -5.267    8.577   10.065
  55.00   4.50   -5.398    8.454   10.031
  55.00   5.00   -5.416    8.620   10.180
  55.00   5.50   -5.434    8.786   10.330
  55.50   4.00   -5.686    8.699   10.392
  55.50   4.50   -5.657    8.594   10.289
  55.50   5.00   -5.632    8.814   10.459
  55.50   5.50   -5.606    9.034   10.632
 ENDREALACC

open GRIB edition 2 file
file name = ./testfile.grib2
date of grib field =            20031111
time of grib field =                  24

End of tests
```

Table 2.9 Output of program *test_read_GRIB1*.

```
retrieve grib field par_id_t
 REALACC(4)
lat of first gridpoint =   89.142
lat step               =   -1.121
number of lat points   =      160
lon of first gridpoint =    0.000
lon step               =    1.125
number of lon points   =      320

    i    j  field(i,j)
   80  160     302.663
   80  161     302.445
   80  162     302.148
   80  163     301.560
   81  160     301.999
   81  161     302.298
   81  162     301.808
   81  163     301.708
   82  160     302.056
   82  161     302.117
   82  162     301.490
   82  163     301.888
   83  160     302.214
   83  161     302.001
   83  162     301.796
   83  163     302.361
 ENDREALACC
```

Table 2.10 Output of program *test_read_GRIB2*.

```
retrieve grib field par_id_10u
date of grib field =    20031111
time of grib field =          24
 WARNING: latitude dimension of field is too small to contain
 WARNING: the read data; truncating the array !!!!!
 original:  nr_lat_points =           160
 truncated: nr_lat_points =            50
 WARNING: longitude dimension of field is too small to contain
 WARNING: the read data; truncating the array !!!!!
 original:  nr_lon_points =           320
 truncated: nr_lon_points =            50
 REALACC(4)

    i    j  field(i,j)
   48   48      -0.414
   48   49       0.477
   48   50      -0.111
   49   48       3.330
   49   49       2.899
   49   50       3.252
   50   48       3.503
   50   49       2.408
   50   50       3.212
 ENDREALACC
```

Table 2.11 Output of program *test_read_GRIB3*.

2.8 Module HDF5Mod

Module *HDF5Mod.F90* in directory `genscat/support/hdf5` contains routines for reading and writing HDF5 files. It is tested by program *TestHDF5*, the output of which is listed in table 2.12. The test program reads in a small HDF5 file called `deflate.h5` and displays some of its contents. After that, it creates a file called `testfile.h5` and writes some data into it. Its contents can be checked e.g. with the command line utility `h5dump`.

```
Successfully opened file deflate.h5 with f_id        72057594037927936
Successfully opened dataset /Dataset1 with d_id       360287970189639680
Successfully closed dataset with d_id       360287970189639680
Successfully opened group / with g_id       144115188075855872
Successfully opened dataset Dataset1 with d_id       360287970189639681
Number of datapoints of dataset       360287970189639681  is        20000
First data values are:
      0       1       2       3       4       0       1       2       3       4
      0       1       2       3       4       0       1       2       3       4
Successfully closed dataset with d_id       360287970189639681
Successfully closed group with g_id       144115188075855872
Successfully closed file with f_id        72057594037927936

End of file reading tests in TestHDF5

Successfully opened file testfile.h5 with f_id        72057594037927937
Successfully created group Group1 with g_id       144115188075855873
Successfully wrote Attribute Attribute1 in group       144115188075855873
Successfully wrote Dataset Dataset1_int_1d in group       144115188075855873
Successfully wrote Dataset Dataset2_int_2d in group       144115188075855873
Successfully wrote Dataset Dataset3_float_1d in group       144115188075855873
Successfully wrote Dataset Dataset4_float_2d in group       144115188075855873
Successfully wrote Dataset Dataset5_string_1d in group
      144115188075855873
Successfully closed group with g_id       144115188075855873
Successfully closed file with f_id        72057594037927937

End of file writing tests in TestHDF5

A HDF5 file called testfile.h5 was created
You can check its contents e.g. using the h5dump utility
```

```
End of TestHDF5
```

Table 2.12 Output of program *TestHDF5*.

2.9 Module *LunManager*

Module *LunManager.F90* in directory `genscat/support/file` contains routines for file unit management. It is tested by program *TestLunManager*, the output of which is listed in table 2.13.

```
Starting fileunit test program
 ===== lun_manager ======
 fileunit:            31  was not in use !!!
 free_lun returns without freeing any fileunit
 fileunit:            88  was not in the range that is handled
 by this module ! (           30  -            39 )
 free_lun returns without freeing any fileunit
 fileunit:            88  was not in the range that is handled
 by this module ! (           30  -            39 )
 enable_lun returns without enabling any fileunit
 fileunit:            88  was not in the range that is handled
 by this module ! (           30  -            39 )
 disable_lun returns without disabling any fileunit
 fileunit:            21  was not in the range that is handled
 by this module ! (           30  -            39 )
 disable_lun returns without disabling any fileunit
 unit:            31  is used?:   F
 unit:            31  is used?:   T
 start of inspect_luns
  lun             0  is open
  lun             0  has a name: stderr
  lun             5  is open
  lun             5  has a name: stdin
  lun             6  is open
  lun             6  has a name: stdout
  lun            31  is open
  lun            31  has a name: TestLunManager.F90
 end of inspect_luns
 fileunit:            31  is still in use !
 disabling it is only possible if it is not used !
 disable_lun returns without disabling any fileunit
 fileunit:            30  is in use
 fileunit:            31  is in use
 fileunit:            32  is still available
 fileunit:            33  is still available
 fileunit:            34  is still available
 fileunit:            35  is still available
 fileunit:            36  is still available
 fileunit:            37  is still available
 fileunit:            38  is still available
 fileunit:            39  is still available
 fileunit:            21  was not in the range that is handled
 by this module ! (           30  -            39 )
 enable_lun returns without enabling any fileunit
 fileunit:            22  was not in the range that is handled
 by this module ! (           30  -            39 )
 enable_lun returns without enabling any fileunit
```

Table 2.13 Output of program *TestLunManager*.

2.10 Module *Numerics*

Module *numerics.F90* in directory `genscat/support/num` contains routines for checking and handling numerical issues like variable sizes and ranges. These are tested by program *test_numerics*, the output of which is listed in Table 2.14.

```
 Starting numerics test program
 ===== representation tests ======
 REALACC(6)
 r4: digits                24
 r4: epsilon       1.1920929E-07
 r4: huge          3.4028235E+38
 r4: minexponent         -125
 r4: maxexponent          128
 r4: precision              6
 r4: radix                  2
 r4: range                 37
 r4: tiny          1.1754944E-38
 ENDREALACC
 REALACC(12)
 r8: digits                53
 r8: epsilon       2.2204460492503131E-016
 r8: huge          1.7976931348623167E+308
 r8: minexponent        -1021
 r8: maxexponent         1024
 r8: precision             15
 r8: radix                  2
 r8: range                307
 r8: tiny          2.2250738585072010E-308
 ENDREALACC
 ===== numerics tests ======
 int1 =    127
 int2 =    32767
 int4 =    2147483647
 int8 =       9223372036854775807
 huge(int1) =           127
 huge(int2) =         32767
 huge(int4) =    2147483647
 huge(int8) =       9223372036854775807
 REALACC(6)  r4   =    1.7000000E+38  ENDREALACC
 REALACC(12) r8   =    1.7000000000000000E+038  ENDREALACC
 ===== check variable sizes  ======
 Variable sizes are as expected
 ===== detect and print variable sizes ======
var_type nr_of_words range precision
       i           4     9
     i1_           1     2
     i2_           2     4
     i4_           4     9
     i8_           8    18
      dr           4    37         6
      s_           4    37         6
      l_           4    37         6
      r_           4    37         6
     r4_           4    37         6
     r8_           8   307        15
 ===== dB conversion test ======
 REALACC(6)
 input test number:           1.2300001E-04
 converted to dB:             -39.10095
 converted back to a real:    1.2299998E-04
 ENDREALACC
 ===== done ======
```

Table 2.14 Output of program *test_mumerics*.

2.11 Module *SingletonFFT*

Module *SingletonFFT* in directory `genscat/support/singletonfft` contains routines for Fast Fourier Transforms. The associated test program is *TestSingleton*. Part of its output is shown in table 2.15.

```
 ==========================================================================
```

```
PROGRAM TestSingleton
Test of SingletonFFT routines by comparing with analytical FT
==========================================================================

Spreading times grid size in dimension 1:   0.1000000      (should be ~ 0.1)
Spreading times grid size in dimension 2:   0.1000000      (should be ~ 0.1)
==========================================================================

 1D            F O R W A R D          B A C K W A R D
            P r e c i s i o n        P r e c i s i o n
 N1         Real         Imag        Real         Imag
-------------------------------------------------------
 32  0.83631E-06  0.10286E-04  0.11921E-06  0.69247E-07
 34  0.61329E-06  0.78932E-05  0.11921E-06  0.11285E-07
 36  0.94782E-06  0.12215E-04  0.11921E-06  0.11036E-06
 38  0.27877E-06  0.20358E-05  0.17881E-06  0.22604E-07
 40  0.83631E-06  0.12143E-04  0.11921E-06  0.54017E-07
 42  0.44603E-06  0.56252E-05  0.77824E-07  0.92940E-07
 44  0.12900E-06  0.27819E-06  0.17881E-06  0.14948E-06
 46  0.94782E-06  0.13554E-04  0.35763E-06  0.34905E-07
 48  0.94782E-06  0.14143E-04  0.23842E-06  0.12666E-06
 50  0.50178E-06  0.66967E-05  0.17881E-06  0.10431E-06
==========================================================================

 2D              F O R W A R D   F F T         B A C K W A R D   F F T
               P r e c i s i o n               P r e c i s i o n
 N1  N2        Real         Imag   Time        Real         Imag   Time
--------------------------------------------------------------------------
 32  32  0.11995E-05  0.20572E-04 0.0000  0.17881E-06  0.10663E-06 0.0000
 32  34  0.10952E-05  0.18179E-04 0.0001  0.11921E-06  0.63061E-07 0.0000
 32  36  0.12516E-05  0.22501E-04 0.0000  0.11921E-06  0.11339E-06 0.0000
 32  38  0.88658E-06  0.82503E-05 0.0001  0.17881E-06  0.66826E-07 0.0001
 32  40  0.12516E-05  0.22430E-04 0.0000  0.17881E-06  0.95745E-07 0.0000
 32  42  0.99089E-06  0.15911E-04 0.0000  0.11921E-06  0.12151E-06 0.0000
 32  44  0.88658E-06  0.10286E-04 0.0001  0.29802E-06  0.17938E-06 0.0001
 32  46  0.11473E-05  0.23840E-04 0.0001  0.35763E-06  0.63112E-07 0.0001
 32  48  0.12516E-05  0.24430E-04 0.0000  0.27816E-06  0.12973E-06 0.0000
 32  50  0.10430E-05  0.16983E-04 0.0000  0.17881E-06  0.11206E-06 0.0000
 34  32  0.11473E-05  0.18179E-04 0.0001  0.11921E-06  0.78046E-07 0.0001
...
 48  50  0.10952E-05  0.20840E-04 0.0001  0.30120E-06  0.12803E-06 0.0001
 50  32  0.99089E-06  0.16983E-04 0.0000  0.17881E-06  0.11192E-06 0.0000
 50  34  0.83443E-06  0.14590E-04 0.0001  0.17881E-06  0.10692E-06 0.0001
 50  36  0.10430E-05  0.18912E-04 0.0001  0.23842E-06  0.11300E-06 0.0001
 50  38  0.46937E-06  0.47101E-05 0.0001  0.17881E-06  0.10619E-06 0.0001
 50  40  0.93873E-06  0.18840E-04 0.0001  0.35763E-06  0.11030E-06 0.0001
 50  42  0.62582E-06  0.12322E-04 0.0001  0.29802E-06  0.11184E-06 0.0001
 50  44  0.46937E-06  0.66967E-05 0.0001  0.29802E-06  0.14250E-06 0.0001
 50  46  0.99089E-06  0.20251E-04 0.0001  0.23842E-06  0.10202E-06 0.0001
 50  48  0.10430E-05  0.20840E-04 0.0001  0.29802E-06  0.15117E-06 0.0001
 50  50  0.57367E-06  0.13393E-04 0.0001  0.35763E-06  0.11255E-06 0.0001
==========================================================================

Program TestSingleton: Resume
Worst case accuracies

                F O R W A R D          B A C K W A R D
             Real         Imag        Real         Imag
-------------------------------------------------------
1D    0.94782E-06  0.14143E-04  0.35763E-06  0.14948E-06
2D    0.13559E-05  0.28287E-04  0.77486E-06  0.28650E-06

Program TestSingleton: Normal termination.
==========================================================================
```

Table 2.15 Output of program *TestSingleton*

2.12 Module *SortMod*

Module *SortMod* in directory `genscat/support/sort` contains two routines for sorting the wind vector solutions found in the inversion step to their probability. The associated test program is

SortModTest. Its output is shown in table 2.16.

```
Test program for the SortMod module
Unsorted array
10.0  9.0  8.0  7.0  6.0  5.0  4.0  3.0  2.0  1.0
After GetSortIndex
 1.0  2.0  3.0  4.0  5.0  6.0  7.0  8.0  9.0 10.0
Sorted array, after SortWithIndex
 1.0  2.0  3.0  4.0  5.0  6.0  7.0  8.0  9.0 10.0
```

Table 2.16 Output of program *SortModTest*

3 PenWP integration test

Directory `penwp/tests` contains one HDF5 file for testing the PenWP executable. File `S1L2A2012006_12113_12114_2.h5.gz` contains (gzipped) Oceansat-2 OSCAT level 2a data from 6 January 2012, 13:51 to 14:03 UTC with 50 km cell spacing, as obtained from ISRO. The files `ECMWF*.grib` contain the necessary NWP data (SST, land-sea mask and wind forecasts) to perform the NWP collocation step.

The user can test the proper functioning of PenWP using the files in the `penwp/tests` directory. To do this, first create a small file containing a list of NWP files, a land sea mask and forecast winds:

```
ls -1 ../data/lsm_hires.grib > nwpflist

ls -1 ECMWF_* >> nwpflist
```

Note that the '-1' contains the number '1' and not the character 'l'. Then, gunzip the HDF5 file:

```
gunzip -c S1L2A2012006_12113_12114_2.h5.gz > S1L2A2012006_12113_12114_2.h5
```

Convert the level 2a input file to BUFR:

```
../execs/oscat_hdf2bufr -f S1L2A2012006_12113_12114_2.h5 -o oscat.bufr
```

Then run PenWP:

```
../execs/penwp_run_4ds -f oscat.bufr -nwpfl nwpflist -mss -mon -noc -verbosity 1
 -genericws 4
```

The result should be two OSCAT level 2 files in BUFR format, called `oscat_20120106_135109_ocsat2_12113_o_500_ovw_l2.bufr` and `oscat_20120106_135109_ocsat2_12113_o_500_ovw_l2.bufr.genws.`

The first file is in NOAA BUFR format and the second file is in KNMI format with generic wind section.

3.1 OSCAT test data

Figure 3.1 shows the global coverage of the OSCAT test run on 50 km. The colours show the magnitude of the wind speed as indicated by the legend. Figure 3.2 shows detailed wind vector plots over the Atlantic west of Africa, with 50 km cell spacing. In the detail plot, a magenta marker on top of the wind arrow denotes land presence. Black wind arrows indicate that the KNMI Quality Control Flag is set.

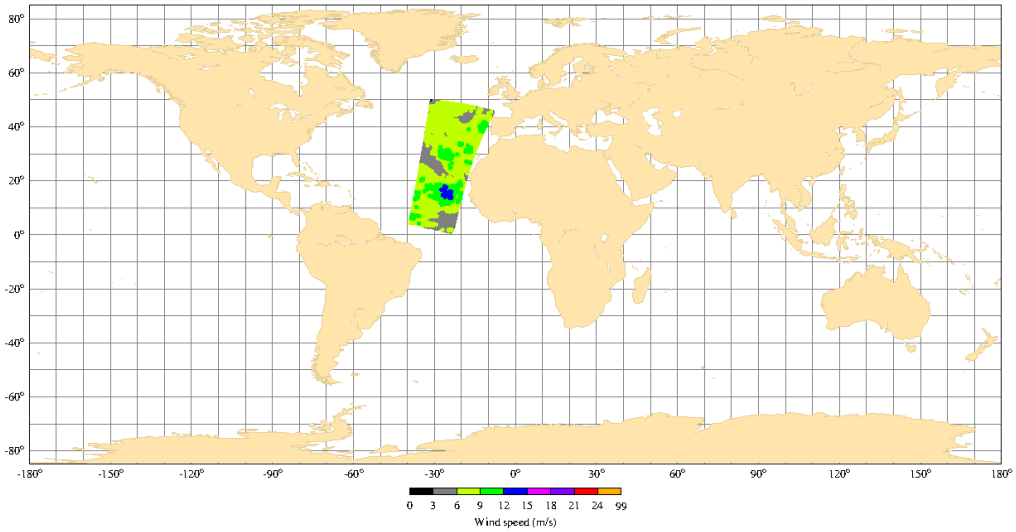

Figure 3.1 Global coverage of the OSCAT test run. Wind speed results for the 50 km product are shown.

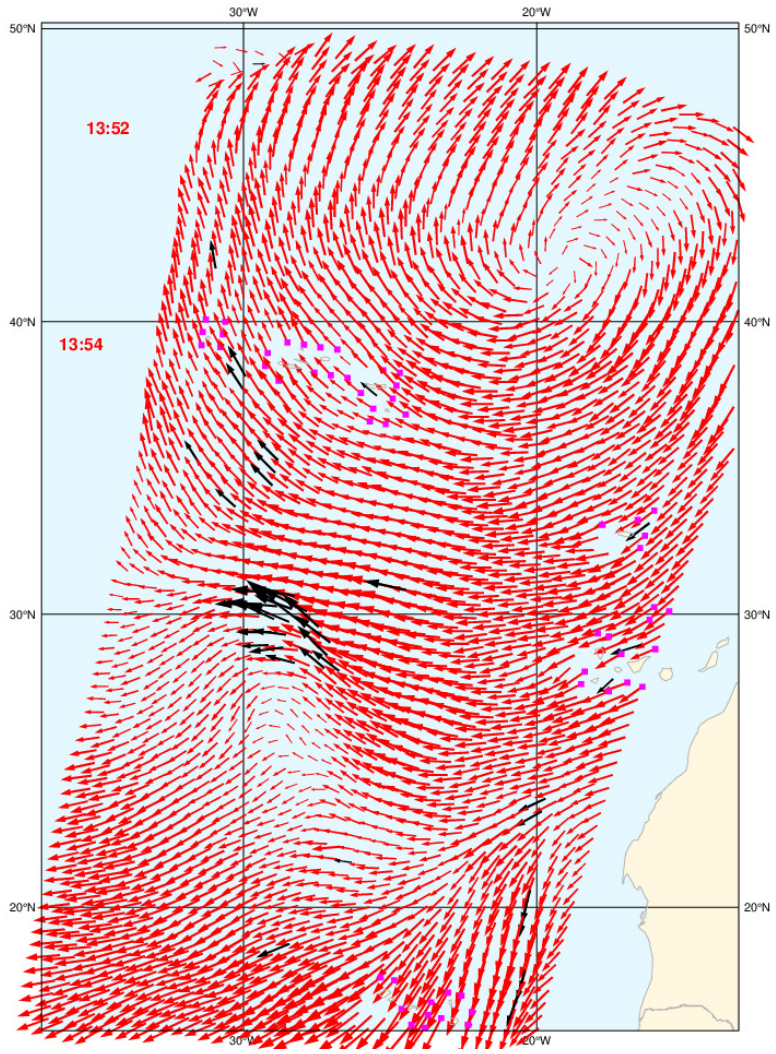

Figure 3.2 Detail plot of the OSCAT test run. Wind vectors for the 50 km product are shown.

The wind vector cells in Figure 3.2 are clearly on a regular swath grid which indicates that the backscatter slice averaging works well. We can simply approximate the WVC spacing from Figure 2. There are 36 wind vectors across track visible spanning a longitude range of ~18°. So the WVC spacing is 18/36=0.50°. The earth circumference at the plot area latitude (20° north) is approximately 40,000×cos(20°)=37,587 km. This leads to a WVC spacing of 0.50/360×37,587=52.2 km, close to the nominal 50 km grid spacing.

The processing times (wall clock time) for the different steps on a Fedora Linux workstation with Intel Xeon 3.6 GHz CPUs and 32 GB memory are:

- 0.5 seconds for L2A HDF5 -> L2A BUFR (oscat_hdf2bufr)
- 2.5 seconds for L2A BUFR -> L2B BUFR (PenWP)

The input file in `penwp/tests` contains 1/8th of a full orbit. This takes approximately 3 seconds to process, hence processing a full orbit would take around 30 seconds maximum. This is well within the maximum allowed processing time of 5 minutes for one orbit. For 25 km products the processing time is approximately 4 times as long, around 2 minutes per orbit, which is still well within the requirement. Hence the wind processing can be done easily in near-real time on an affordable computer system.

Table 3.1 shows one decoded Wind Vector Cell of the resulting output file in NOAA BUFR format and table 3.2 the same WVC in KNMI BUFR format with generic wind section.

1	SATELLITE IDENTIFIER	421.0000	CODE TABLE
2	DIRECTION OF MOTION OF MO	195.0000	deg
3	SATELLITE SENSOR INDICATO	MISSING	CODE TABLE
4	WIND SCATTEROMETER GEOPHY	9.0000	CODE TABLE
5	SOFTWARE IDENTIFICATION (	2208.0000	Numeric
6	CROSS TRACK RESOLUTION	50000.0000	m
7	ALONG TRACK RESOLUTION	50000.0000	m
8	ORBIT NUMBER	12113.0000	Numeric
9	YEAR	2012.0000	a
10	MONTH	1.0000	mon
11	DAY	6.0000	d
12	HOUR	13.0000	h
13	MINUTE	51.0000	min
14	SECOND	9.0000	s
15	LATITUDE (COARSE ACCURACY	46.1700	deg
16	LONGITUDE (COARSE ACCURAC	-11.4400	deg
17	TIME DIFFERENCE QUALIFIER	5.0000	CODE TABLE
18	SECOND	0.0000	s
19	ALONG TRACK ROW NUMBER	MISSING	Numeric
20	CROSS-TRACK CELL NUMBER	6.0000	Numeric
21	SEAWINDS WIND VECTOR CELL	0.0000	FLAG TABLE
22	MODEL WIND DIRECTION AT 1	314.0900	deg
23	MODEL WIND SPEED AT 10M	4.4500	m/s
24	NUMBER OF VECTOR AMBIGUIT	2.0000	Numeric
25	INDEX OF SELECTED WIND VE	1.0000	Numeric
26	TOTAL NUMBER OF SIGMA-0 M	4.0000	Numeric
27	PROBABILITY OF RAIN	MISSING	Numeric
28	SEAWINDS NOF* RAIN INDEX	13.0000	Numeric
29	INTENSITY OF PRECIPITATIO	MISSING	kg m-2 s-1
30	ATTENUATION CORRECTION OF	MISSING	dB
31	WIND SPEED AT 10 M	4.6700	m/s
32	FORMAL UNCERTAINTY IN WIN	MISSING	m/s
33	WIND DIRECTION AT 10 M	327.5000	deg
34	FORMAL UNCERTAINTY IN WIN	1.6100	deg
35	LIKELIHOOD COMPUTED FOR S	0.6650	Numeric
36	WIND SPEED AT 10 M	4.2100	m/s
37	FORMAL UNCERTAINTY IN WIN	MISSING	m/s
38	WIND DIRECTION AT 10 M	180.0000	deg
39	FORMAL UNCERTAINTY IN WIN	2.5700	deg

40	LIKELIHOOD COMPUTED FOR S	0.3350	Numeric
41	WIND SPEED AT 10 M	MISSING	m/s
42	FORMAL UNCERTAINTY IN WIN	MISSING	m/s
43	WIND DIRECTION AT 10 M	MISSING	deg
44	FORMAL UNCERTAINTY IN WIN	MISSING	deg
45	LIKELIHOOD COMPUTED FOR S	MISSING	Numeric
46	WIND SPEED AT 10 M	MISSING	m/s
47	FORMAL UNCERTAINTY IN WIN	MISSING	m/s
48	WIND DIRECTION AT 10 M	MISSING	deg
49	FORMAL UNCERTAINTY IN WIN	MISSING	deg
50	LIKELIHOOD COMPUTED FOR S	MISSING	Numeric
51	ANTENNA POLARISATION	0.0000	CODE TABLE
52	TOTAL NUMBER (WITH RESPEC	58.0000	Numeric
53	BRIGHTNESS TEMPERATURE	124.9000	K
54	STANDARD DEVIATION BRIGHT	9.1000	K
55	ANTENNA POLARISATION	1.0000	CODE TABLE
56	TOTAL NUMBER (WITH RESPEC	42.0000	Numeric
57	BRIGHTNESS TEMPERATURE	198.0000	K
58	STANDARD DEVIATION BRIGHT	12.8000	K
59	NUMBER OF INNER-BEAM SIGM	1.0000	Numeric
60	LATITUDE (COARSE ACCURACY	46.1700	deg
61	LONGITUDE (COARSE ACCURAC	-11.4200	deg
62	ATTENUATION CORRECTION ON	0.1700	dB
63	RADAR LOOK ANGLE	133.0400	deg
64	RADAR INCIDENCE ANGLE	48.9600	deg
65	ANTENNA POLARISATION	0.0000	CODE TABLE
66	SEAWINDS NORMALIZED RADAR	-31.7800	dB
67	KP VARIANCE COEFFICIENT (	1.0040	Numeric
68	KP VARIANCE COEFFICIENT (	0.0000	Numeric
69	KP VARIANCE COEFFICENT (G	-86.2750	dB
70	SEAWINDS SIGMA-0 QUALITY	0.0000	FLAG TABLE
71	SEAWINDS SIGMA-0 MODE	0.0000	FLAG TABLE
72	SEAWINDS LAND/ICE SURFACE	0.0000	FLAG TABLE
73	SIGMA-0 VARIANCE QUALITY	MISSING	Numeric
74	NUMBER OF OUTER-BEAM SIGM	1.0000	Numeric
75	LATITUDE (COARSE ACCURACY	46.1900	deg
76	LONGITUDE (COARSE ACCURAC	-11.4400	deg
77	ATTENUATION CORRECTION ON	0.1700	dB
78	RADAR LOOK ANGLE	154.6300	deg
79	RADAR INCIDENCE ANGLE	57.9000	deg
80	ANTENNA POLARISATION	1.0000	CODE TABLE
81	SEAWINDS NORMALIZED RADAR	-26.2300	dB
82	KP VARIANCE COEFFICIENT (	1.0040	Numeric
83	KP VARIANCE COEFFICIENT (	0.0000	Numeric
84	KP VARIANCE COEFFICENT (G	-77.6000	dB
85	SEAWINDS SIGMA-0 QUALITY	0.0000	FLAG TABLE
86	SEAWINDS SIGMA-0 MODE	8192.0000	FLAG TABLE
87	SEAWINDS LAND/ICE SURFACE	0.0000	FLAG TABLE
88	SIGMA-0 VARIANCE QUALITY	MISSING	Numeric
89	NUMBER OF INNER-BEAM SIGM	1.0000	Numeric
90	LATITUDE (COARSE ACCURACY	46.1700	deg
91	LONGITUDE (COARSE ACCURAC	-11.4400	deg
92	ATTENUATION CORRECTION ON	0.1700	dB
93	RADAR LOOK ANGLE	76.6300	deg
94	RADAR INCIDENCE ANGLE	48.9600	deg
95	ANTENNA POLARISATION	0.0000	CODE TABLE
96	SEAWINDS NORMALIZED RADAR	-33.8500	dB
97	KP VARIANCE COEFFICIENT (	1.0040	Numeric
98	KP VARIANCE COEFFICIENT (	0.0000	Numeric
99	KP VARIANCE COEFFICENT (G	-86.2560	dB
100	SEAWINDS SIGMA-0 QUALITY	0.0000	FLAG TABLE
101	SEAWINDS SIGMA-0 MODE	4096.0000	FLAG TABLE
102	SEAWINDS LAND/ICE SURFACE	0.0000	FLAG TABLE
103	SIGMA-0 VARIANCE QUALITY	MISSING	Numeric
104	NUMBER OF OUTER-BEAM SIGM	1.0000	Numeric
105	LATITUDE (COARSE ACCURACY	46.1700	deg
106	LONGITUDE (COARSE ACCURAC	-11.4500	deg
107	ATTENUATION CORRECTION ON	0.1700	dB
108	RADAR LOOK ANGLE	55.5400	deg
109	RADAR INCIDENCE ANGLE	57.9000	deg
110	ANTENNA POLARISATION	1.0000	CODE TABLE
111	SEAWINDS NORMALIZED RADAR	-32.0100	dB
112	KP VARIANCE COEFFICIENT (	1.0040	Numeric
113	KP VARIANCE COEFFICIENT (	0.0000	Numeric

114	KP VARIANCE COEFFICENT (G	-78.2850	dB
115	SEAWINDS SIGMA-0 QUALITY	0.0000	FLAG TABLE
116	SEAWINDS SIGMA-0 MODE	12288.0000	FLAG TABLE
117	SEAWINDS LAND/ICE SURFACE	0.0000	FLAG TABLE
118	SIGMA-0 VARIANCE QUALITY	MISSING	Numeric

Table 3.1 Wind Vector Cell in NOAA BUFR format

1	SATELLITE IDENTIFIER	421.0000	CODE TABLE
2	DIRECTION OF MOTION OF MO	195.0000	deg
3	SATELLITE SENSOR INDICATO	MISSING	CODE TABLE
4	WIND SCATTEROMETER GEOPHY	9.0000	CODE TABLE
5	SOFTWARE IDENTIFICATION (	13.0000	Numeric
6	CROSS TRACK RESOLUTION	50000.0000	m
7	ALONG TRACK RESOLUTION	50000.0000	m
8	ORBIT NUMBER	12113.0000	Numeric
9	YEAR	2012.0000	a
10	MONTH	1.0000	mon
11	DAY	6.0000	d
12	HOUR	13.0000	h
13	MINUTE	51.0000	min
14	SECOND	9.0000	s
15	LATITUDE (COARSE ACCURACY	46.1700	deg
16	LONGITUDE (COARSE ACCURAC	-11.4400	deg
17	TIME DIFFERENCE QUALIFIER	5.0000	CODE TABLE
18	SECOND	0.0000	s
19	ALONG TRACK ROW NUMBER	MISSING	Numeric
20	CROSS-TRACK CELL NUMBER	6.0000	Numeric
21	TOTAL NUMBER OF SIGMA-0 M	4.0000	Numeric
22	PROBABILITY OF RAIN	MISSING	Numeric
23	SEAWINDS NOF* RAIN INDEX	13.0000	Numeric
24	INTENSITY OF PRECIPITATIO	MISSING	kg m-2 s-1
25	ATTENUATION CORRECTION OF	MISSING	dB
26	ANTENNA POLARISATION	0.0000	CODE TABLE
27	TOTAL NUMBER (WITH RESPEC	58.0000	Numeric
28	BRIGHTNESS TEMPERATURE	124.9000	K
29	STANDARD DEVIATION BRIGHT	9.1000	K
30	ANTENNA POLARISATION	1.0000	CODE TABLE
31	TOTAL NUMBER (WITH RESPEC	42.0000	Numeric
32	BRIGHTNESS TEMPERATURE	198.0000	K
33	STANDARD DEVIATION BRIGHT	12.8000	K
34	NUMBER OF INNER-BEAM SIGM	1.0000	Numeric
35	LATITUDE (COARSE ACCURACY	46.1700	deg
36	LONGITUDE (COARSE ACCURAC	-11.4200	deg
37	ATTENUATION CORRECTION ON	0.1700	dB
38	RADAR LOOK ANGLE	133.0400	deg
39	RADAR INCIDENCE ANGLE	48.9600	deg
40	ANTENNA POLARISATION	0.0000	CODE TABLE
41	SEAWINDS NORMALIZED RADAR	-31.7800	dB
42	KP VARIANCE COEFFICIENT (	1.0040	Numeric
43	KP VARIANCE COEFFICIENT (	0.0000	Numeric
44	KP VARIANCE COEFFICENT (G	-86.2750	dB
45	SEAWINDS SIGMA-0 QUALITY	0.0000	FLAG TABLE
46	SEAWINDS SIGMA-0 MODE	0.0000	FLAG TABLE
47	SEAWINDS LAND/ICE SURFACE	0.0000	FLAG TABLE
48	SIGMA-0 VARIANCE QUALITY	MISSING	Numeric
49	NUMBER OF OUTER-BEAM SIGM	1.0000	Numeric
50	LATITUDE (COARSE ACCURACY	46.1900	deg
51	LONGITUDE (COARSE ACCURAC	-11.4400	deg
52	ATTENUATION CORRECTION ON	0.1700	dB
53	RADAR LOOK ANGLE	154.6300	deg
54	RADAR INCIDENCE ANGLE	57.9000	deg
55	ANTENNA POLARISATION	1.0000	CODE TABLE
56	SEAWINDS NORMALIZED RADAR	-26.2300	dB
57	KP VARIANCE COEFFICIENT (	1.0040	Numeric
58	KP VARIANCE COEFFICIENT (	0.0000	Numeric
59	KP VARIANCE COEFFICENT (G	-77.6000	dB
60	SEAWINDS SIGMA-0 QUALITY	0.0000	FLAG TABLE
61	SEAWINDS SIGMA-0 MODE	8192.0000	FLAG TABLE
62	SEAWINDS LAND/ICE SURFACE	0.0000	FLAG TABLE
63	SIGMA-0 VARIANCE QUALITY	MISSING	Numeric
64	NUMBER OF INNER-BEAM SIGM	1.0000	Numeric

65	LATITUDE (COARSE ACCURACY	46.1700	deg
66	LONGITUDE (COARSE ACCURAC	-11.4400	deg
67	ATTENUATION CORRECTION ON	0.1700	dB
68	RADAR LOOK ANGLE	76.6300	deg
69	RADAR INCIDENCE ANGLE	48.9600	deg
70	ANTENNA POLARISATION	0.0000	CODE TABLE
71	SEAWINDS NORMALIZED RADAR	-33.8500	dB
72	KP VARIANCE COEFFICIENT (	1.0040	Numeric
73	KP VARIANCE COEFFICIENT (	0.0000	Numeric
74	KP VARIANCE COEFFICENT (G	-86.2560	dB
75	SEAWINDS SIGMA-0 QUALITY	0.0000	FLAG TABLE
76	SEAWINDS SIGMA-0 MODE	4096.0000	FLAG TABLE
77	SEAWINDS LAND/ICE SURFACE	0.0000	FLAG TABLE
78	SIGMA-0 VARIANCE QUALITY	MISSING	Numeric
79	NUMBER OF OUTER-BEAM SIGM	1.0000	Numeric
80	LATITUDE (COARSE ACCURACY	46.1700	deg
81	LONGITUDE (COARSE ACCURAC	-11.4500	deg
82	ATTENUATION CORRECTION ON	0.1700	dB
83	RADAR LOOK ANGLE	55.5400	deg
84	RADAR INCIDENCE ANGLE	57.9000	deg
85	ANTENNA POLARISATION	1.0000	CODE TABLE
86	SEAWINDS NORMALIZED RADAR	-32.0100	dB
87	KP VARIANCE COEFFICIENT (	1.0040	Numeric
88	KP VARIANCE COEFFICIENT (	0.0000	Numeric
89	KP VARIANCE COEFFICENT (G	-78.2850	dB
90	SEAWINDS SIGMA-0 QUALITY	0.0000	FLAG TABLE
91	SEAWINDS SIGMA-0 MODE	12288.0000	FLAG TABLE
92	SEAWINDS LAND/ICE SURFACE	0.0000	FLAG TABLE
93	SIGMA-0 VARIANCE QUALITY	MISSING	Numeric
94	SOFTWARE IDENTIFICATION (	2208.0000	Numeric
95	GENERATING APPLICATION	91.0000	CODE TABLE
96	MODEL WIND SPEED AT 10M	4.4500	m/s
97	MODEL WIND DIRECTION AT 1	314.0900	deg
98	ICE PROBABILITY	MISSING	Numeric
99	ICE AGE ("A" PARAMETER)	MISSING	dB
100	WIND VECTOR CELL QUALITY	0.0000	FLAG TABLE
101	NUMBER OF VECTOR AMBIGUIT	2.0000	Numeric
102	INDEX OF SELECTED WIND VE	1.0000	Numeric
103	DELAYED DESCRIPTOR REPLIC	4.0000	Numeric
104	WIND SPEED AT 10 M	4.6700	m/s
105	WIND DIRECTION AT 10 M	327.5000	deg
106	BACKSCATTER DISTANCE	-1.6000	Numeric
107	LIKELIHOOD COMPUTED FOR S	-0.1770	Numeric
108	WIND SPEED AT 10 M	4.2100	m/s
109	WIND DIRECTION AT 10 M	180.0000	deg
110	BACKSCATTER DISTANCE	-2.6000	Numeric
111	LIKELIHOOD COMPUTED FOR S	-0.4750	Numeric
112	WIND SPEED AT 10 M	MISSING	m/s
113	WIND DIRECTION AT 10 M	MISSING	deg
114	BACKSCATTER DISTANCE	MISSING	Numeric
115	LIKELIHOOD COMPUTED FOR S	MISSING	Numeric
116	WIND SPEED AT 10 M	MISSING	m/s
117	WIND DIRECTION AT 10 M	MISSING	deg
118	BACKSCATTER DISTANCE	MISSING	Numeric
119	LIKELIHOOD COMPUTED FOR S	MISSING	Numeric

Table 3.2 Wind Vector Cell in KNMI BUFR format with generic wind section

From the plots and tables in this section it is clear that:

- Output can be provided in two BUFR formats.
- The Wind Vector Cell spacing is 50 km, see fields 6 and 7 in the BUFR outputs and the computation based on Figure 3.2 above.
- The output contains latitude, longitude, time, orbit and node numbers, NWP background wind vector, WVC quality flag, and information on the radar backscatter including σ^0 and K_p data.
- A consistent wind field is obtained which proves that both HDF5 and GRIB data are read

successfully.

- The atmospheric attenuations are present in the BUFR output (fields 62, 77, 92 and 107 in the NOAA BUFR format).

The test was re-run with the BUFR output file as input and this results in a new output file with the same wind information. Hence, it is clear that PenWP accepts BUFR data as input as well as HDF5.

Table 3.3 shows what happens when the MLE value exceeds the threshold for Quality Control. The MLE of the fourth wind solution (the selected one by ambiguity removal) is contained in field 44 and has a value of 40.31. This is above the threshold value of 14.4 corresponding to wind speeds close to 8 m/s. The Wind Vector Cell Quality (field 21) has an integer value of 1028, i.e., Fortran bits 10 and 2 are set, corresponding to the flags for KNMI Quality Control and Rain.

21	SEAWINDS WIND VECTOR CELL	1028.0000	FLAG TABLE
22	MODEL WIND DIRECTION AT 1	152.4700	deg
23	MODEL WIND SPEED AT 10M	6.4400	m/s
24	NUMBER OF VECTOR AMBIGUIT	4.0000	Numeric
25	INDEX OF SELECTED WIND VE	3.0000	Numeric
26	TOTAL NUMBER OF SIGMA-0 M	4.0000	Numeric
27	PROBABILITY OF RAIN	MISSING	Numeric
28	SEAWINDS NOF* RAIN INDEX	13.0000	Numeric
29	INTENSITY OF PRECIPITATIO	MISSING	kg m-2 s-1
30	ATTENUATION CORRECTION OF	MISSING	dB
31	WIND SPEED AT 10 M	7.2600	m/s
32	FORMAL UNCERTAINTY IN WIN	MISSING	m/s
33	WIND DIRECTION AT 10 M	292.5000	deg
34	FORMAL UNCERTAINTY IN WIN	2.1800	deg
35	LIKELIHOOD COMPUTED FOR S	0.9980	Numeric
36	WIND SPEED AT 10 M	8.6100	m/s
37	FORMAL UNCERTAINTY IN WIN	MISSING	m/s
38	WIND DIRECTION AT 10 M	117.5000	deg
39	FORMAL UNCERTAINTY IN WIN	10.9100	deg
40	LIKELIHOOD COMPUTED FOR S	0.0020	Numeric
41	WIND SPEED AT 10 M	8.8900	m/s
42	FORMAL UNCERTAINTY IN WIN	MISSING	m/s
43	WIND DIRECTION AT 10 M	150.0000	deg
44	FORMAL UNCERTAINTY IN WIN	40.3100	deg
45	LIKELIHOOD COMPUTED FOR S	0.0000	Numeric
46	WIND SPEED AT 10 M	7.9600	m/s
47	FORMAL UNCERTAINTY IN WIN	MISSING	m/s
48	WIND DIRECTION AT 10 M	0.0000	deg
49	FORMAL UNCERTAINTY IN WIN	40.3800	deg
50	LIKELIHOOD COMPUTED FOR S	0.0000	Numeric

Table 3.3 Part of Wind Vector Cell in NOAA BUFR format, rejected by Quality Control

3.2 HY-2B, HY-2C and HY-2D test data

One of the new features in PenWP version 4.0 is the ability to process data from the Haiyang 2B/2C/2D satellites. In this section basic tests are presented to show that the software package is able to read in level 1b data from these new platforms and will process the data into wind data in BUFR format. For the scientific validation of these winds we refer to [9]. For the tests in this section, input files from the near-real time OSI SAF processing were taken, together with ECMWF wind forecast GRIB files.

Test for **HY-2B** input data, first convert level 1b HDF5 file into BUFR format:

```
penwp/execs/hscat_l1b_bufr -f
H2B_OPER_SCA_L1B_EG_20220809T054034_20220809T073000_19017_dps_20.h5 -o tmp.bufr
```

The program converts the HDF5 data and reports that HY-2B data are found:

```
hscat_l1b_bufr
hscat_l1b_bufr - input data characteristics
hscat_l1b_bufr - data is from HY-2B
hscat_l1b_bufr
hscat_l1b_bufr - NSOAS processor version                10
hscat_l1b_bufr - start orbit number                  19016
hscat_l1b_bufr - end orbit number                    19018
hscat_l1b_bufr - number of frames                    13000
hscat_l1b_bufr
hscat_l1b_bufr - output data characteristics
hscat_l1b_bufr - cell separation (pixel size)        25000
hscat_l1b_bufr - time increment per row          3.859506180105948
hscat_l1b_bufr - nr. of data rows                     1772
hscat_l1b_bufr - nr. of cells per row                   76
hscat_l1b_bufr - leftmost cell containing HH            10
hscat_l1b_bufr - rightmost cell containing HH           67
hscat_l1b_bufr
penwp: write_bufr_file - file written: tmp.bufr
```

Subsequently, the PenWP wind processor is used to compute winds for the temporary file tmp.bufr:

```
penwp/execs/penwp_run_4ds -f tmp.bufr -nwpfl nwpflist -noc -mss -sstcor -verbosity 1
```

This results in a HY-2B BUFR output file of non-zero size:

```
-rw-r--r-- 11285304 Aug  9 16:05 hscat_20220809_053815_hy_2b__19016_o_250_ovw_l2.bufr
```

Test for **HY-2C** input data, first convert level 1b HDF5 file into BUFR format:

```
penwp/execs/hscat_l1b_bufr -f
H2C_OPER_SCA_L1B_EG_20220809T045540_20220809T064445_09452_dps_20.h5 -o tmp.bufr
```

The program will convert the HDF5 file into BUFR and reports that HY-2C data are found:

```
hscat_l1b_bufr
hscat_l1b_bufr - input data characteristics
hscat_l1b_bufr - data is from HY-2C
hscat_l1b_bufr
hscat_l1b_bufr - NSOAS processor version                10
hscat_l1b_bufr - start orbit number                   9451
hscat_l1b_bufr - end orbit number                     9453
hscat_l1b_bufr - number of frames                    13000
hscat_l1b_bufr
hscat_l1b_bufr - output data characteristics
hscat_l1b_bufr - cell separation (pixel size)        25000
hscat_l1b_bufr - time increment per row          3.846048263684521
hscat_l1b_bufr - nr. of data rows                     1775
hscat_l1b_bufr - nr. of cells per row                   76
hscat_l1b_bufr - leftmost cell containing HH            10
hscat_l1b_bufr - rightmost cell containing HH           66
hscat_l1b_bufr
penwp: write_bufr_file - file written: tmp.bufr
```

Subsequently, the PenWP wind processor is used to compute winds for the temporary file tmp.bufr:

```
penwp/execs/penwp_run_4ds -f tmp.bufr -nwpfl nwpflist -noc -mss -sstcor -verbosity 1
```

This results in a HY-2C BUFR output file of non-zero size:

```
-rw-r--r-- 11277393 Aug  9 16:36 hscat_20220809_045320_hy_2c__09451_o_250_ovw_l2.bufr
```

Test for **HY-2D** input data, first convert level 1b HDF5 file into BUFR format:

```
penwp/execs/hscat_l1b_bufr -f
H2D_OPER_SCA_L1B_EG_20220809T042914_20220809T061820_06062_dps_22.h5 -o tmp.bufr
```

The program will convert the HDF5 file into BUFR and reports that HY-2D data are found:

```
 hscat_l1b_bufr
 hscat_l1b_bufr - input data characteristics
 hscat_l1b_bufr - data is from HY-2D
 hscat_l1b_bufr
 hscat_l1b_bufr - NSOAS processor version                10
 hscat_l1b_bufr - start orbit number                   6061
 hscat_l1b_bufr - end orbit number                     6063
 hscat_l1b_bufr - number of frames                    13000
 hscat_l1b_bufr
 hscat_l1b_bufr - output data characteristics
 hscat_l1b_bufr - cell separation (pixel size)         25000
 hscat_l1b_bufr - time increment per row           3.843817754262200
 hscat_l1b_bufr - nr. of data rows                      1775
 hscat_l1b_bufr - nr. of cells per row                    76
 hscat_l1b_bufr - leftmost cell containing HH             10
 hscat_l1b_bufr - rightmost cell containing HH            66
 hscat_l1b_bufr
 penwp: write_bufr_file - file written: tmp.bufr
```

Subsequently, the PenWP wind processor is used to compute winds for the temporary file tmp.bufr:

```
penwp/execs/penwp_run_4ds -f tmp.bufr -nwpfl nwpflist -noc -mss -sstcor -verbosity 1
```

This results in a HY-2D BUFR output file of non-zero size:

```
-rw-r--r-- 11688131 Aug  9 16:40 hscat_20220809_042656_hy_2d__06061_o_250_ovw_l2.bufr
```

On top of that, a test was done to check the proper handling of corrupted input data. In case of a corrupted input file the software should terminate with a proper error message and without unwanted effects like segmentation faults. In any case, no output file should be produced. To construct a corrupted input file, the Linux '`head`' command was used on a valid file.

```
head -5 H2B_OPER_SCA_L1B_EG_20220809T054034_20220809T073000_19017_dps_20.h5 > tmp.h5
```

This results in a very small, truncated file.

```
-rw-r--r--  2205 Aug  9 16:41 tmp.h5
```

Run the program to try to convert this input file into BUFR format:

```
penwp/execs/hscat_l1b_bufr -f tmp.h5 -o tmp.bufr
```

The program will show error messages and terminate without producing output:

```
HDF5-DIAG: Error detected in HDF5 (1.12.0) thread 0:
  #000: H5F.c line 793 in H5Fopen(): unable to open file
    major: File accessibility
    minor: Unable to open file
  #001: H5VLcallback.c line 3500 in H5VL_file_open(): open failed
    major: Virtual Object Layer
    minor: Can't open object
  #002: H5VLcallback.c line 3465 in H5VL__file_open(): open failed
    major: Virtual Object Layer
    minor: Can't open object
  #003: H5VLnative_file.c line 100 in H5VL__native_file_open(): unable to open file
    major: File accessibility
    minor: Unable to open file
  #004: H5Fint.c line 1707 in H5F_open(): unable to read superblock
    major: File accessibility
    minor: Read failed
  #005: H5Fsuper.c line 621 in H5F__super_read(): truncated file: eof = 2205, sblock-
>base_addr = 0, stored_eof = 55948608
    major: File accessibility
    minor: File has been truncated
 hscat_l1b_bufr - error in h5f_open
Warning: ieee_inexact is signaling
FORTRAN STOP
```

4 Validation tests

There are several methods to validate scatterometer winds. Scatterometer winds are routinely compared with NWP data and in situ buoy winds in the OSI SAF project. See https://scatterometer.knmi.nl/osisaf/ for more information. In the scope of this Test Report, we show the results of a validation study of PenWP winds versus model wind forecasts from the ECMWF model. The correct implementation of the ice screening algorithm is demonstrated in section 4.2.

4.1 PenWP winds versus ECMWF winds

We compared the Oceansat-2 OSCAT 50 km winds from PenWP with ECMWF forecast winds from the ERA5 reanalysis (+3 to +21 hours forecasts from the 06 UTC and 18 UTC runs). The OSCAT data are level 1b data version 1.3 from ISRO from 9 and 10 February 2012 (29 orbits), reprocessed with PenWP.

Figure 4.1 shows the collocations of the OSCAT and ECMWF winds. Contoured histograms are shown for wind speed, wind direction and *u* and *v* wind components and after rejection of Quality Controlled (KNMI QC flagged) wind vectors. The ECMWF winds are stress-equivalent 10m winds. In the wind direction plots, only those wind vectors where the model wind speed is at least 4 m/s are taken into account. The bin sizes for the histograms are 0.5 m/s for wind speed, *u* and *v*, and 2.5° for wind direction.

From the contour plots it is clear that biases are generally low. We obtain wind component standard deviations of 1.12 m/s in *u* and 1.06 m/s in *v* directions. This is comparable to the values we found for Haiyang-2B compared to ECMWF winds from the operational model [9]: approximately 1.20 m/s in *u* and 1.14 m/s in *v* for the 25-km product and approximately 1.05 m/s in *u* and 0.99 m/s in *v* for the 50-km product in the same period of the year.

These numbers can be compared to the accuracy requirements for all OSI SAF scatterometer wind products stated in the OSI SAF Product Requirements Document [4]: 'Better than 2 m/s in wind component std. dev. with a bias of less than 0.5 m/s in wind speed on a monthly basis'. Hence the bias and wind component standard deviations for the PenWP winds are well within the general OSI SAF product requirements [11]. The reason for not using buoy data for verification is that we need at least 2 to 3 months of data to gather enough statistics. This is normally done in Operational Readiness Reviews for OSI SAF wind products. ECMWF comparisons can be gathered with only one day of data and give a good indication of the quality of the retrieved winds, i.e., if the winds are within. OSI SAF requirements when comparing with ECMWF winds, they will also be within OSI SAF requirements when comparing with buoy winds.

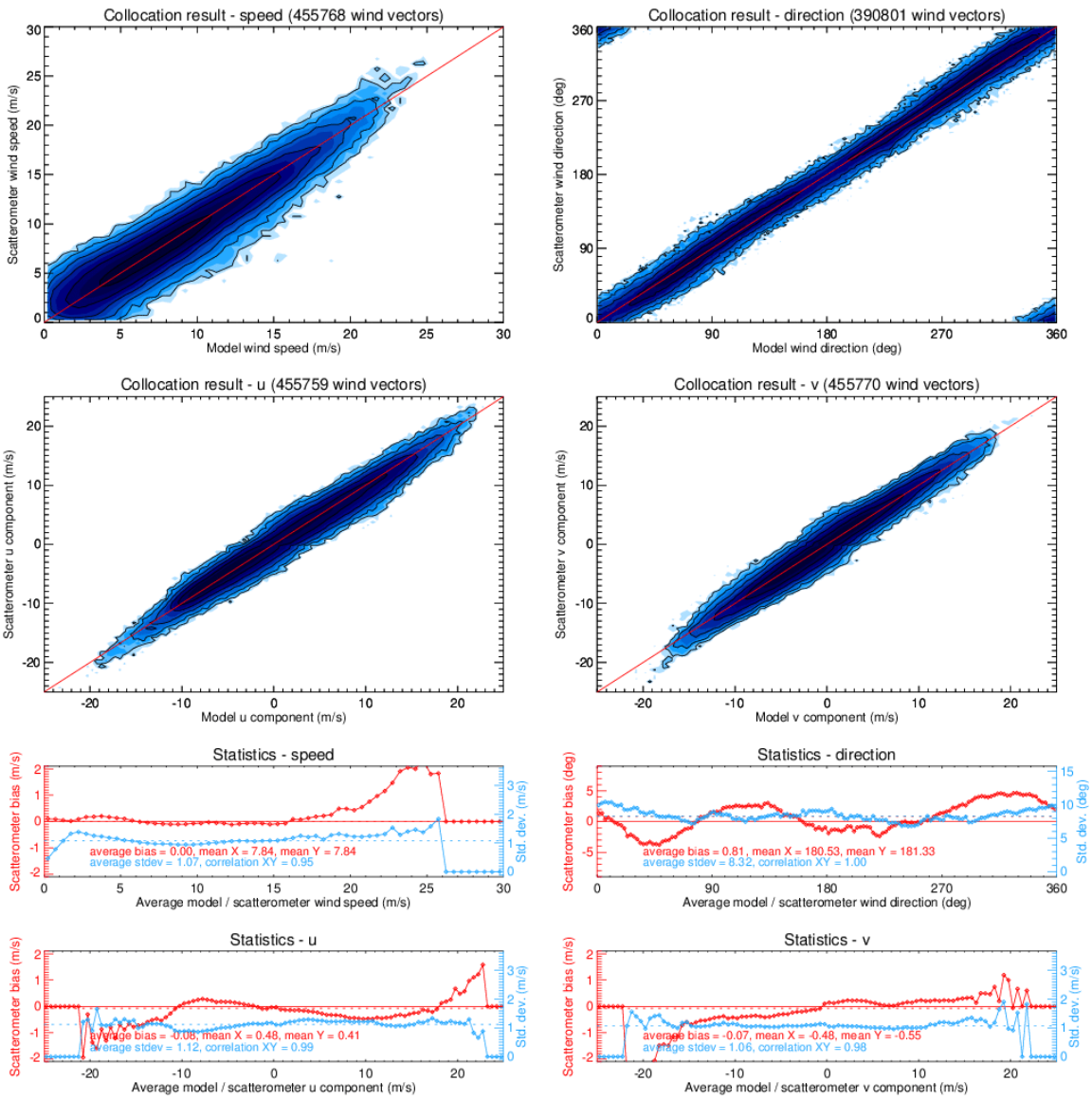

Figure 4.1 Collocation results of Oceansat-2 winds from PenWP and ECMWF ERA5 forecast winds. Biases and standard deviations in bottom plots are in m/s for wind speed and components, in degrees for wind direction.

4.2 Ice screening test

Figure 4.2 shows the ice maps for North and South poles after processing two days of data. The test data are the same as in the previous section, i.e., 9 and 10 February 2012. Ice maps of the North Pole and South Pole are provided. The blue parts in the maps indicate open water; the black parts correspond to land areas or areas not visited within these two days. The gray scale is a measure of the ice *A*-parameter (albedo). Multi year ice has in general a higher albedo than first year ice, so lighter areas correspond to older ice. In the scope of this report we did not verify the ice extent in detail with other measurements. More information about the ice screening algorithm can be found in [10].

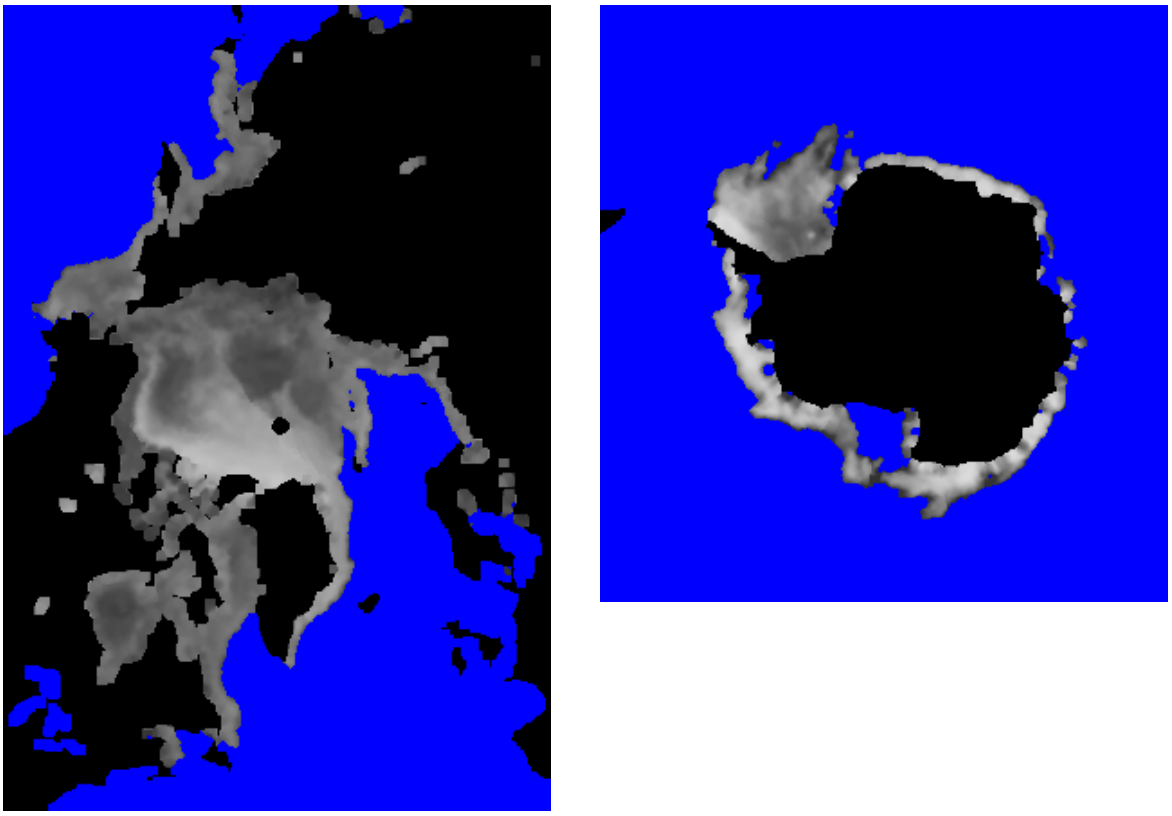

Figure 4.2 PenWP ice maps for North Pole (left) and South Pole (right).

5 Portability tests

The PenWP software package inherits its portability by using strict Fortran 90 code (with a few low level routines for reading and writing binary in C). PenWP is delivered with a complete make system. The Makeoptions include file of genscat takes care of the different settings needed under various platforms. This Makeoptions file is also used for the ASCAT wind processor AWDP and the CFOSAT scatterometer wind processor CWDP.

The default platform for development is a Linux work station. Different Fortran 90 compilers were used to compile both genscat and PenWP. Table 5.1 provides an overview of the platforms and compilers on which PenWP was tested successfully.

Platform	Operating system	Fortran compiler
Intel-based workstation	Fedora Linux v34	Portland f90 v11.10-0 Gfortran v11.2.1-1
Linux cluster	Redhat Linux v7.9	Gfortan v4.8.5-44
Apple MacBook	MacOS X Darwin	GNU Gfortran

Table 5.1 Supported platforms and compilers for PenWP.

6 User documentation tests

The user documentation (readme files within the software package and the PenWP user documents, [1], [2], [3]) have been provided to beta testers for review. The beta tester's comments have been implemented in the user documentation. User feedback on the documentation will also be implemented in future versions of the documentation.

References

[1] Verhoef, A., Vogelzang, J., Verspeek, J. and Stoffelen, A., 2022, *PenWP User Manual and Reference Guide,* Report NWPSAF-KN-UD-009, EUMETSAT.

[2] Verhoef, A., Vogelzang, J., Verspeek, J. and Stoffelen, A., 2022, *PenWP Product Specification,* Report NWPSAF-KN-DS-002, EUMETSAT.

[3] Verhoef, A., Vogelzang, J., Verspeek, J. and Stoffelen, A., 2022, *PenWP Top Level Design,* Report NWPSAF-KN-DS-001, EUMETSAT.

[4] NWP SAF, 2016, *Development Procedures for Software Deliverables,* NWPSAF-MO-SW-002, EUMETSAT.

[5] Freilich, M.H., 1999, *The SeaWinds Algorithm Theoretical Basis Document*, NASA, (available on http://eospso.gsfc.nasa.gov/sites/default/files/atbd/atbd-sws-01.pdf).

[6] Padia, K, 2010, *Oceansat-2 Scatterometer algorithms for sigma-0, processing and products format*, Version 1.1, April 2010, ISRO.

[7] Liu, D.C., and Nocedal, J., 1989 *On the limited memory BFGS method for large scale optimization methods*, Mathematical Programming, 45, pp. 503-528.

[8] Vogelzang, J., 2007, *Two dimensional variational ambiguity removal (2DVAR).* Report NWPSAF-KN-TR-004, EUMETSAT. (Available on http://www.knmi.nl/scatterometer/publications/).

[9] Verhoef, A., Vogelzang, J. and Stoffelen, A., 2021 *Scientific Validation Report (SVR) for the HY-2 winds,* report SAF/OSI/CDOP3/KNMI/TEC/RP/393, EUMETSAT.

[10] Belmonte Rivas, M. and Stoffelen, A, 2011, *New Bayesian algorithm for sea ice detection with QuikSCAT,* IEEE Transactions on Geoscience and Remote Sensing, 49, 6, 1894-1901, doi:10.1109/TGRS.2010.2101608.

[11] OSI SAF, 2021, *CDOP 3 Product Requirement Document*, SAF/OSI/CDOP3/MF/MGT/PL/2-001, EUMETSAT

[12] Wang, Z., Stoffelen, A., Zhang, B., He, Y., Lin, W., Li, X., 2019, *Inconsistencies in scatterometer wind products based on ASCAT and OSCAT-2 collocations,* Remote Sensing of Environment, 225, 207-216,

doi:10.1016/j.rse.2019.03.005.

[13] Wang, Z. et al., 2017,
SST Dependence of Ku- and C-Band Backscatter Measurements, IEEE Journal of Selected Topics in Applied Earth O, 10, 5, 2135-2146, doi:10.1109/JSTARS.2016.2600749

[14] Verhoef, A. and A. Stoffelen, 2021,
Advances in Ku-band scatterometer Quality Control, version 1.1, report SAF/OSI/CDOP3/KNMI/SCI/TN/404

[15] Verhoef, A., Vogelzang, J. and Stoffelen, A., 2022
Scientific Validation Report (SVR) for the Ku-band wind data records, report SAF/OSI/CDOP3/KNMI/TEC/RP/415

Appendix A: Acronyms

Name	Description
ASCAT	Advanced SCATterometer on Metop
AWDP	ASCAT Wind Data Processor
BUFR	Binary Universal Form for the Representation of data
C-band	Radar wavelength at about 5 cm
ECMWF	European Centre for Medium-range Weather Forecasts
EUMETSAT	European Organization for the Exploitation of Meteorological Satellites
genscat	generic scatterometer software routines
HSCAT	Scatterometer onboard of the Chinese Haiyang-2 series satellites
ISRO	Indian Space Research Organisation
KNMI	Koninklijk Nederlands Meteorologisch Instituut (Royal Netherlands Meteorological Institute)
Ku-band	Radar wavelength at about 2 cm
L1b	Level 1b product
LSM	Land Sea Mask
Metop	Meteorological Operational Satellite
MLE	Maximum Likelihood Estimator
MSS	Multiple Solution Scheme
NRCS	Normalized Radar Cross-Section (σ^0)
NWP	Numerical Weather Prediction
OSI	Ocean and Sea Ice
OSCAT	Scatterometer onboard of the Indian Oceansat and ScatSat satellites
OWDP	OSCAT Wind Data Processor
PenWP	Pencil beam Wind Processor
QC	Quality Control
SAF	Satellite Application Facility
SDP	SeaWinds Data Processor
SST	Sea Surface Temperature
WVC	Wind Vector Cell, also called node or cell

Table A.1 List of acronyms.