

# Optimisation of Rotating, Range-Gated Fanbeam Scatterometer for Wind Retrieval

ESA/ESTEC Contract No. 14383/00/NL/DC

task 2a report, GMF and wind field definition, and wind retrieval  
by: dr. J. de Kloe, KNMI,  
version 27-3-2003.

# Contents

<b>1</b>	<b>Reference Documents</b>	<b>3</b>
<b>2</b>	<b>Introduction</b>	<b>4</b>
<b>3</b>	<b>Proposed wind field scenarios</b>	<b>5</b>
3.1	Chooosen windfields . . . . .	5
3.2	Verification approach . . . . .	6
3.3	Software design . . . . .	7
3.4	Parameters . . . . .	7
<b>4</b>	<b>The Geophysical Model Function</b>	<b>9</b>
4.1	Construction of the functions . . . . .	9
4.2	Software design . . . . .	9
4.3	Verification approach . . . . .	10
4.4	Parameters . . . . .	10
<b>5</b>	<b>The geophysical noise contributions</b>	<b>12</b>
5.1	Algoritm . . . . .	12
5.2	Software design . . . . .	12
<b>6</b>	<b>The Level 1b generator</b>	<b>13</b>
6.1	Software design . . . . .	13
6.2	Verification approach . . . . .	13
6.3	File formats . . . . .	14
<b>7</b>	<b>The Wind Retrieval</b>	<b>15</b>
7.1	Algoritm . . . . .	15
7.2	Software design . . . . .	18
7.3	Verification approach . . . . .	18
7.4	Parameters . . . . .	18
<b>8</b>	<b>The Wind Direction Skill</b>	<b>20</b>
8.1	The problem of ambiguous solutions . . . . .	20
8.2	The normalised RMS . . . . .	20
8.3	Application of NRMS . . . . .	22
<b>9</b>	<b>Acronyms</b>	<b>23</b>

# 1 Reference Documents

- [RD1] “Optimisation of Rotating, Range-Gated Fanbeam Scatterometer for Wind Retrieval”  
Task 2b Report: Definition of Scatterometer Simulator, Astrium Dec. 2001.
- [RD2] “Optimisation of Rotating, Range-gated Fanbeam Scatterometer for Wind Retrieval”,  
Task 1 technical report, Review of the requirements and Scatterometer Concept,  
MBPT, oct. 17, 2000.
- [RD3] “Scatterometer Data Interpretation: Transfer Function Estimation and Validation”,  
Ad Stoffelen and David Andersen, Journal of Geophysical Research, 102(C3), 5767-  
5780, 1997b.
- [RD4] Thesis A.C.M. Stoffelen, University of Utrecht, 1998, ISBN 90-393-1708-9
- [RD5] “A model function for the ocean normalized cross section at 14 GHz derived from  
NSCAT observations”, F.J. Wentz and D.K. Smith, Journal of Geophysical Research,  
Vol. 104, No. C5, 11499-11507, 1999.
- [RD6] “Polarimetric Scatterometry: A promising technique for improving ocean surface wind  
measurements from space”, Tsai et al., IEEE Transactions on Geoscience and Remote  
Sensing, Vol. 38, No. 4, 1903-1921, 2000.
- [RD7] “Simulation of the Wind Performance of ASCAT”, OSI SAF technical note, by Ad  
Stoffelen, KNMI 1998.
- [RD8] “Towards the real-time use of Quikscat winds”, A. Stoffelen, J. de Vries, A. Voorrips,  
beleidscommissie remote sensing, report nr. USP-2/00-26.
- [RD9] “Optimisation of Rotating, Range-Gated Fanbeam Scatterometer for Wind Retrieval”  
Task 3a Report: FoM definition and validation, KNMI, 2002.

## 2 Introduction

This document defines a number of parts to be used by the RFSCAT simulator. Together with the parts defined in [RD1] they will form the complete simulator.

A review of the concept of RFSCAT is given in [RD2] and will not be repeated here. Some additional choices on the concept are given in section 2.3 of [RD1].

The overall goal of the RFSCAT study is to simulate the performance of the proposed system by performing an end-to-end simulation of windfield retrieval with the RFSCAT system. This will enable the system to be optimised with respect to the many design choices that can be made. The simulation will start with an input windfield and a set of parameters defining the antenna system and the applied processing. The output will be a set of possible wind solutions for each gridpoint (WVC). From the difference between input and output windfield a Figure-of-Merit (FoM) will be calculated which will be used as a metric for optimising the RFSCAT system.

The setup of the simulation is given in section 3 of [RD1]. The part named “Geophysical Module” in Fig. 3-1 of [RD1] will be delivered by KNMI and is further described below.

All parts of the Geophysical module are written in the Fortran90 programming language. They have been tested on a SGI system, using UNIX (IRIX release 6.5 IP27) and the MIPS-pro compiler (version 7.2.1). They have also been tested without problems on a SUN system using UNIX (SunOS 5.8 Generic 108528-11 September 2001), using the Sun WorkShop 6.2 Fortran 95 compiler. As no operating system specific language extensions are used, the software is expected to be portable to all platforms for which a Fortran90 compiler is available.

The subjects covered in this report are:

- proposed wind field scenarios (section 3)
- GMF model (section 4)
- expected geophysical noise contributions (section 5)
- the Level 1b Generator (section 6)
- wind retrieval (section 7)
- winddirection skill 8
- acronyms (section 9)

### 3 Proposed wind field scenarios

#### 3.1 Chosen windfields

To test the RFSCAT concept a set of input windvectors is necessary. Several choices are possible here. Windfields can be artificially created, taken from a numerical weather prediction (NWP) model, or adapted from actual measurements by another scatterometer (for example Seawinds on Quikscat). Alternatively the whole idea of a spatially coherent windfield can be discarded and replaced by a systematic scan of the wind domain using discrete steps in the windcomponents (this resembles a uniform distribution over all windvectors upto a certain maximum windspeed) or by a set of random windvectors, representing a realistic global wind vector distribution.

For testing of the RFSCAT concept we will follow the last approach and create an artificial set of random windvectors. Each windcomponent is chosen from a Gaussian distribution centered at 0 m/s and with a halfwidth of 5.5 m/s. Since the windvectors are chosen at random no spatial relation exists between them. In this way it is prevented that for example seasonal or local preferences in winddirection will cause a non-uniform sampling of the wind domain. An example of such a set of random windvectors is given in Fig. 1.

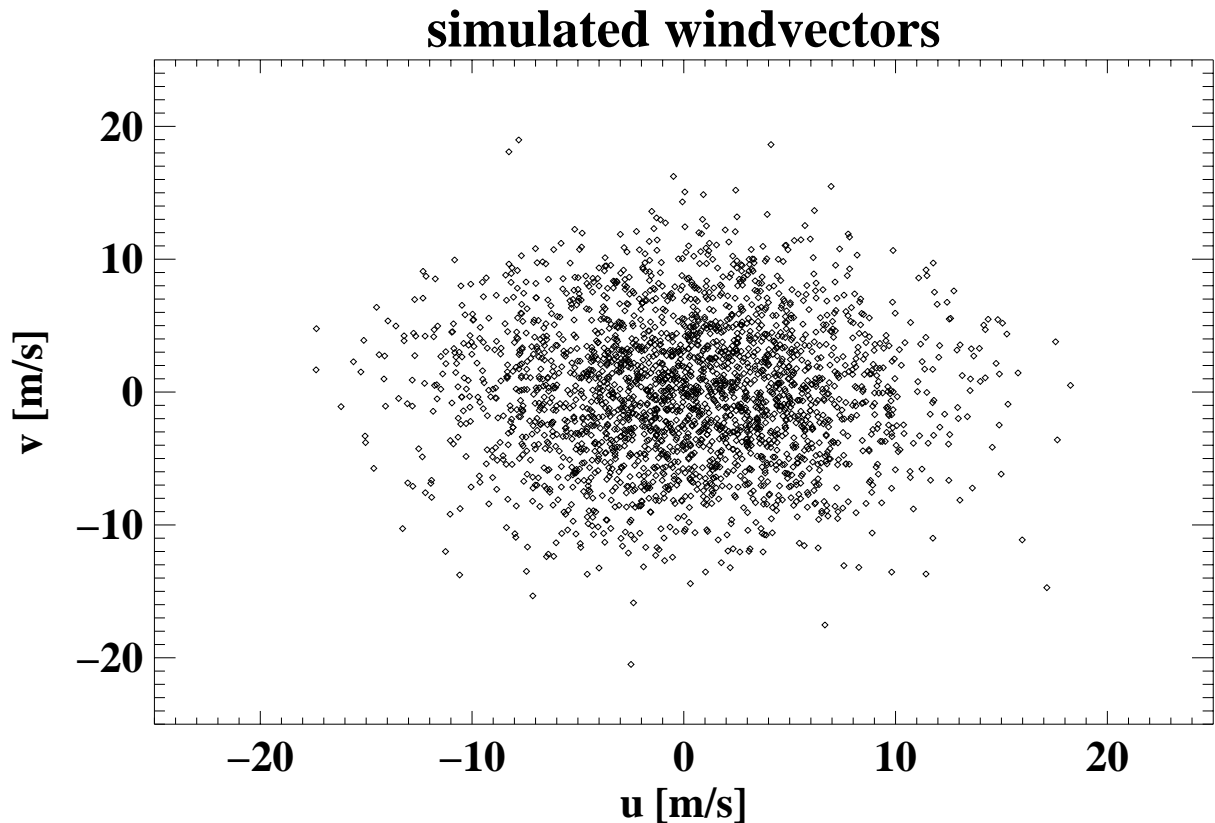


Figure 1: A simulated set of 2500 windvectors within the range of 0 - 25 m/s.

To justify this approach, in Fig. 2 a set of windvectors has been extracted from an ECMWF NWP model, on locations defined by one simulated RFSCAT orbit. It can clearly be seen that the sampling is far from regular, even though the number of windvectors is much

larger than shown in Fig. 1. To obtain a large enough set of windvectors clearly a much larger dataset is needed, which would need an unnecessary large amount of simulations.

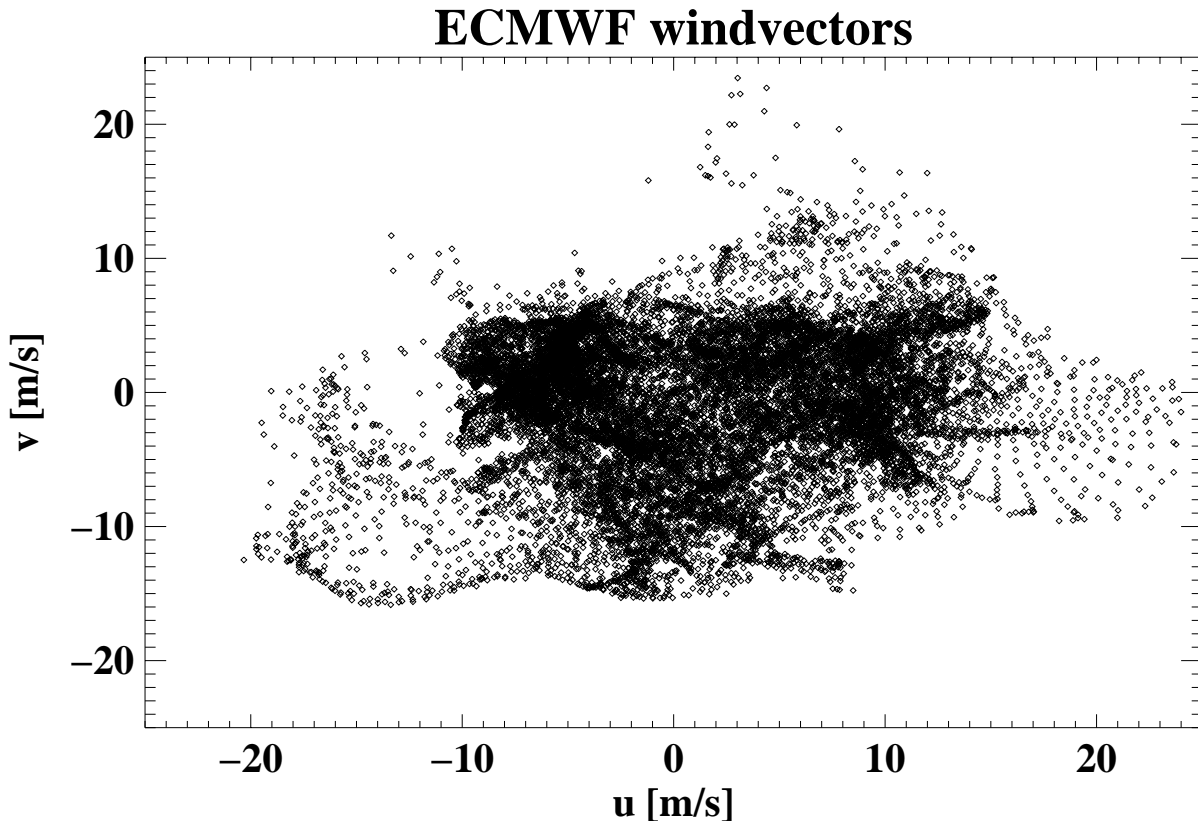


Figure 2: A set of 16414 windvectors extracted from an ECMWF NWP model (date 20020226, analysis of 12 h), on locations defined by one simulated RFSCAT orbit. 8634 vectors have been discarded because they were located above land or ice.

However, even a very large set of windvectors will not sample the measurement space regularly, as is shown in the histograms of Fig. 3. This is an example for the ECMWF forecast windvectors on the position of the real Quikscat on Seawinds instrument. It is clear from this figure that especially the  $v$  component (east-west) has a large deviation from the Gaussian shape.

### 3.2 Verification approach

To verify this approach a test can be done using a windfield produced by the ECMWF forecast model as simulation input. Finally it may be interesting to test the effect of the scatterometer measurement on coherent structures (rotational, laminar, etc.). To enable these tests the simulation program will accept a file describing a windfield, as alternative to the random calculation of windvectors. This file will be produced by a separate program which is of no importance for the simulation itself, so it is not described in detail here. The fileformat to be used for this input windfield is given in section 7.5 of [RD1] and is not repeated here.

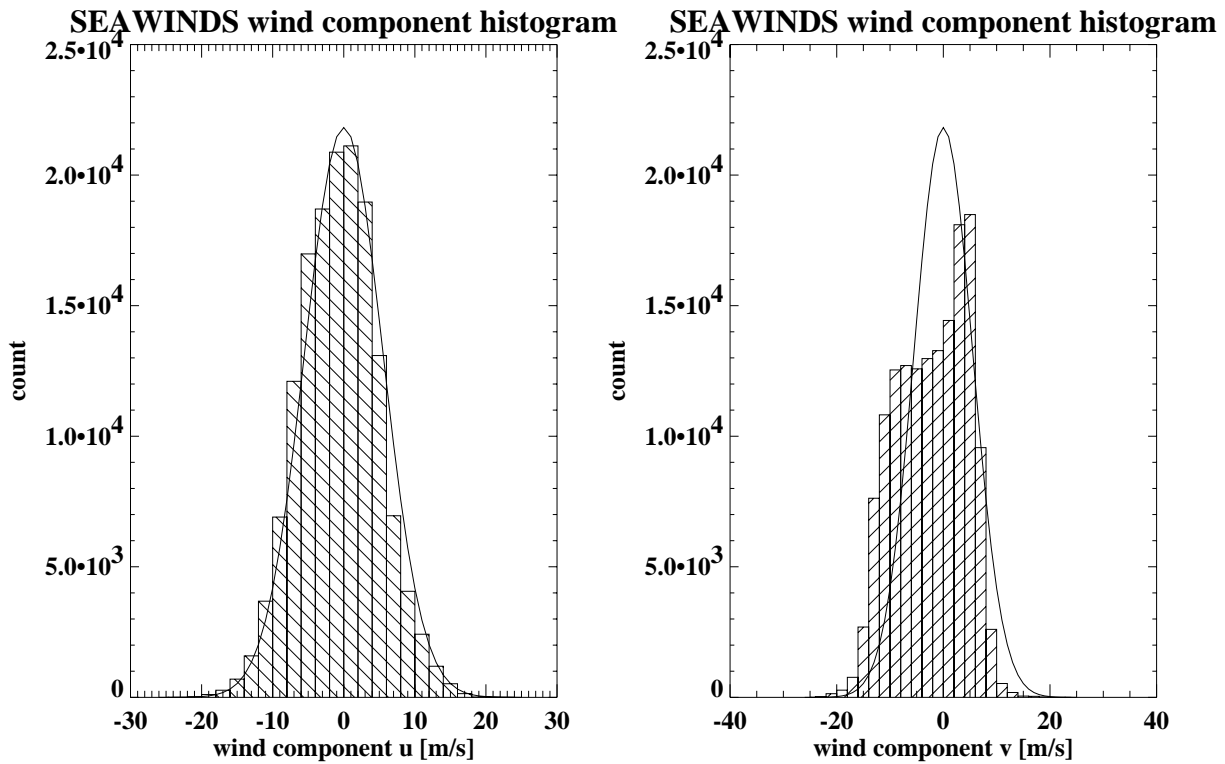


Figure 3: A set of 150413 windvectors extracted from an ECMWF NWP model (2 to 4 feb. 2002, forecast), on locations defined by the Seawinds satellite orbit.

### 3.3 Software design

This part of the simulation software is called the “windfield generator” and is implemented in the form of a Fortran90 module. It contains a function to retrieve one windvector at a time, which is either calculated from the described distribution, or taken from the input file. A second function returns the number of available windvectors, which is given in the inputfile, or specified as a constant for the random determined windvector case. Furthermore a helpfunction is provided that closes the inputfile.

Two functions that handle noise are also provided. First a function is provided to calculate the noise due to the variability of the windfield inside the defined WVC. This function takes the choosen measurement resolution into account (for more details see sec. 5). Second a function is provided which adds noise to the background windfield that is used by the simulation. This is especially usefull when simulation results are to be compared to real measured scatterometer data. During the optimisation of the rfscat concept this noise should be set to zero.

### 3.4 Parameters

Important parameters in the windfield generator are:

- the choice between a windfield from file, or a generated set of random windvectors
- the possibility to select a range of windspeeds

The importance of different windscenarios has been discussed above. The possibility to select a range of windfields is especially useful for windfields taken at random from a given distribution. From first tests of the program we know that inversion for very low windspeeds is very difficult, if not impossible in the presence of the usual noise levels in the system. So for windspeeds below 0.8 m/s no useful winddirection will be found. When examining the program results these low windspeed cases may be dominant and may obscure problems at higher windspeeds. Therefore it should be possible to exclude them from the statistics. This is preferably done by the windfield comparison module, but can for testing purposes also be done using this range setting.

A maximum windspeed of 25 m/s has been chosen, since for the applied distribution almost no vectors above that value are generated. It is also possible to select for example a smaller range between for example 14 and 15 m/s, in order to test the simulator behaviour as a function of windspeed. Finally a restriction is useful since the look up table ranges from 0.2 to 50 m/s. In combination with a windfield from an inputfile, the restriction in windspeed range is not used.

The parameters for this module are given in table 1. The choice between using an input file or generating random or gridded windvectors is made using CPP switches. The windfilename can be set using a CPP definition. The number of windvectors to use in a simulation can be explicitly given in the headerfile, or can be derived from the pseudoL1b file.

Table 1: Windfield generator parameters

ID	Name	Unit	type	description
WG-10	windfilename		string	path and filename of the input windfield file
WG-20	dist_halfwidth	m/s	real	halfwidth of the windspeed distribution
WG-30	min_wind	m/s	real	minimum value of the range of windspeeds
WG-40	max_wind	m/s	real	maximum value of the range of windspeeds
the following parameters control the systematic scan of the winddomain				
WG-50	u_startval	m/s	real	start for scanning u comp.
WG-60	v_startval	m/s	real	start for scanning v comp.
WG-70	u_step	m/s	real	step for scanning u comp.
WG-80	v_step	m/s	real	step for scanning v comp.
WG-90	nr_of_u_steps		integer	nr of u steps for scan
WG-100	nr_of_v_steps		integer	nr of v steps for scan

For additional flexibility, the possibility was added to define the filename for the input windvectors as a commandline option. This is done by giving the option: “-w windfile.dat”. This will override the path and filename defined in the simulate.h headerfile.



## 4 The Geophysical Model Function

### 4.1 Construction of the functions

The Geophysical Model Function (GMF) is the function providing the normalised radar backscatter coefficient, or normalised radar cross-section (nracs). It depends on the measurement geometry (relative azimuth and incidence angle), beam parameters (frequency, polarisation) and the windspeed and direction. The GMF is a function devised to fit the available experimental data as good as possible, but since experimental knowledge is growing year by year refinements are now and then considered.

For the RFSCAT project the CMOD4 function for C-band VV-polarisation (so transmitting and receiving in vertical polarisation) is used [RD3, RD4]. This function has been used for operational windretrieval for the ERS-2 Scatterometer data (see section 4.2.1 of [RD2]). For Ku-band the NSCAT2 functions are used for VV and HH polarisation [RD5]. These functions have been used for the final reprocessing of all NSCAT data (see section 4.2.1 of [RD2]).

A GMF function for C-band HH-polarisation is not readily available. Therefore it was estimated by scaling the CMOD4 function with the HH/VV ratio as defined by the NSCAT2 HH and VV functions (see section 4.2.3 of [RD2]). The estimated function is called CMOD4H.

A GMF function for polarimetric measurements does not yet exist for both C-band and Ku-band. Therefore a similar scaling procedure was necessary. The equations necessary to do this are given in [RD6]. Furthermore a depolarisation ratio of -15dB, a symmetry factor of 0.5 and a signal attenuation of 1.0 were chosen. Using these formulas and assumptions a new function called NSCAT2P was derived from the NSCAT2 functions, and a new function called CMOD4HP was derived from the CMOD4 and the CMOD4H functions.

The derivation of the CMOD4H, CMOD4HP and NSCAT2P functions has been performed by V. Wismann of Ifars and is available for the project at the moment (see section 4.2.4 of [RD2]).

### 4.2 Software design

The GMF functions will be used in two parts of the program. First they are used to calculate a pseudo measurement from an input windvector. Then they are used many times inside the wind retrieval module while doing the inversion. Therefore the code has been inserted in the wind retrieval module. The GMF values can be requested by using a function call named "calc\_sigma0", with some parameters describing the relative azimuth, the incidence angle and the polarisation. In order to be able to use different frequency bands, the parameter named "polarisation" is in fact also used to set the right frequency, so to use the right GMF function. This parameter is read from the pseudoL1b file, which gives a two character code for each measurement. The first character is "C" or "K", to choose between C-band and Ku-band. The second character is "V", "H", or "P", to choose between vertical and horizontal polarisation, and a polarimetric measurement scheme.

For reasons of calculating efficiency a look up table (LUT) is used to store the function values. The necessary data is loaded automatically in the LUT at the first function call of "calc\_sigma0". Everytime a value is requested from the function the proper table elements

are taken and if necessary interpolation between them is done. The fileformat to be used for the LUT is described in section 7.4 of [RD1] and will not be repeated here.

Interpolation for incidence angle is always done. Interpolation for windspeed and/or azimuth can be switched on and off, depending on the desired accuracy and available time to do the calculations.

### 4.3 Verification approach

Verification of the constructed NRCS functions has been done in [RD2] by comparing the resulting values with the available data in literature (see sections 4.2.1 upto 4.2.4 of [RD2]). Verification of the usage of the LUT and interpolation between the table elements has been done by directly implementing the actual CMOD4 function as specified in [RD4] in the Fortran90 code, and comparing with its values.

### 4.4 Parameters

The parameters for this module are given in table 2. GM-10 to GM-90 specify the LUT and are defined in the inversion.f90 code. GM-100 to GM-160 specify the path and filenames for the files containing the LUT's.

Table 2: GMF parameters

ID	Name	Unit	type	description
GM-10	LUT_min_windspeed	m/s	real	start value in windspeed
GM-20	LUT_windspeed_step	m/s	real	step value in windspeed
GM-30	LUT_nr_windspeed_steps	m/s	integer	nr of steps in windspeed
GM-40	LUT_min_winddir	m/s	real	start value in azimuth
GM-50	LUT_winddir_step	m/s	real	step value in azimuth
GM-60	LUT_nr_winddir_steps	m/s	integer	nr of steps in azimuth
GM-70	LUT_min_theta	m/s	real	start value in incidence angle
GM-80	LUT_theta_step	m/s	real	step value in incidence angle
GM-90	LUT_nr_theta_steps	m/s	integer	nr of steps in incidence angle
GM-100	LUT_path		string	path for the LUT's
GM-110	filename_c_vv		string	filename for the CMOD4 LUT
GM-120	filename_c_hh		string	filename for the CMOD4H LUT
GM-130	filename_c_pol		string	filename for the CMOD4HP LUT
GM-140	filename_ku_vv		string	filename for the NSCAT2-VV LUT
GM-150	filename_ku_hh		string	filename for the NSCAT2-HH LUT
GM-160	filename_ku_pol		string	filename for the NSCAT2-P LUT

In addition to this the interpolation in between the LUT elements may be set. This is done by CPP commands defined inside the file inversion.h. Possible choices are 1D (only incidence angle), 2D (incidence angle and azimuth), 2DV (incidence angle and windspeed)

and 3D (incidence angle, azimuth and windspeed). For the best results the use of 3D interpolation is recommended. 1D interpolation is not a good idea, and was mainly implemented to maintain compatibility with previous versions of the inversion software.

For additional flexibility, the possibility was added to define the pathname for the LUT as a commandline option. This is done by giving the option: “-lp LUT-path”. This will override the path defined in the inversion.h headerfile.

## 5 The geophysical noise contributions

Geophysical noise is the effect of local variations of the wind inside the region of the WVC. It has been experimentally found that this can be treated as an extra source of noise, on top of the noise produced by the instrument. The squares of the two types of noise should then be added to obtain the square of the total noise in the system.

### 5.1 Algorithm

It was found that the geophysical noise is a quadratic function of windspeed, increasing dramatically for low windspeeds, and going to practically zero (i.e. negligible compared to instrument noise) above 16 m/s [RD7]. The following function is used to describe this for windspeeds below 16 m/s (but only for a wvc resolution of 50 km):

$$\text{geo\_noise} = 0.644 \times 10^{-3} \times (\text{windspeed} - 16.0)^2 \quad (1)$$

Thus for 16 m/s the `geo_noise` is 0, for 9 m/s the `geo_noise` is 0.032 (3.2 %) and for 3 m/s the `geo_noise` is 0.109 (10.9 %). For windspeeds above 16 m/s `geo_noise` is set to zero.

This relation can be adjusted for use with a different resolution  $r$  (in km) in the following way (for details, see section 3.2 of [RD9]):

$$\text{geo\_noise} = 0.644 \times 10^{-3} \times \left(\frac{r}{50}\right)^{\frac{1}{3}} \times (\text{windspeed} - 16.0)^2 \quad (2)$$

### 5.2 Software design

The geophysical noise is calculated by calling the function “`calc_noise_due_to_windvar`”. This function is part of the windfield generator module which is described in section 3.3. It is called separately for each windvector.

There are no free parameters in this part of the software. A description of the verification of this concept can be found in [RD7].

## 6 The Level 1b generator

The Level 1b (L1b) Generator is the part of the Geophysical module that calculates the simulated measurements and adds the noise to the measurements. In order to do this it needs the wind vector provided by the Windfield generator and the geometry calculated by the Pseudo L1b generator. The resulting simulated measurements are passed on to the windfield retrieval module.

### 6.1 Software design

The L1b generator actually consists of the main loop of the Geophysical Module. It loops over the windvector cells provided by the Pseudo L1b generator, and does a number of subroutine calls for each WVC. First a call is made to the subroutine “read\_meas\_geometry” which reads the next entry from the Pseudo L1b file. Then the subroutine “define\_windfield” is called to get a windvector. Then the NRCS values for all views are calculated by calling the function “calc\_nrcs\_values”. Following this the Kp values are retrieved from the Pseudo L1b file by calling the subroutine “set\_noise\_variance\_kp”, and the geophysical noise is calculated by calling the function “calc\_noise\_due\_to\_windvar”. Then gaussian distributed noise is added to the simulated measurement (using the combined geophysical and instrumental noise) by calling the subroutine “add\_instr\_and\_geo\_noise\_to\_nrcs”. Finally the wind retrieval is done by calling the subroutine “invert\_nrcs\_to\_winds” (see section 7), and the resulting windvectors are stored in arrays to be able to write them to file after the loop is done.

Some additional code was added to the main loop to perform the calculations for several wind vector cells in parallel. For this the MPI library is used. Since this is an ambarassingly parallel problem, the tasks are totally independant and only a minimum of communication is necessary, a speedup of close to 100% is expected. The work is devided over the processors using the node-numbers, and each processor generates its own outputfile. A small help program is provided that allows combining all these partial files into one resultfile.

No parameters are used in this part of the code. Most information is delivered by the called subroutines. Only some switches can be set in the simulate.h header file, in order to choose between ascii/binary output, to switch binning on/off, and to enable or disable the MPI code.

### 6.2 Verification approach

Since in this part of the software a number of modules are combined a separate verification is not possible. Verification of the complete Geophysical module has been done by running the simulation without noise. In that case the produced windvectors can be easily compared to the input windvectors and verified. This gave perfect results for the retrieved closest windvector, without any bias, and only minimal rms (caused by the finite steps in the LUT of the GMF).

### 6.3 File formats

In addition to the ascii file formats as described in sections 7.2 and 7.3 of [RD1] also a binary fileformat will be used as possible output. This is necessary to improve performance in the IDL code of the Windfield Comparison Module. In order to use this a maximum number solutions of 4 is taken. The contents of this file will be the input windvector, and the 4 possible retrieved windvectors (sorted by rank) for each processed WVC. If less then 4 solutions are found the elements will be set to zero. Also 2 vectors defining the the satellite heading are included.

The fileformat will consist of 4byte integers and 4byte reals, with the following contents:

- one integer containing the number of processed WVC's.

Then each WVC is described using 3 integers and 22 reals. These are stored in the file as arrays with `nr_of_wvcs` elements. So the file will contain:

- an array containing the WVC number
- an array containing the number of wind solutions (beware: cases with zero or only one solution are also valid possibilities)
- an array containing a wind quality code, specifying whether the wind retrieval was succesfull or not, and if not, what was the problem.
- 6 arrays containing the 2 vectors defining the satellite heading.
- 2 arrays containing the 2 components of the input windvector
- 2 arrays containing the 2 components of the first rank solution
- 2 arrays containing the 2 components of the second rank solution
- 2 arrays containing the 2 components of the third rank solution
- 2 arrays containing the 2 components of the fourth rank solution
- an array containing the MLE of the first rank solution
- an array containing the MLE of the second rank solution
- an array containing the MLE of the third rank solution
- an array containing the MLE of the fourth rank solution
- 2 arrays containing the lat,lon position of the WVC

For efficiency reasons inside IDL it is preferred to write the complete arrays in this order to the file, in stead of writing the array elements containing the results for each WVC together. The wind directions will be defined using the east-west direction for the u-component (east is positive), and the counter clock wise direction perpendicular to this for the v-component (north is positive).

## 7 The Wind Retrieval

The wind retrieval or inversion of the scatterometry data will be done by the well known algorithm that is used for the current and past scatterometer instruments as well (for example for Seawinds on Quikscat).

### 7.1 Algorithm

A maximum likelihood estimator (MLE) is calculated for a given geometry and simulated measurement using:

$$\text{MLE} = \sum_{\text{views}} \frac{(\sigma_{\text{tried}}^0 - \sigma_{\text{measured}}^0)^2}{f^{\text{norm}}} \quad (3)$$

with  $\sigma_{\text{measured}}^0$  the measured backscatter coefficient  $\sigma^0$  of a view,  $\sigma_{\text{tried}}^0$  the LUT value for the geometry of the current measurement, and for current wind speed and direction that is tried out,  $f^{\text{norm}}$  a weighting factor, for Seawinds it is taken proportional to the instrument noise in the system (i.e.  $f^{\text{norm}} = (k_p \sigma^0)^2$ ), in order to let the more accurate measurements have more impact than the very noise ones (however for ERS or ASCAT it is not proportional).

When different types of measurements are combined, the summation is a combination of all measurements, so adding all different types of views. The possibility exists to give a different weight to the polarimetric measurements, as compared to the horizontally and vertically polarised measurements. However, this tuning has not yet been used, since equal weights give already very good results.

The 2 dimensional winddomain (speed and direction) is then scanned to find the local minima that are present in this plane. These should give the set of possible windvectors that give  $\sigma^0$  values that best coincide with the measured values.

The scanning of the winddomain is split in two parts. First for each winddirection (scanned using fixed steps) the windspeed is scanned to find the windspeed having minimum MLE (see Fig. 4). It is known that in this direction only one minimum should occur, so the procedure is fairly simple. Starting from a first guess the slope downhill (in MLE) is followed with given steps until the minimum is found. Then, if it is chosen to do so, the steps are reduced in size, and the procedure is repeated using the found minimum as first guess. If the steps have reached a specified minimum size the search is halted, and the windspeed giving this minimum MLE is stored in an array, together with this minimum MLE value.

To improve accuracy without having to use excessively small steps it is possible to use a parabolic fit to the 3 points around the minimum in windspeed, to refine the minimum position. Since the  $\sigma^0$  function against windspeed is usually not strongly curved the main shape of the minimum will be caused by the squares in the MLE formula, and is expected to be very similar to a parabola. Therefore this will work rather well.

The array of found windspeeds giving minimum MLE (this is called the minimum valley in the wind domain) defines a curve, which is again used in a minimum search in the next part. Now the winddirection is scanned using the same fixed steps as mentioned above, in search of a local minimum in MLE (see Figs. 5 and 6). The number of minima in this direction is not known in advance. In special cases there may be just one, but usually 2 to 4 are found. However, also 5 to 10 minima have occurred, and in some cases with numerical

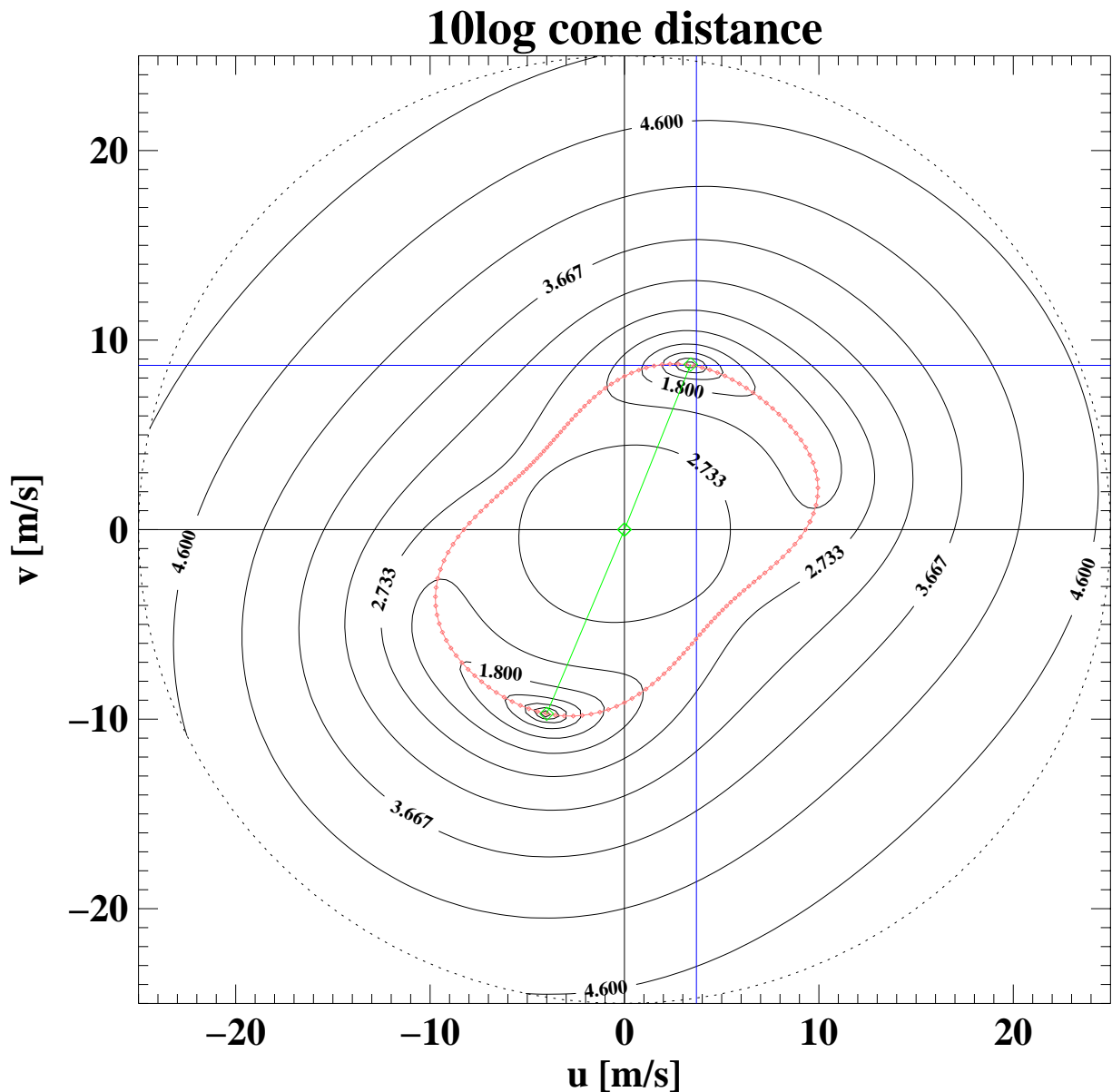


Figure 4: *The MLE function in the two dimensional wind domain (contours). The crossing blue lines indicate the input wind vector of the simulation. The red line indicates the minimum-valley (i.e. minimum MLE for each wind direction). The green vectors indicate the found wind solutions. The closest is not identical to the position indicated by the blue cross due to the simulated noise.*

difficulties (for example on the edge of the LUT, or for too large steps in windspeed or wind direction) many more local minima can occur. In rare cases it can even happen that no solutions are found.

Also for the minimum search along wind direction a refinement can be done by using a parabolic fit to the 3 points around the minimum. However the shape of this function does not have to be close to a parabola, so this may not always improve the result in the same way as it does for the windspeed. Especially when the polarimetric measurement scheme is



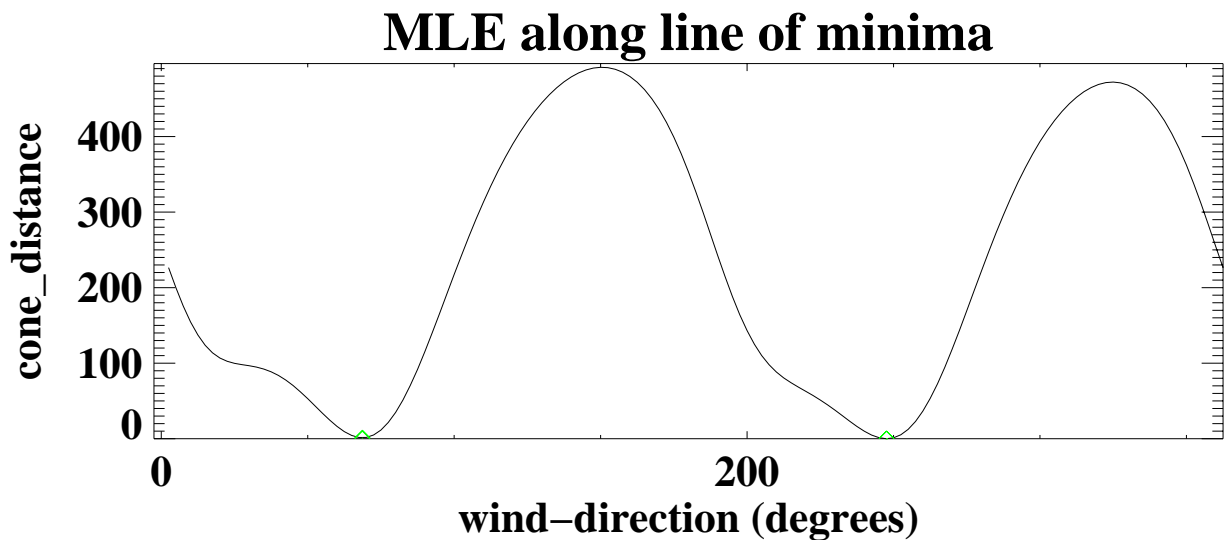


Figure 5: *The MLE function along the minimum valley, indicated by the red line in Fig. 4. The green diamonds indicate the found minima.*

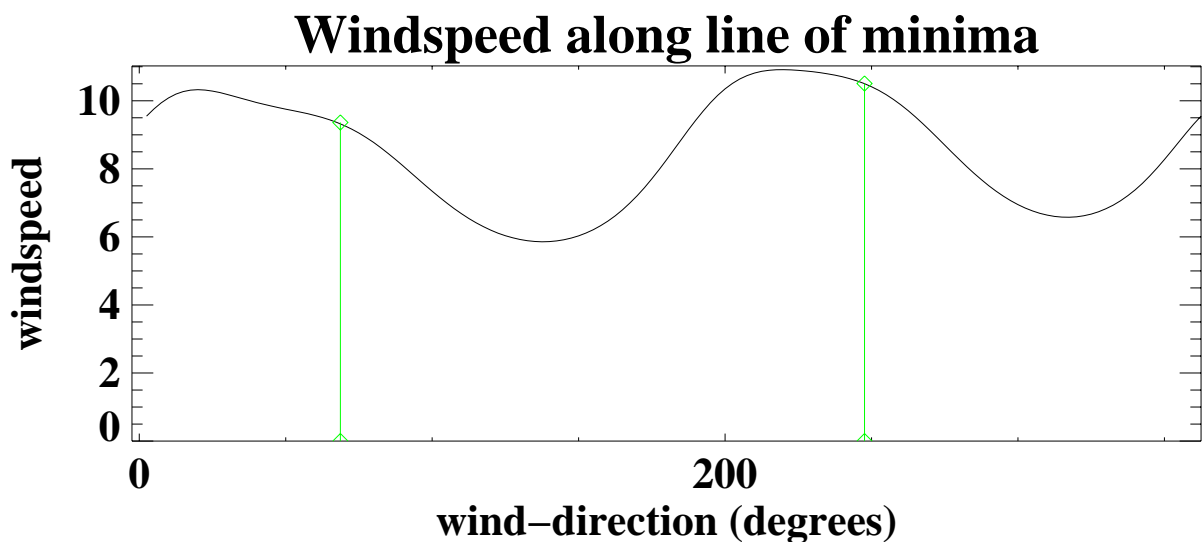


Figure 6: *The windspeed along the minimum valley. The green lines indicate the found solutions.*

used, it is advised not to use this option.

The solutions are then sorted by MLE value, since the lowest MLE values have the largest chance of being the right wind solution (which is how the MLE = Maximum Likelihood Estimator should behave).

Sometimes a smoothing is used on the MLE curve of the minimum valley, but this produces systematic errors in winddirection. The solutions will be biased towards certain directions (which in turn depend on the varying measurement geometry, so are difficult to spot). For compatibility reasons it is available in the inversion module, but should preferably not be used. A better way of avoiding numerical problems is taking smaller steps through

the GMF and avoiding the edge of the LUT.

## 7.2 Software design

The wind retrieval module is implemented as a Fortran90 module containing for the user two important subroutines: “invert\_one\_wvc” and “calc\_sigma0”. Furthermore a number of variables are defined to communicate with the invert routine. The input variables are arrays defining for each view the measurement geometry, the  $\sigma^0$  and the noise properties. The output variables are arrays defining for each found solution the MLE and the windvector, and 2 integers giving the found number of solutions and a wind quality code (defining whether the inversion was succesfull or not).

## 7.3 Verification approach

The windfield retrieval (inversion) routine has already been extensively tested using real data. This was done using measurements from seawinds on quikscat, and comparing the result by the wind field retrieval routine, to the result of an older routine that is in use now for the operational use of this data. From this it was found that the new routine performs slightly better if the proper choices are made for stepping through the wind domain. This is at the expense of more calculating time. When the settings are choosen to match the behaviour of the old retrieval routine, an exact agreement between the results of both routines was found.

## 7.4 Parameters

Due to the many choices that can be made inside the windfield retrieval module, a large number of parameters is available to tune it. They are defined as CPP commands, and processed at compile time in order not to negatively influence the program runtime. These settings are stored in the file inversion.h.

The most important switches are:

- PARABOLIC\_MIN\_SEARCH to define that a parabolic fit is done to find the minimum while scanning the windspeed.
- PARABOLIC\_WINDDIR\_SEARCH to define that a parabolic fit is done to find the minimum while scanning the winddirection
- NR\_OF\_WINDDIRSTEPS the number of steps to do while scanning along the wind-direction (if this is equal to the steps in the table, interpolation in this direction is useless).
- INITIAL\_WS\_STEP initial step to be used while searching for the minimum by scanning the windspeed
- MINIMUM\_WS\_STEP if the stepsize drops below this value no more refining of the stepsize is done while scanning the windspeed.
- REFINER\_WS\_STEP factor used to refine the windspeed step size (by multiplication).

- `WS_INITIAL_GUESS` first guess when doing the minimum search in windspeed for the first azimuth angle. For subsequent angles the minimum found for the previous angle is used as first guess.
- `MAX_SOL` defines the maximum number of solutions that the inversion module generates. If more minima are present only the `MAX_SOL` solutions having the lowest MLE value are given as result.
- `USE_SMOOTHING` specifies that the MLE curve along the minimum valley should be smoothed in order to eliminate unnecessary minima.
- `NO_NORMALISATION` do not use normalisation in eq. 3. This means putting  $f^{norm} = 1$ . This has been proven to be the optimal setting for the ERS scatterometer.
- `USE_MEAS_SIGMA0_FOR_NORMFACTOR` by default the normalisation factor  $f^{norm}$  is taken to be  $(k_p \sigma^0)^2$ , where for  $\sigma^0$  the value of  $\sigma_{tried}^0$  is taken. By setting this switch this is behaviour changed to using  $\sigma_{measured}^0$ . This has been proven to be a good setting for the Seawinds scatterometer on Quikscat (so will probably also be a good thing for RFSCAT).

If other normalisations should be possible they can be easily added to the software.

## 8 The Wind Direction Skill

As was discussed in the Simulator Design Review meeting held on december 18, 2001, at ESTEC, a method is needed to be able to compare the direction of the windvector resulting from the simulation, with the direction of the windvector that was used as input. It was proposed to use the so called “Normalised RMS” (NRMS) for this [RD8]. In this section a short description of this method with applicable equations is given. For verification of this idea see [RD8].

At a later stage in the project this NRMS was abandoned again, and replaced by a new “Figure of Merit” which is described in detail in [RD9].

### 8.1 The problem of ambiguous solutions

When comparing or validating an inverted wind direction against a reference wind direction, i.e. the input wind direction in our case, it is easiest to compute just the RMS difference. However, it is clear that if more ambiguous solutions are found the RMS number will become smaller, which would give a false sence of improvement. In the meantime the information content of the set of solutions decreases with an increasing number of solutions. Therefore an addition is proposed to the usual RMS calculation.

### 8.2 The normalised RMS

To estimate the quality of the closest solution in a fair way, a normalisation factor is defined which equals the expected value in case of no skill in the system:

$$\text{NRMS} = \text{RMS} \left( \frac{\phi^C - \phi^T}{\langle (\phi^C - \phi^T)^2 \rangle^{\frac{1}{2}}} \right) \quad (4)$$

in which  $\phi^T$  is the input (=true) winddirection,  $\phi^C$  is the inverted solution which is closest to the true solution, and  $\langle \rangle$  denotes the expected value in case there is no skill.

Now define  $\phi_1$  to be the closest solution  $\phi^C$ , define  $\phi_2$  as the counterclockwise and  $\phi_0$  as the clockwise neighbouring solution (see Fig. 7). Also assume the angles are sorted in counterclockwise direction, so first  $\phi_1$ , then  $\phi_2$  (if it exists), then  $\phi_3$  (if it exists), etc.

Then define the boundaries between the solutions as:

$$\phi_i^B = \frac{\phi_{i-1} + \phi_i}{2} \quad (5)$$

in which  $i$  can be the number of each solution and for convenience we define  $\phi_0 = \phi_N - 2\pi$ . (with N the number of solutions).

Now it is reasonable to assume that the true wind direction  $\phi^T$  is somewhere between  $\phi_1^B$  and  $\phi_2^B$ . Taking a uniform distribution between these two boundaries the probability  $p(\phi^T)$  becomes:

$$p(\phi^T) = \begin{cases} (\phi_2^B - \phi_1^B)^{-1} & \text{if } \phi_1^B \leq \phi^T \leq \phi_2^B \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Using this distribution we find:

$$\langle (\phi^C - \phi^T)^2 \rangle = \frac{[(\phi_2 - \phi_1)^3 + (\phi_1 - \phi_0)^3]}{12(\phi_2 - \phi_0)} \quad (7)$$

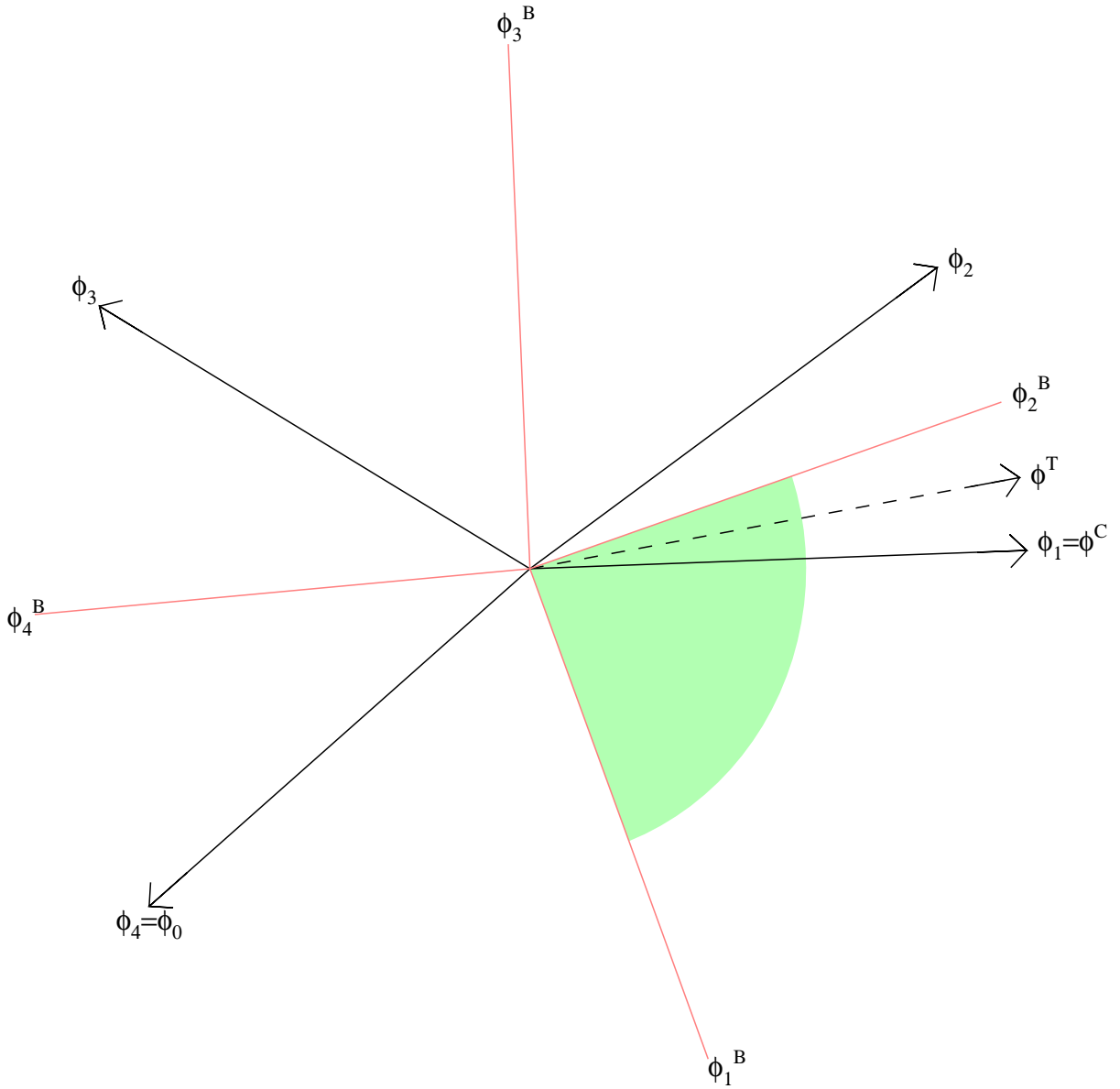


Figure 7: An example (with 4 solutions) of the windvector definitions and sector boundaries used for calculating the NRMS.  $\phi_1$  to  $\phi_4$  are the four solutions for this case.  $\phi_1^B$  to  $\phi_4^B$  are the boundaries of the sectors surrounding each solution. The sector indicated in green is the one used for normalisation by the NRMS.  $\phi^T$  is the true wind, which is only used to determine which of the four solutions is the closest ( $= \phi^C$ ).

For one solution equation 7 yields just a constant number:

$$\langle (\phi^C - \phi^T)^2 \rangle = \frac{\pi^2}{3} \quad (8)$$

For two solutions the angle between the two solutions becomes important, and equation 7 yields (after setting  $\phi_0 = \phi_2 - 2\pi$ ):

$$\langle (\phi^C - \phi^T)^2 \rangle = \frac{1}{4}(\phi_1 - \phi_2)^2 + \frac{1}{2}\pi(\phi_1 - \phi_2) + \frac{\pi^2}{3} \quad (9)$$

For three solutions equation 7 yields (after setting  $\phi_0 = \phi_3 - 2\pi$ ):

$$\langle (\phi^C - \phi^T)^2 \rangle = \frac{(\phi_1 - \phi_3 + 2\pi)^3 + (\phi_2 - \phi_1)^3}{12(\phi_2 - \phi_3 + 2\pi)} \quad (10)$$

Finally for four solutions equation 7 yields (after setting  $\phi_0 = \phi_4 - 2\pi$ ):

$$\langle (\phi^C - \phi^T)^2 \rangle = \frac{(\phi_1 - \phi_4 + 2\pi)^3 + (\phi_2 - \phi_1)^3}{12(\phi_2 - \phi_4 + 2\pi)} \quad (11)$$

### 8.3 Application of NRMS

In applying the NRMS for optimisation of scatterometer wind retrieval we found the clear benefit, as expected, with respect to the use of the common wind direction RMS difference.

However, a note of caution may be appropriate, since we found that the NRMS may improve in certain cases where ambiguous (anyway unlikely) solutions are removed. But, on the other hand, at the same time some noise is added to the closest solution. The latter obviously reduces the information content of the data, but the effect of reducing the number of ambiguities can dominate the NRMS, thus reducing its value, and thereby indicating improvement.

Therefore the use of the NRMS alone is not recommended, and it should be used in combination with the RMS wind direction difference, and the number of ambiguities, in order to control wind direction performance. To overcome this problem of not being able to describe the merits of the system in one number, a new Figure of Merit was developed which is explained in detail in [RD9].

## 9 Acronyms

C-band	Radar wavelength at about 5 cm
ECMWF	European center for medium-range weather forecasts
FoM	figure of merit
GMF	geophysical model function
IDL	interactive data language
IFARS	Institut für angewandte Fernerkundung
KNMI	Koninklijk Nederlands Meteorologisch Instituut (Royal Dutch Meteorological Institute)
Ku-band	Radar wavelength at about 2 cm
L1b	Level 1b product (geo-located, calibrated data in physical units)
LUT	look up table
MLE	maximum likelyhood estimator
MPI	message passing interface
NRCS	normalised radar cross-section
NRMS	normalised root mean square (see section 8)
RDx	reference document number x (see section 1)
RFSCAT	rotating fanbeam scatterometer
RMS	root mean square
WVC	wind vector cell, also called node