

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

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

task 3a report, FoM definition and validation  
by: dr. J. de Kloe, KNMI, and dr. ir. A.C.M. Stoffelen, KNMI,  
version 03-04-2003.

# Contents

<b>References</b>	<b>4</b>
<b>1 Summary</b>	<b>5</b>
<b>2 Introduction</b>	<b>6</b>
<b>3 Definition of the new Figure of Merit</b>	<b>7</b>
3.1 Probabilistic approach . . . . .	7
3.2 Resolution adjustment . . . . .	8
3.3 Implementation 1 . . . . .	8
3.4 Implementation 2 . . . . .	10
3.5 Correction to implementation 2 . . . . .	10
3.6 Implementation 3 . . . . .	10
3.7 Implementation 4 . . . . .	11
<b>4 Testing the new Figure of Merit</b>	<b>12</b>
4.1 Implementation 1 . . . . .	12
4.1.1 1 solution, exact calculation . . . . .	12
4.1.2 2 solutions, 180 degrees apart, exact calculation . . . . .	12
4.1.3 2 solutions, general case, exact calculation . . . . .	16
4.1.4 3 and more solutions, exact calculation . . . . .	17
4.1.5 Synthetic solutions . . . . .	20
4.1.6 Comparison with ERS data . . . . .	24
4.1.7 Comparison with Seawinds data . . . . .	24
4.2 Implementation 2 . . . . .	24
4.2.1 Synthetic solutions . . . . .	25
4.2.2 Comparison with Seawinds data . . . . .	27
4.2.3 Another small bug . . . . .	28
4.3 Implementation 3 . . . . .	29
4.3.1 1 solution . . . . .	29
4.3.2 2 solutions, 180 degrees apart, exact calculation . . . . .	29
4.3.3 2 solutions, general case, exact calculation . . . . .	30
4.3.4 3 and more solutions, exact calculation . . . . .	31
4.3.5 Synthetic solutions . . . . .	31
4.3.6 Comparison with ERS data . . . . .	34
4.3.7 Comparison with Seawinds data . . . . .	34
4.4 Implementation 4 . . . . .	34
4.4.1 Synthetic solutions . . . . .	35
4.4.2 Comparison with Seawinds data . . . . .	36
<b>5 Conclusions</b>	<b>37</b>
5.1 implementations 1 and 2 . . . . .	37
5.2 implementations 3 and 4 . . . . .	37
5.3 overall conclusion . . . . .	38



## References

- [1] “Optimisation of Rotating, Range-Gated Fanbeam Scatterometer for Wind Retrieval”, Task 1 Technical Report: Review of the requirements and scatterometer concept” MPB technologies Inc, October 17, 2000.
- [2] “Optimisation of Rotating, Range-Gated Fanbeam Scatterometer for Wind Retrieval”, Task 2a Report: GMF and wind field definition, and wind retrieval, KNMI, April 2002.
- [3] “Optimisation of Rotating, Range-Gated Fanbeam Scatterometer for Wind Retrieval”, Task 3b Report (results of the system optimisation), Ifars, 2002.
- [4] “Statement of guidance regarding how well satellite and in situ sensor capabilities meet WMO user requirements in several application areas”, WMO Satellite reports, 2001, Sat-26, Technical document WMO/TD no. 1052.
- [5] “Analysis methods for numerical weather prediction”, Lorenc, A.C., 1986, Quart J. R. Meteorol. Soc. 112, pp. 1177-1194.
- [6] “Scatterometry”, Stoffelen, A.C.M., Thesis University of Utrecht, 1998, <http://www.library.uu.nl/digiarchief/dip/diss/01840669/inhoud.htm>
- [7] “RFscat Note on the FoM”, Ad Stoffelen, 24-5-2002.
- [8] “Quality control and wind retrieval for SeaWinds, final report of the EUMETSAT Quikscat fellowship” M. Portabella and A. Stoffelen, KNMI Scientific report, WR-2002-01, section 4.2.
- [9] “Error modelling of scatterometer in-situ, and ECMWF model winds; A calibration refinement”, Ad Stoffelen, 1996, KNMI Technical report; TR-193, sec.5.2, p.10.

# 1 Summary

In this document a newly developed Figure of Merit (FoM), to be used to judge the performance of the RFSCAT simulator, is discussed and validated.

First this new FoM is defined (for two different implementations). The first with equal weights for each solution, the second with a weight depending on the angle of the solutions to each other.

However, at the end of the simulations a relevant bug was found in the implementation of the FoM. The two corrected implementations have therefore been added as implementation 3 and 4.

Then this FoM is tested in different ways. In a number of simple cases an exact calculation is done and compared to the numerical output of the FoM implementation. Then in a number of cases a number of synthetic wind solutions is used as input for the FoM implementation. The results for several cases are compared to each other and to real ERS and Seawinds data.

For 1 and 2 solutions the first implementation of the new Figure of Merit is very well understood. The exact calculations and the synthetic approach agree very well with each other. Also the real ERS data agrees very well with the results.

For the case of 3 and 4 solutions the implementation of the new Figure of Merit shows some differences with the exact solution. A reason for this has not been found.

Some difference has been found between implementation 1 and 2 of the FoM. Somewhat less difference was present between implementation 3 and 4. Adding different weights to the solutions while determining the distribution functions is of some relevance. This might be increased by taking the solution probability after inversion (residual) in stead of the sectors between the solutions as weights.

From the scans of the Standard Deviation (SD) of the first rank solution around the true wind, it is clear that the FoM has some useful properties. From the scans of the angle between the solutions it is clear that this does not give much variation of the FoM, which is not the intended behaviour.

Finally the results of real Seawinds data is given, which is significantly better than the synthetic calculations for 2, 3 or 4 solutions (for all implementations). This is explained by the fact that the background ECMWF wind is probably of better quality, than assumed in the calculation.

The overall conclusion is thus that this Figure of Merit is usefull, but not completely understood, and there certainly is room for improvement/tuning of the FoM.

## **2 Introduction**

For real scatterometer winds heuristic methods exist to determine their quality. These methods are based on experience and not fully objective. Furthermore it would be very time consuming to build this experience-based quality measure, for each of the scenarios that will be tested by the RFSCAT simulator.

Therefore we think it is appropriate to search for a more fundamental way of exploring wind quality.

The problem of determining quality is caused by the ambiguous nature of the wind direction; multiple solutions exist with varying probability. Ranking scores and closest solution to reference wind vector RMS differences are used for validation, but these need to be combined in a FoM in order to provide a reasonable quality indication.

In the following section we will derive an objective score based on probabilistic arguments.

### 3 Definition of the new Figure of Merit

The quality measure desired is dictated by a specification.

This specification is driven by the fact that the wind is largely a known quantity. The dynamic range of wind components is governed by a standard deviation of about 5.5 m/s [2], whereas the required accuracy is about 1.5 m/s [4]. This is a complication for designing an objective quality measure, i.e., ambiguities much outside the specification range are not bad, since these can be removed by using prior knowledge. However, those inside the specification range are clearly detrimental, as these can in real life not be distinguished from each other and may cause detrimental impacts.

To overcome this problem, an approach similar to that used in meteorological analysis is taken.

#### 3.1 Probabilistic approach

Lorenz [5] provides a probabilistic formalism for meteorological analysis. In this formalism probability density functions (PDF) are used to describe the physical observations and the model (a priori) background information.

Any physical observation is subject to random observational errors. In our case we know the true wind state  $\vec{v}_t$  and we can express the errors by:

$$P(\vec{v}_o | \vec{v} = \vec{v}_t) = P_o(\vec{v}_o - \vec{v}) \quad (1)$$

i.e. the probability for finding an observed windvector  $\vec{v}_o$ , for a given true windvector  $\vec{v}_t$ , is a function of the difference of both vectors.

A complication in our case is that the observed wind  $\vec{v}_o$  is not unique, but several ambiguities exist, that may be ranked in order of decreasing probability. However, over a distribution of  $\vec{v}_t$  we can obtain the distribution of  $\vec{v}_o$  and plot the PDF  $P_o(\vec{v}_o - \vec{v}_t)$  including all ambiguities.

Note that for several subdomains of  $\vec{v}$  and for different across-track cells this will be a varying multi-modal distribution.

The background information encapsulates our knowledge before the observation is taken. It results from observations in the past and knowledge about atmospheric flow ([6]; chapter I). The background that is used in meteorological analysis has a quality that is compatible with the specification by WMO of the required wind quality. It can be modelled by a Gaussian distribution in the wind vector components ([6], chapter IV):

$$P(\vec{v} = \vec{v}_t) = P_b(\vec{v}_b - \vec{v}) \propto \exp\left(-\frac{[\vec{v}_b - \vec{v}]^2}{2\sigma_b^2}\right) \quad (2)$$

The meteorological analysis resulting from the observations and the a priori background information incorporates both observations and background and is given by (see Eq (11) of [5]):

$$\begin{aligned} P_a(\vec{v}) &= P(\vec{v} = \vec{v}_t | \vec{v}_o) \\ &\propto P(\vec{v}_o | \vec{v} = \vec{v}_t) \times P(\vec{v} = \vec{v}_t) \\ &\propto P_o(\vec{v}_o - \vec{v}) \times \exp\left(-\frac{[\vec{v}_b - \vec{v}]^2}{2\sigma_b^2}\right) \end{aligned} \quad (3)$$

$P_a(\vec{v})$  incorporates the improved knowledge of the wind field with respect to the background, and represents the analysis probability. As such we can e.g. compute:

$$RMS(\vec{v}_a) = \sqrt{\int_v \int_u (\vec{v} - \vec{v}_t)^2 P_a(\vec{v}) dudv} \quad (4)$$

where  $\vec{v} = (u, v)$  is a decomposition of the wind vector in its west-east and south-north components respectively. This RMS should obviously be related to  $\sigma_b$  and be smaller then this to be useful (the maximum is  $\sigma_b$ ). Similarly, mean component, mean speed, or direction errors, or indeed other statistical modes to measure quality (RMS, bias, etc.) can be obtained through integration of the analysis wind distribution.

### 3.2 Resolution adjustment

For the implementation of equation 4 the value of  $\sigma_b$  is needed. For a scatterometer resolution of 50 km to be used by NWP this value is known to be 1.5 m/s [4]. To adjust its value to different resolutions the relation given in [1], page 24, sec.3.1.2.4, is used. This equation relates the expected standard deviation to the wind component variability spectrum for a given resolution  $r$ :

$$\sigma^2 = 2 \int_{2\pi/r}^{\infty} 0.25 \times 10^{-2} k^{-\frac{5}{3}} dk = 0.5 \times 10^{-2} \left[ -\frac{3}{2} k^{-\frac{2}{3}} \right]_{2\pi/r}^{\infty} = \frac{3}{4} \times 10^{-2} \left( \frac{2\pi}{r} \right)^{-\frac{2}{3}} \quad (5)$$

with  $r$  in [m],  $\sigma$  in [m/s] and  $k$  in [1/m]. Using this, the ratio between two SD values for two different resolutions,  $r_1$  and  $r_2$ , can be expressed as:

$$\frac{\sigma_{r_1}^2}{\sigma_{r_2}^2} = \left( \frac{2\pi}{r_1} \right)^{-\frac{2}{3}} \left( \frac{r_2}{2\pi} \right)^{-\frac{2}{3}} = \left( \frac{r_2}{r_1} \right)^{-\frac{2}{3}} \quad (6)$$

this is used to transform the known value for 50 km resolution, into the SD of the wanted resolution  $r_1$  (now in [km]):

$$\sigma_{r_1} = \sigma_{r_2} \left( \frac{r_1}{r_2} \right)^{\frac{1}{3}} = 1.5 \left( \frac{r_1}{50} \right)^{\frac{1}{3}} \text{ [m/s]} \quad (7)$$

### 3.3 Implementation 1

The Figure of Merit is based on the RMS defined by eq. 4. For both wind components  $u$  and  $v$ , a score is calculated as follows:

- the distribution function for the difference between found and reference wind is determined ( $P_o(\Delta u)$ ). To do this, 201 bins between -50 and +50 m/s with a binsize of 0.5 m/s are used. At first all found solutions have equal weight, and the vectors are just counted in each bin. So in case of 1 solution, the weight is 1, in case of 2 solutions the weight of each solution is 1/2, in case of 3 solutions the weight of each solution is 1/3, and in case of 4 solutions the weight of each solution is 1/4. Then the distribution is normalised by dividing all bins by the total number of nodes, in order to have an integral of 1.



- then a background distribution is calculated ( $P_b(\Delta u)$ ), using a Gaussian distribution with a halfwidth of the expected background SD of  $\sigma_b$ .
- then the found distribution  $P_o(\Delta u)$  and the background distribution  $P_b(\Delta u)$  are multiplied and normalised to 1 to get an analysis distribution  $P_{a,u}(\Delta u)$ , conform eq. 4.
- then this analysis distribution is used to calculate a weighted-RMS of  $\Delta u$  using:

$$\text{RMS}_u = \sqrt{\sum_{bins} \Delta u^2 P_{a,u}(\Delta u)} \quad (8)$$

(the summation is done over all bins) and in a similar way for the v-component.

- finally the found RMS value is divided by the expected SD to get the score for this windcomponent, so:  $\text{score}_u = \text{RMS}_u / \sigma_b$  and  $\text{score}_v = \text{RMS}_v / \sigma_b$ .

In the determination of  $P_o$  the weight of the solutions corresponds to their probability (which we don't know). In reality not all solutions have equal probability. Portabella and Stoffelen [8] developed methods to compute these probabilities for Seawinds but since these methods are not generic an additional factor is needed. This is resolved here by using the  $\text{score}_r$ . An improvement of the FoM would be in the correct determination of these probabilities, which would remove the need for  $\text{score}_r$ .

The ranking score is determined in the following way (note that  $P_o$  is used rather than  $P_a$  which was suggested in [7]. This was a mistake, as is further explained in sections 3.6 and 4.3):

- the 2-dimensional observed probability  $P_{o2}(\Delta u, \Delta v) = P_{o,u}(\Delta u)P_{o,v}(\Delta v)$  is summed for all solutions, using only the first rank solution:

$$r_1 = \sum_{1st\ rank} P_{o2}(\Delta u_1, \Delta v_1) \quad (9)$$

- then a similar summation is done using the higher ranked solutions (only if they are existing):

$$r_r = \sum_{j=2nd,3rd,4th\ rank} P_{o2}(\Delta u_j, \Delta v_j) \quad (10)$$

- then a ranking score is calculated from these numbers:

$$\text{score}_r = \frac{2r_r}{r_1 + r_r} \quad (11)$$

Finally the 3 ranking scores are combined as follows:

$$\text{FoM} = 0.4 \times \text{score}_u + 0.4 \times \text{score}_v + 0.2 \times \text{score}_r \quad (12)$$

### 3.4 Implementation 2

Now in step 1, while determining  $P_o(\Delta u)$ , different weights for different solutions are used. As weight the sector-angle is used for each solution. A sector-angle is defined as the angle between the middle of the current and the clock-wise next solution, and the middle of the current and the counter-clock-wise next solution.

In cases with 1 or 2 solutions nothing changes, since for 1 solution the weight is not relevant, and for 2 solutions the sector-angles for both solutions are always 180 degrees, so always equal.

In cases with 3 or 4 solutions this becomes important. The effect will be that solutions that are close together get less weight, and a solution that is well separated from the other solutions gets more weight.

This assumption for the solution probability will be an improvement to uniform solutions weighting, as was used in implementation 1, but it still ignores system characteristics that determine the real solution probabilities (see [8]). Therefore usage of  $\text{score}_r$  is still necessary.

This case has not been tested analytically, because of the resulting complexity of the equations.

This implementation is the one used in the Task 3b report to evaluate the performance of the simulated RFSCAT instrument.

### 3.5 Correction to implementation 2

After the conclusion of all calculations, during the writing of this report, a small bug was discovered in the FoM-code which is given in appendix A. The calculation of the  $r_1$  score should also include the cases in which just one solution is found. Therefore the line

```
IF (numsol(i) ge 2) THEN $
```

before the calculation of  $r_1$  should be replaced by:

```
IF (numsol(i) ge 1) THEN $
```

For this purpose a switch called “`apply_correction_to_implementation_2`” has been added to the code.

The impact of the bug on the reported results will be almost unnoticeable since the case with just 1 solution almost never occurs. Only in simulations with a polarimetric measurement this may happen.

In the synthetic results for 1 solution the  $\text{score}_r$  is always zero, so also in this case there is no effect.

The effect of the bug will be that the FoM seems a little worse than it actually should be, since the  $r_1$  score is too low (a higher  $r_1$  improves the FoM).

This correction has not been applied to the synthetic calculations described in section 4.

### 3.6 Implementation 3

After all the simulations needed for this project had been concluded, another bug was found in the Figure of Merit routine. In equations 9 and 10  $P_{o,2}$  was used, which is now corrected to  $P_{a,2}$ . This correction to implementation 1 is called implementation 3. Since all results

were already produced it was decided to study the properties of the FoM with and without bug, in order to be able to decide how important it is.

### **3.7 Implementation 4**

The bug mentioned in section 3.6 also applies to implementation 2. This correction to implementation 2 is called implementation 4. So in this case  $P_{a,2}$  is used for the score<sub>r</sub> calculation, and also different weights are used for the  $P_o$  calculation.

## 4 Testing the new Figure of Merit

### 4.1 Implementation 1

#### 4.1.1 1 solution, exact calculation

For simple cases an exact calculation can be done for the FoM scheme. If for all nodes just 1 solution is found, with a given standard deviation and no systematic error, then the distribution functions become:

$$P_o(\Delta u) = c_1 \exp\left(\frac{-\Delta u^2}{2\sigma_1^2}\right) \quad (13)$$

$$P_b(\Delta u) = c_2 \exp\left(\frac{-\Delta u^2}{2\sigma_b^2}\right) \quad (14)$$

with  $\sigma_1 = \sigma_b = 1.5$  m/s,  $\Delta u \equiv u_t - u_1$ .  $u_t$  is the u-component of the true (reference) wind, and  $u_1$  the u-component of the first (and in this case only) wind solution. From the normalisation the values of  $c_1$  and  $c_2$  can be found.

Using this, the analysis distribution becomes

$$P_{a,u} = P_o(\Delta u) \times P_b(\Delta u) = c_3 \exp\left(\frac{-\Delta u^2}{2\sigma_1^2} + \frac{-\Delta u^2}{2\sigma_b^2}\right) = c_3 \exp\left(\frac{-\Delta u^2}{2\sigma_a^2}\right) \quad (15)$$

with again  $c_3$  determined by the normalisation, and  $\frac{1}{\sigma_a^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_b^2}$ .

Now the integral needed for the RMS score becomes:

$$\begin{aligned} \text{RMS}_u^2 &= \int_{-\infty}^{\infty} \Delta u^2 P_a(\Delta u) d\Delta u = \int_{-\infty}^{\infty} c_3 \Delta u^2 \exp\left(\frac{-\Delta u^2}{2\sigma_a^2}\right) d\Delta u = \\ & \left[ -c_3 \sigma_a^2 \Delta u \exp\left(\frac{-\Delta u^2}{2\sigma_a^2}\right) \right]_{\Delta u=-\infty}^{\infty} + c_3 \sigma_a^2 \int_{-\infty}^{\infty} \exp\left(\frac{-\Delta u^2}{2\sigma_a^2}\right) d\Delta u = \\ & 0 + \sigma_a^2 \int_{-\infty}^{\infty} c_3 \exp\left(\frac{-\Delta u^2}{2\sigma_a^2}\right) d\Delta u = \sigma_a^2 \int_{-\infty}^{\infty} P_a(\Delta u) d\Delta u = \sigma_a^2 \end{aligned} \quad (16)$$

because the last integral equals one, due to the normalisation of  $P_a(\Delta u)$ . So

$$\text{RMS}_u = \sqrt{\sigma_a^2} = \frac{1}{\sqrt{\frac{1}{\sigma_1^2} + \frac{1}{\sigma_b^2}}} = \frac{\sigma_b}{\sqrt{2}} \quad (17)$$

and  $\text{score}_u = \frac{\text{RMS}_u}{\sigma_b} = \frac{1}{\sqrt{2}} = 0.707$ . The same value is found for  $\text{score}_v$ .

For just 1 solution  $\text{score}_r = 0$ , so the FoM becomes  $0.8 \times \frac{1}{\sqrt{2}} = 0.566$ . This is the numerical value for the Figure of Merit for a standard deviation of the reference windfield of  $\sigma_b = 1.5$  m/s, and  $\sigma_1 = \sigma_b$ .

#### 4.1.2 2 solutions, 180 degrees apart, exact calculation

In case of a second ambiguous solution also an exact calculation can be done, if a position of the 2 solutions relative to each other is assumed. The simplest case is the ERS geometry,

in which the 2 solutions have a similar wind speed and are always about 180 degrees apart. In this case the observed probability becomes:

$$P_o(\Delta u) = P_{o,1}(\Delta u) + P_{o,2}(\Delta u) \quad (18)$$

For the parallel solution the same distribution is used:

$$P_{o,1}(\Delta u_1) = c_1 \exp\left(\frac{-\Delta u_1^2}{2\sigma_1^2}\right) \quad (19)$$

With  $\Delta u_1 \equiv u_t - u_1$ . Again  $\sigma_1 = \sigma_b = 1.5$  m/s is taken.

But the SD for the second solution is very different now, and can be calculated as follows: (using  $\Delta u_2 \equiv u_t - u_2 = u_t + u_1$  and  $u_2 = -u_1$ )

$$\begin{aligned} P_{o,2}(\Delta u_2) &= \int_{u_t=-\infty}^{\infty} \int_{u_1=-\infty}^{\infty} P_t(u_t) P_{o,1}(\Delta u_1) \delta(u_1 + u_2) du_1 du_t = \\ &= \int_{u_t=-\infty}^{\infty} \int_{u_1=-\infty}^{\infty} P_t(u_t) P_{o,1}(u_t - u_1) \delta(u_1 + u_2) du_1 du_t = \\ &= \int_{u_t=-\infty}^{\infty} P_t(u_t) P_{o,1}(u_t + u_2) du_t = \int_{u_t=-\infty}^{\infty} P_t(u_t) P_{o,1}(2u_t - \Delta u_2) du_t = \\ &= \int_{u_t=-\infty}^{\infty} \exp\left(\frac{-u_t^2}{2\sigma_w^2}\right) \exp\left(\frac{-(2u_t - \Delta u_2)^2}{2\sigma_1^2}\right) du_t = \\ &= \int_{u_t=-\infty}^{\infty} \exp\left(\frac{-u_t^2}{2\sigma_w^2}\right) \exp\left(\frac{-(u_t - \frac{1}{2}\Delta u_2)^2}{2(\frac{1}{2}\sigma_1)^2}\right) du_t = \\ &= \exp\left(\frac{-\frac{1}{4}\Delta u_2^2}{2\left(\frac{1}{4}\right)\sigma_2^2}\right) = \exp\left(\frac{-\Delta u_2^2}{2\sigma_2^2}\right) \end{aligned} \quad (20)$$

with  $\frac{1}{4}\sigma_2^2 \equiv \sigma_w^2 + \frac{1}{4}\sigma_1^2$  so  $\sigma_2^2 = 4\sigma_w^2 + \sigma_1^2$ , and  $\sigma_w = 5.5$  m/s.

In short notation, using expectation values, this can also be written as: (using  $\Delta u_2 \equiv u_t - u_2 = u_t + u_1$  and  $u_2 = -u_1$ , and uncorrelated  $u_t$  and  $\Delta u_1$ )

$$\begin{aligned} \sigma_2^2 &= \langle \Delta u_2^2 \rangle = \langle (u_t - u_2)^2 \rangle = \langle (u_t + u_1)^2 \rangle = \\ &= \langle (2u_t - \Delta u_1)^2 \rangle = \langle 4u_t^2 - 4u_t\Delta u_1 + \Delta u_1^2 \rangle = \\ &= 4 \langle u_t^2 \rangle + 0 + \langle \Delta u_1^2 \rangle = 4\sigma_w^2 + \sigma_1^2 \end{aligned} \quad (21)$$

So the distribution of the anti-parallel solutions is also just Gaussian, but with a width of  $\sigma_2 = \sqrt{4\sigma_w^2 + \sigma_1^2} = 11.10$  m/s. So:

$$P_o(\Delta u) = c_1 \exp\left(\frac{-\Delta u^2}{2\sigma_1^2}\right) + c_2 \exp\left(\frac{-\Delta u^2}{2\sigma_2^2}\right) \quad (22)$$

Since the amount of parallel and anti-parallel solutions is taken to be equal in this 2 solution case, the two constants  $c_1$  and  $c_2$  in equation 22 can be found by normalisation of each of the two components of  $P_o$  to 0.5. This gives:  $c_1 = \frac{0.5}{\sigma_1\sqrt{2\pi}}$  and  $c_2 = \frac{0.5}{\sigma_2\sqrt{2\pi}}$ .

Now again  $P_a$  can be calculated:

$$\begin{aligned}
P_a(\Delta u) &= P_o(\Delta u) \times P_b(\Delta u) = \\
c_4 \exp\left(\frac{-\Delta u^2}{2}\left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_b^2}\right)\right) &+ c_5 \exp\left(\frac{-\Delta u^2}{2}\left(\frac{1}{\sigma_2^2} + \frac{1}{\sigma_b^2}\right)\right) = \\
c_4 \exp\left(\frac{-\Delta u^2}{2\sigma_{a,1}^2}\right) &+ c_5 \exp\left(\frac{-\Delta u^2}{2\sigma_{a,2}^2}\right)
\end{aligned} \tag{23}$$

with  $\frac{1}{\sigma_{a,1}^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_b^2}$  and  $\frac{1}{\sigma_{a,2}^2} = \frac{1}{\sigma_2^2} + \frac{1}{\sigma_b^2}$ .

The constants  $c_4$  and  $c_5$  again follow from normalisation, and have the same ratio as  $c_1$  and  $c_2$  so  $\frac{c_4}{c_5} = \frac{c_1}{c_2} = \frac{\sigma_2}{\sigma_1}$  so  $c_5 = \frac{\sigma_1}{\sigma_2}c_4$ . Integration of  $P_a$  gives:  $c_4\sigma_{a,1}\sqrt{2\pi} + c_5\sigma_{a,2}\sqrt{2\pi} \equiv 1$  so  $c_4\sigma_{a,1} + c_4\frac{\sigma_1}{\sigma_2}\sigma_{a,2} = \frac{1}{\sqrt{2\pi}}$  so:

$$c_4 = \frac{1}{\sqrt{2\pi}(\sigma_{a,1} + \frac{\sigma_1}{\sigma_2}\sigma_{a,2})} \tag{24}$$

and:

$$c_5 = \frac{1}{\sqrt{2\pi}(\sigma_{a,2} + \frac{\sigma_2}{\sigma_1}\sigma_{a,1})} \tag{25}$$

Now the integration for  $\text{RMS}_u$  becomes:

$$\begin{aligned}
\text{RMS}_u^2 &= \int_{\Delta u=-\infty}^{\infty} \Delta u^2 P_a(\Delta u) d\Delta u = \\
\int_{\Delta u=-\infty}^{\infty} \left\{ c_4 \Delta u^2 \exp\left(\frac{-\Delta u^2}{2\sigma_{a,1}^2}\right) + c_5 \Delta u^2 \exp\left(\frac{-\Delta u^2}{2\sigma_{a,2}^2}\right) \right\} d\Delta u &= \\
c_4 \sigma_{a,1}^2 \sigma_{a,1} \sqrt{2\pi} + c_5 \sigma_{a,2}^2 \sigma_{a,2} \sqrt{2\pi} &= \\
\frac{\sigma_2 \sigma_{a,1}^3 + \sigma_1 \sigma_{a,2}^3}{\sigma_2 \sigma_{a,1} + \sigma_1 \sigma_{a,2}} &= \frac{13.247 + 4.927}{11.775 + 2.230} = 1.298
\end{aligned} \tag{26}$$

so  $\text{RMS}_u = 1.139$  and  $\text{score}_u = 0.759$ .

The ranking score for this ERS case can be calculated using eqs. 9 and 10. The 2-dimensional function  $P_{o,2}(\Delta u, \Delta v)$  is probed with the frequency with which vectors occur for each combination of  $\Delta u$  and  $\Delta v$ . This probing frequency  $P_{p,1}$  for the 1st rank solution is given by:

$$\begin{aligned}
P_{p,1}(\Delta u, \Delta v) &= n_{\text{nodes}} \times f_{\text{norm},1}^2 \times [QP_{o,1}(\Delta u) + (1-Q)P_{o,2}(\Delta u)] \\
&\times [QP_{o,1}(\Delta v) + (1-Q)P_{o,2}(\Delta v)]
\end{aligned} \tag{27}$$

In which  $Q$  is the fraction of cases in which the 1st rank solution is right.

The normalisation factor  $f_{\text{norm},1}$  is determined by the fact that  $f_{\text{norm},1} \times [QP_{o,1} + (1-Q)P_{o,2}]$  should be normalised to 1. So  $f_{\text{norm},1} \times [Qc_1\sigma_1\sqrt{2\pi} + (1-Q)c_2\sigma_2\sqrt{2\pi}] = 1$ . After inserting the expressions for  $c_1$  and  $c_2$ , this gives  $f_{\text{norm},1} = 2$ .

The probing frequency  $P_{p,2}$  for the 2nd rank solution is given by:

$$\begin{aligned}
P_{p,2}(\Delta u, \Delta v) &= n_{\text{nodes}} \times f_{\text{norm},2}^2 \times [(1-Q)P_{o,1}(\Delta u) + QP_{o,2}(\Delta u)] \\
&\times [(1-Q)P_{o,1}(\Delta v) + QP_{o,2}(\Delta v)]
\end{aligned} \tag{28}$$

Again the normalisation factor  $f_{norm,2}$  is determined by the fact that:  $f_{norm,2} \times [(1 - Q)P_{o,1} + QP_{o,2}]$  should be normalised to 1. This gives  $f_{norm,2} = 2$ . Using this the  $r_1$  score becomes:

$$\begin{aligned}
r_1 &= \int_{\Delta u=-\infty}^{\infty} \int_{\Delta v=-\infty}^{\infty} P_{p,1}(\Delta u, \Delta v) \times P_{o,2}(\Delta u, \Delta v) d\Delta v d\Delta u \\
&= \mathbf{n}_{nodes} \times f_{norm,1}^2 \times \\
&\left( \int_{\Delta u=-\infty}^{\infty} [QP_{o,1}(\Delta u) + (1 - Q)P_{o,2}(\Delta u)] \times [P_{o,1}(\Delta u) + P_{o,2}(\Delta u)] d\Delta u \right)^2 \\
&= \mathbf{n}_{nodes} \times f_{norm,1}^2 \times \\
&\left( \int_{\Delta u=-\infty}^{\infty} [QP_{o,1}^2(\Delta u) + (1 - Q)P_{o,2}^2(\Delta u) + P_{o,1}(\Delta u)P_{o,2}(\Delta u)] d\Delta u \right)^2 \\
&= \mathbf{n}_{nodes} \times f_{norm,1}^2 \times \left( \int_{\Delta u=-\infty}^{\infty} \left[ Qc_1^2 \exp\left(\frac{-\Delta u^2}{\sigma_1^2}\right) + (1 - Q)c_2^2 \exp\left(\frac{-\Delta u^2}{\sigma_2^2}\right) + \right. \right. \\
&\quad \left. \left. c_1c_2 \exp\left(\frac{-\Delta u^2}{2\sigma_1^2} + \frac{-\Delta u^2}{2\sigma_2^2}\right) \right] d\Delta u \right)^2 \\
&= \mathbf{n}_{nodes} \times f_{norm,1}^2 \times \left( Qc_1^2\sigma_1\sqrt{\pi} + (1 - Q)c_2^2\sigma_2\sqrt{\pi} + c_1c_2\sigma_s\sqrt{2\pi} \right)^2 \tag{29}
\end{aligned}$$

with  $\sigma_s = \frac{1}{\sqrt{\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}}}$ .

So, after inserting the values for  $c_1$ ,  $c_2$ , and  $\sigma_s$  this becomes:

$$r_1 = \frac{\mathbf{n}_{nodes}}{16\pi} \times \left( \frac{Q}{\sigma_1} + \frac{(1 - Q)}{\sigma_2} + \frac{\sqrt{2}}{\sqrt{\sigma_1^2 + \sigma_2^2}} \right)^2 \tag{30}$$

Similarly,  $r_r$  follows from:

$$\begin{aligned}
r_r &= \int_{\Delta u=-\infty}^{\infty} \int_{\Delta v=-\infty}^{\infty} P_{p,2}(\Delta u, \Delta v) \times P_{o,2}(\Delta u, \Delta v) d\Delta v d\Delta u \\
&= \mathbf{n}_{nodes} \times f_{norm,2}^2 \times \\
&\left( \int_{\Delta u=-\infty}^{\infty} [(1 - Q)P_{o,1}(\Delta u) + QP_{o,2}(\Delta u)] \times \right. \\
&\quad \left. [P_{o,1}(\Delta u) + P_{o,2}(\Delta u)] d\Delta u \right)^2 \\
&= \mathbf{n}_{nodes} \times f_{norm,2}^2 \times \\
&\left( \int_{\Delta u=-\infty}^{\infty} [(1 - Q)P_{o,1}^2(\Delta u) + QP_{o,2}^2(\Delta u) + \right. \\
&\quad \left. P_{o,1}(\Delta u)P_{o,2}(\Delta u)] d\Delta u \right)^2 \\
&= \mathbf{n}_{nodes} \times f_{norm,2}^2 \times \\
&\left( \int_{\Delta u=-\infty}^{\infty} \left[ (1 - Q)c_1^2 \exp\left(\frac{-\Delta u^2}{\sigma_1^2}\right) + Qc_2^2 \exp\left(\frac{-\Delta u^2}{\sigma_2^2}\right) + \right. \right. \\
&\quad \left. \left. c_1c_2 \exp\left(\frac{-\Delta u^2}{2\sigma_1^2} + \frac{-\Delta u^2}{2\sigma_2^2}\right) \right] d\Delta u \right)^2 \\
&= \mathbf{n}_{nodes} \times 4 \times \\
&\left( (1 - Q)c_1^2\sigma_1\sqrt{\pi} + Qc_2^2\sigma_2\sqrt{\pi} + c_1c_2\sigma_s\sqrt{2\pi} \right)^2 \tag{31}
\end{aligned}$$

So, after inserting the values for  $c_1$ ,  $c_2$ , and  $\sigma_s$  this becomes:

$$r_r = \frac{n_{\text{nodes}}}{16\pi} \times \left( \frac{(1-Q)}{\sigma_1} + \frac{Q}{\sigma_2} + \frac{\sqrt{2}}{\sqrt{\sigma_1^2 + \sigma_2^2}} \right)^2 \quad (32)$$

In an ideal case for  $n_{\text{sol}}=2$ ,  $Q=1$  which gives:  $r_1 = 0.629 \times \frac{n_{\text{nodes}}}{16\pi}$  and  $r_r = 0.0468 \times \frac{n_{\text{nodes}}}{16\pi}$ , so  $\text{score}_r = 0.139$ .

However, for the ERS case both ranks have (almost) equal probability (i.e.  $Q = 0.5$ ). From this it follows that  $r_1 = r_r$ . So equation 11 gives  $\text{score}_r = 1.0$ .

Combining the scores for 2 solutions, the FoM becomes  $0.8 \times 0.759 + 0.2 \times 1.0 = 0.807$  for  $Q=0.5$ , or  $\text{FoM}=0.635$  for  $Q=1$ .

#### 4.1.3 2 solutions, general case, exact calculation

A similar calculation for 2 solutions, in which the second solution is uncorrelated to the first and best solution can be done. This is not very realistic, since in most cases there will be at least a correlation in wind speed between the different solutions of a node, so it gives an upper limit of the FoM (i.e. an estimate for its worst possible value).

First we need the probability for the second solution, which in this case has just the same SD as the reference wind:

$$P_2(u) = P_t(u) = c \exp\left(\frac{-u^2}{2\sigma_w^2}\right) \quad (33)$$

From this we find:

$$\begin{aligned} P_{o,2}(\Delta u_2) &= \int_{u_t=-\infty}^{\infty} \int_{u_2=-\infty}^{\infty} P_t(u_t) P_2(u_2) \delta(\Delta u_2 - u_t + u_2) du_2 du_t = \\ &= \int_{u_t=-\infty}^{\infty} P_t(u_t) P_2(u_t - \Delta u_2) du_t = \\ &= \int_{u_t=-\infty}^{\infty} c \exp\left(\frac{-u_t^2}{2\sigma_w^2}\right) \exp\left(\frac{-(u_t - \Delta u_2)^2}{2\sigma_w^2}\right) du_t = \\ &= c_2 \exp\left(\frac{-\Delta u_2}{2\sigma_2^2}\right) \end{aligned} \quad (34)$$

with  $\sigma_2^2 = 2\sigma_w^2$ .

This again can also be calculated using expectation values:

$$\sigma_2^2 = \langle \Delta u_2^2 \rangle = \langle (u_t - u_2)^2 \rangle = \langle u_2^2 - 2u_t u_2 + u_t^2 \rangle = \langle u_2^2 \rangle + 0 + \langle u_t^2 \rangle = 2\sigma_w^2 \quad (35)$$

This gives a SD of  $\sigma_2 = \sqrt{2}\sigma_w = 7.778$  m/s.

Now, using eq. 26, the calculation for  $\text{RMS}_u$  becomes:

$$\text{RMS}_u^2 = \frac{\sigma_2 \sigma_{a,1}^3 + \sigma_1 \sigma_{a,2}^3}{\sigma_2 \sigma_{a,1} + \sigma_1 \sigma_{a,2}} = \frac{9.281 + 4.793}{8.250 + 2.209} = 1.346 \quad (36)$$

so  $\text{RMS}_u = 1.160$  and  $\text{score}_u = 0.773$ , and the FoM (again assuming  $Q=0.5$  and thus  $\text{score}_r=1.0$ ) becomes  $0.8 \times 0.773 + 0.2 \times 1.0 = 0.818$ . For  $Q=1$  the  $r_1$  and  $r_r$  scores are calculated using equations 30 and 32. This gives:  $r_1 = 0.714$  and  $r_r = 0.094$ , so  $\text{score}_r = 0.233$  and  $\text{FoM} = 0.665$ .



#### 4.1.4 3 and more solutions, exact calculation

In case of three or more ambiguous solutions an exact calculation can also easily be done (for implementation 1). The ambiguous solutions are again assumed to be totally uncorrelated to each other, and to the first and best solution (i.e. taking again an estimate for its worst possible value).

In case of 3 solutions the only change is the ratio between the two constants  $c_1$  and  $c_2$ . The number of 1st rank solutions is now only one third of the total amount of solutions (assuming that all solutions except the first have an identical distribution). This gives:  $c_1 = \frac{1}{3} \frac{1}{\sigma_1 \sqrt{2\pi}}$  and  $c_2 = \frac{2}{3} \frac{1}{\sigma_2 \sqrt{2\pi}}$ .

More general, for  $n_{\text{sol}}$  solutions, and assuming that all solutions except the first rank have a distribution of  $P_{o,2}$ , :  $c_1 = \frac{1}{n_{\text{sol}} \sigma_1 \sqrt{2\pi}}$  and  $c_2 = \frac{n_{\text{sol}} - 1}{n_{\text{sol}} \sigma_2 \sqrt{2\pi}}$ .

Using this general result for  $c_1$  and  $c_2$  equations 23 up to 26 can be rewritten.

Now  $P_a$  still has the same form as for 2 solutions (see eq. 23):

$$\begin{aligned} P_a(\Delta u) &= P_o(\Delta u) \times P_b(\Delta u) = \\ c_4 \exp\left(\frac{-\Delta u^2}{2}\left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_b^2}\right)\right) &+ c_5 \exp\left(\frac{-\Delta u^2}{2}\left(\frac{1}{\sigma_2^2} + \frac{1}{\sigma_b^2}\right)\right) = \\ c_4 \exp\left(\frac{-\Delta u^2}{2\sigma_{a,1}^2}\right) &+ c_5 \exp\left(\frac{-\Delta u^2}{2\sigma_{a,2}^2}\right) \end{aligned} \quad (37)$$

with  $\frac{1}{\sigma_{a,1}^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_b^2}$  and  $\frac{1}{\sigma_{a,2}^2} = \frac{1}{\sigma_2^2} + \frac{1}{\sigma_b^2}$ .

The constants  $c_4$  and  $c_5$  again follow from normalisation, and have the same ratio as  $c_1$  and  $c_2$ , but now the changed general expressions for  $c_1$  and  $c_2$  are used. So  $\frac{c_4}{c_5} = \frac{c_1}{c_2} = \frac{\sigma_2}{(n_{\text{sol}} - 1)\sigma_1}$  so  $c_5 = \frac{(n_{\text{sol}} - 1)\sigma_1}{\sigma_2} c_4$ . Integration of  $P_a$  gives:  $c_4 \sigma_{a,1} \sqrt{2\pi} + c_5 \sigma_{a,2} \sqrt{2\pi} \equiv 1$  so  $c_4 \sigma_{a,1} + c_4 \frac{(n_{\text{sol}} - 1)\sigma_1}{\sigma_2} \sigma_{a,2} = \frac{1}{\sqrt{2\pi}}$  so equations 24 and 25 change into:

$$c_4 = \frac{1}{\sqrt{2\pi} \left( \sigma_{a,1} + \frac{(n_{\text{sol}} - 1)\sigma_1}{\sigma_2} \sigma_{a,2} \right)} \quad (38)$$

and:

$$c_5 = \frac{1}{\sqrt{2\pi} \left( \sigma_{a,2} + \frac{\sigma_2}{(n_{\text{sol}} - 1)\sigma_1} \sigma_{a,1} \right)} \quad (39)$$

Now the integration for  $\text{RMS}_u$  becomes:

$$\begin{aligned} \text{RMS}_u^2 &= \int_{\Delta u = -\infty}^{\infty} \Delta u^2 P_a(\Delta u) d\Delta u = \\ \int_{\Delta u = -\infty}^{\infty} \left\{ c_4 \Delta u^2 \exp\left(\frac{-\Delta u^2}{2\sigma_{a,1}^2}\right) + c_5 \Delta u^2 \exp\left(\frac{-\Delta u^2}{2\sigma_{a,2}^2}\right) \right\} d\Delta u &= \\ c_4 \sigma_{a,1}^2 \sigma_{a,1} \sqrt{2\pi} + c_5 \sigma_{a,2}^2 \sigma_{a,2} \sqrt{2\pi} &= \\ \frac{\sigma_2 \sigma_{a,1}^3 + \sigma_1 \sigma_{a,2}^3 (n_{\text{sol}} - 1)}{\sigma_2 \sigma_{a,1} + \sigma_1 \sigma_{a,2} (n_{\text{sol}} - 1)} & \end{aligned} \quad (40)$$

With  $n_{\text{sol}}=3$  the  $\text{RMS}_u$  becomes:

$$\text{RMS}_u^2 = \frac{\sigma_2 \sigma_{a,1}^3 + 2\sigma_1 \sigma_{a,2}^3}{\sigma_2 \sigma_{a,1} + 2\sigma_1 \sigma_{a,2}} = \frac{9.281 + 2 \times 4.793}{8.250 + 2 \times 2.209} = 1.489 \quad (41)$$

So  $\text{score}_u = 0.814$ .

The ranking score for this case is calculated in a similar way as was done for the 2 solution case. Again eqs. 9 and 10 are used.

The 2-dimensional function  $P_{o,2}(\Delta u, \Delta v)$  is again probed with the frequency with which vectors occur for each combination of  $\Delta u$  and  $\Delta v$ . This probing frequency  $P_{p,1}$  for the 1st rank solution is identical to the frequency for the 2 solution case, and is given by:

$$P_{p,1}(\Delta u, \Delta v) = n_{\text{nodes}} \times f_{\text{norm},1}^2 \times [QP_{o,1}(\Delta u) + (1-Q)P_{o,2}(\Delta u)] \times [QP_{o,1}(\Delta v) + (1-Q)P_{o,2}(\Delta v)] \quad (42)$$

In which  $Q$  is the fraction of cases in which the 1st rank solution is right.

The normalisation factor  $f_{\text{norm},1}$  is determined by the fact that  $f_{\text{norm},1} \times [QP_{o,1} + (1-Q)P_{o,2}]$  should be normalised to 1. So  $f_{\text{norm},1} \times [Qc_1\sigma_1\sqrt{2\pi} + (1-Q)c_2\sigma_2\sqrt{2\pi}] = 1$ . After inserting the expressions for  $c_1$  and  $c_2$ , this gives  $f_{\text{norm},1} = \frac{n_{\text{sol}}}{Q+(1-Q)(n_{\text{sol}}-1)}$ .

The probing frequency  $P_{p,2,3,4}$  for the 2nd, 3rd and 4th rank solution is given by:

$$P_{p,2,3,4}(\Delta u, \Delta v) = n_{\text{nodes}} \times (n_{\text{sol}} - 1) \times f_{\text{norm},2}^2 \times [(1-Q)P_{o,1}(\Delta u) + (n_{\text{sol}} - 1 + Q - 1)P_{o,2}(\Delta u)] \times [(1-Q)P_{o,1}(\Delta v) + (n_{\text{sol}} - 1 + Q - 1)P_{o,2}(\Delta v)] \quad (43)$$

Again the normalisation factor  $f_{\text{norm},2}$  is determined by the fact that:  $f_{\text{norm},2} \times [(1-Q)P_{o,1} + (n_{\text{sol}} - 1 + Q - 1)P_{o,2}]$  should be normalised to 1.

This gives  $f_{\text{norm},2} = \frac{n_{\text{sol}}}{(1-Q)+(n_{\text{sol}}-1+Q-1)(n_{\text{sol}}-1)}$ .

Using this the  $r_1$  score becomes:

$$\begin{aligned} r_1 &= \int_{\Delta u=-\infty}^{\infty} \int_{\Delta v=-\infty}^{\infty} P_{p,1}(\Delta u, \Delta v) \times P_{o,2}(\Delta u, \Delta v) d\Delta v d\Delta u \\ &= n_{\text{nodes}} \times f_{\text{norm},1}^2 \times \left( \int_{\Delta u=-\infty}^{\infty} [QP_{o,1}(\Delta u) + (1-Q)P_{o,2}(\Delta u)] \times [P_{o,1}(\Delta u) + P_{o,2}(\Delta u)] d\Delta u \right)^2 \\ &= n_{\text{nodes}} \times f_{\text{norm},1}^2 \times \left( \int_{\Delta u=-\infty}^{\infty} [QP_{o,1}^2(\Delta u) + (1-Q)P_{o,2}^2(\Delta u) + P_{o,1}(\Delta u)P_{o,2}(\Delta u)] d\Delta u \right)^2 \\ &= n_{\text{nodes}} \times f_{\text{norm},1}^2 \times \left( \int_{\Delta u=-\infty}^{\infty} \left[ Qc_1^2 \exp\left(\frac{-\Delta u^2}{\sigma_1^2}\right) + (1-Q)c_2^2 \exp\left(\frac{-\Delta u^2}{\sigma_2^2}\right) + c_1c_2 \exp\left(\frac{-\Delta u^2}{2\sigma_1^2} + \frac{-\Delta u^2}{2\sigma_2^2}\right) \right] d\Delta u \right)^2 \\ &= n_{\text{nodes}} \times f_{\text{norm},1}^2 \times \left( Qc_1^2\sigma_1\sqrt{\pi} + (1-Q)c_2^2\sigma_2\sqrt{\pi} + c_1c_2\sigma_s\sqrt{2\pi} \right)^2 \quad (44) \end{aligned}$$

with  $\sigma_s = \frac{1}{\sqrt{\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}}}$ .

So, after inserting the values for  $c_1$ ,  $c_2$ , and  $\sigma_s$  this becomes:

$$r_1 = \frac{n_{\text{nodes}}}{4\pi n_{\text{sol}}^2 [Q + (1-Q)(n_{\text{sol}} - 1)]^2} \times \left( \frac{Q}{\sigma_1} + \frac{(1-Q)(n_{\text{sol}} - 1)^2}{\sigma_2} + \frac{\sqrt{2}(n_{\text{sol}} - 1)}{\sqrt{\sigma_1^2 + \sigma_2^2}} \right)^2 \quad (45)$$

Similarly,  $r_r$  follows from:

$$\begin{aligned}
r_r &= \int_{\Delta u=-\infty}^{\infty} \int_{\Delta v=-\infty}^{\infty} P_{p,2,3,4}(\Delta u, \Delta v) \times P_{o,2}(\Delta u, \Delta v) d\Delta v d\Delta u \\
&= \mathbf{n}_{\text{nodes}} \times (\mathbf{n}_{\text{sol}} - 1) \times f_{\text{norm},2}^2 \times \\
&\left( \int_{\Delta u=-\infty}^{\infty} [(1-Q)P_{o,1}(\Delta u) + (\mathbf{n}_{\text{sol}} - 1 + Q - 1)P_{o,2}(\Delta u)] \times \right. \\
&\quad \left. [P_{o,1}(\Delta u) + P_{o,2}(\Delta u)] d\Delta u \right)^2 \\
&= \mathbf{n}_{\text{nodes}} \times (\mathbf{n}_{\text{sol}} - 1) \times f_{\text{norm},2}^2 \times \\
&\left( \int_{\Delta u=-\infty}^{\infty} [(1-Q)P_{o,1}^2(\Delta u) + (\mathbf{n}_{\text{sol}} - 1 + Q - 1)P_{o,2}^2(\Delta u) + \right. \\
&\quad \left. (\mathbf{n}_{\text{sol}} - 1)P_{o,1}(\Delta u)P_{o,2}(\Delta u)] d\Delta u \right)^2 \\
&= \mathbf{n}_{\text{nodes}} \times (\mathbf{n}_{\text{sol}} - 1) \times f_{\text{norm},2}^2 \times \\
&\left( \int_{\Delta u=-\infty}^{\infty} \left[ (1-Q)c_1^2 \exp\left(\frac{-\Delta u^2}{\sigma_1^2}\right) + (\mathbf{n}_{\text{sol}} - 1 + Q - 1)c_2^2 \exp\left(\frac{-\Delta u^2}{\sigma_2^2}\right) + \right. \right. \\
&\quad \left. \left. (\mathbf{n}_{\text{sol}} - 1)c_1c_2 \exp\left(\frac{-\Delta u^2}{2\sigma_1^2} + \frac{-\Delta u^2}{2\sigma_2^2}\right) \right] d\Delta u \right)^2 \\
&= \mathbf{n}_{\text{nodes}} \times (\mathbf{n}_{\text{sol}} - 1) \times \frac{\mathbf{n}_{\text{sol}}^2}{[(1-Q) + (\mathbf{n}_{\text{sol}} - 1 + Q - 1)(\mathbf{n}_{\text{sol}} - 1)]^2} \times \\
&\quad \left( (1-Q)c_1^2\sigma_1\sqrt{\pi} + (\mathbf{n}_{\text{sol}} - 1 + Q - 1)c_2^2\sigma_2\sqrt{\pi} + (\mathbf{n}_{\text{sol}} - 1)c_1c_2\sigma_s\sqrt{2\pi} \right)^2 \tag{46}
\end{aligned}$$

So, after inserting the values for  $c_1$ ,  $c_2$ , and  $\sigma_s$  this becomes:

$$\begin{aligned}
r_r &= \frac{\mathbf{n}_{\text{nodes}}(\mathbf{n}_{\text{sol}} - 1)}{4\pi\mathbf{n}_{\text{sol}}^2} \times \frac{1}{[(1-Q) + (\mathbf{n}_{\text{sol}} - 1 + Q - 1)(\mathbf{n}_{\text{sol}} - 1)]^2} \times \\
&\quad \left( \frac{(1-Q)}{\sigma_1} + \frac{(\mathbf{n}_{\text{sol}} - 1 + Q - 1)(\mathbf{n}_{\text{sol}} - 1)^2}{\sigma_2} + \frac{(\mathbf{n}_{\text{sol}} - 1)^2\sqrt{2}}{\sqrt{\sigma_1^2 + \sigma_2^2}} \right)^2 \tag{47}
\end{aligned}$$

When equations 45 and 47 are used, with  $Q = 0.5$  and  $\mathbf{n}_{\text{sol}}=3$  this gives:

$$r_r = \frac{\mathbf{n}_{\text{nodes}}}{4\pi \times 9 \times \left(\frac{3}{2}\right)^2} \times \left( \frac{1}{\sigma_1} + \frac{1}{2} \times \frac{2^2}{\sigma_2} + \frac{2\sqrt{2}}{\sqrt{\sigma_1^2 + \sigma_2^2}} \right)^2 \tag{48}$$

and:

$$r_r = \frac{\mathbf{n}_{\text{nodes}} \times 2}{4\pi \times 9 \times \left(\frac{7}{2}\right)^2} \times \left( \frac{1}{\sigma_1} + \frac{3}{2} \times \frac{2^2}{\sigma_2} + \frac{2^2\sqrt{2}}{\sqrt{\sigma_1^2 + \sigma_2^2}} \right)^2 \tag{49}$$

Now we fill in the values for  $\sigma_1$  and  $\sigma_2$  and get  $\text{score}_r = 1.150$ . For  $Q=1$  this would be:  $\text{score}_r = 0.532$ . Combining the scores for the 3 solution case, the FoM becomes  $0.8 \times 0.814 + 0.2 \times 1.150 = 0.881$  for  $Q=0.5$ , and  $\text{FoM} = 0.758$  for  $Q=1$ .

And in case of 4 solutions, using eq. 40 with  $\mathbf{n}_{\text{sol}}=4$ , the  $\text{RMS}_u$  becomes:

$$\text{RMS}_u^2 = \frac{\sigma_2\sigma_{a,1}^3 + 3\sigma_1\sigma_{a,2}^3}{\sigma_2\sigma_{a,1} + 3\sigma_1\sigma_{a,2}} = \frac{9.281 + 3 \times 4.793}{8.250 + 3 \times 2.209} = 1.590 \tag{50}$$

So  $\text{score}_u = 0.841$ .

The ranking score again follows from eqs. 45 and 47, using  $n_{\text{sol}}=4$  and  $Q=0.5$ :

$$r_1 = \frac{n_{\text{nodes}}}{4\pi \times 16 \times 2^2} \times \left( \frac{1}{2} + \frac{1}{2} \times 3^2 + \frac{3\sqrt{2}}{\sqrt{\sigma_1^2 + \sigma_2^2}} \right)^2 \quad (51)$$

and:

$$r_r = \frac{n_{\text{nodes}} \times 3}{4\pi \times 16 \times 8^2} \times \left( \frac{1}{2} + \frac{5}{2} \times 3^2 + \frac{3^2\sqrt{2}}{\sqrt{\sigma_1^2 + \sigma_2^2}} \right)^2 \quad (52)$$

From this we obtain  $\text{score}_r = 1.353$  for  $Q=0.5$ . Similarly we obtain  $\text{score}_r = 0.796$  for  $Q=1$ . Combining the scores for 4 solutions, the FoM becomes  $0.8 \times 0.841 + 0.2 \times 1.353 = 0.943$  for  $Q=0.5$ , and  $\text{FoM} = 0.832$  for  $Q=1$ .

#### 4.1.5 Synthetic solutions

The implementation of the FoM calculation was tested by generating artificial wind vector solutions, and using them as input for the FoM routine. The results are summarised in table 1.

- First, for a case with 1 solution with a SD of 1.5 m/s around the true wind, 5001 wind vectors were generated. These have been used as input for the FoM calculation (which is given in Appendix 6). This yielded:  $\text{score}_u = 0.716$ ,  $\text{score}_v = 0.697$ , and  $\text{score}_r = 0.0$ , so  $\text{FoM} = 0.565$ . This is very close to the expected values of:  $\text{score}_u = 0.707$ ,  $\text{score}_v = 0.707$ ,  $\text{score}_r = 0.0$ , and  $\text{FoM} = 0.566$ . To test how these numbers vary for different values of the SD of the wind solutions a scan was done for different SD values. The results are given in Fig. 1.
- Then, a second solution was added with 180 degree ambiguity, and both solutions were in 50 % of all cases swapped ( $Q=0.5$ ). This yielded:  $\text{score}_u = 0.759$ ,  $\text{score}_v = 0.759$ , and  $\text{score}_r = 1.002$ , so  $\text{FoM} = 0.808$ . This is very close to the expected values of:  $\text{score}_u = 0.759$ ,  $\text{score}_v = 0.759$ ,  $\text{score}_r = 1.0$ , and  $\text{FoM} = 0.807$ . For  $Q=1$  the  $\text{score}_r = 0.141$ , and  $\text{FoM} = 0.635$ . This again is very close to the expected values of:  $\text{score}_r = 0.139$ , and  $\text{FoM} = 0.635$ . Also for this case, with 2 solutions and 180 degree ambiguity, a scan of the SD was done. In the calculations for this scan the swapping was switched off ( $Q=1$ ), because when in 50 % of all cases the 1st and 2nd solution are swapped ( $Q=0.5$ ), this always results in a  $\text{score}_r=1$  independant of SD, which is not very informative). The results are given in Fig. 2. The found values for  $\text{score}_u$  and  $\text{score}_v$  are very similar to the values found in case of 1 solution. (except for a SD below 1 m/s, which is not very important for the current application of the FoM). The values for  $\text{score}_r$  increase more gradually with SD. This is not very important for the ERS case, since the equal probability of both solutions causes 50 % of the 1st rank solutions to point the wrong way. Therefore  $\text{score}_r$  is very close to 1 in this case. For seawinds and RFscat this will be different.

Also a scan was done over the angle between the 1st and 2nd rank solution for the case with 2 solutions (again for  $Q=1$ ). The results are given in Fig. 3. The variation

Table 1: Values for the FoM for implementation 1.

$n_{\text{sol}}$	$\text{score}_u$	$\text{score}_v$	$\text{score}_r$ ( $Q = \frac{1}{2}$ )	$\text{score}_r$ ( $Q = 1$ )	FoM ( $Q = \frac{1}{2}$ )	FoM ( $Q = 1$ )
Exact values						
1	0.707	0.707	0.0	0.0	0.566	0.566
2 (180 deg.amb.)	0.759	0.759	1.0	0.139	0.807	0.635
2 (uncorrelated)	0.773	0.773	1.0	0.233	0.818	0.665
3 (uncorrelated)	0.814	0.814	1.150	0.532	0.881	0.758
4 (uncorrelated)	0.841	0.841	1.353	0.796	0.943	0.832
Values resulting from synthetic windvectors						
1	0.716	0.697	0.0	0.0	0.565	0.565
2 (180 deg.amb.)	0.759	0.759	1.002	0.141	0.808	0.635
2 (random angle)	0.772	0.770	0.985	0.370	0.814	0.691
2 (uncorrelated)	0.773	0.778	0.981	0.226	0.816	0.665
$Q = \frac{1}{2}$						
3 (125 deg.amb.)	0.805	0.799	1.003		0.842	
3 (random angle)	0.805	0.804	1.080		0.859	
3 (uncorrelated)	0.809	0.816	1.061		0.862	
4 (random angle)	0.826	0.822	1.159		0.891	
4 (uncorrelated)	0.838	0.842	1.175		0.907	
$Q = 1$						
3 (125 deg.amb.)	0.807	0.798		0.407		0.723
3 (random angle)	0.810	0.801		0.555		0.755
3 (uncorrelated)	0.815	0.817		0.533		0.759
4 (random angle)	0.827	0.823		0.734		0.807
4 (uncorrelated)	0.836	0.848		0.793		0.832
Real data						
2 (real ERS data)	0.761	0.742		0.768		0.755
var (real Seawinds data)	0.725	0.701		0.416		0.653

in the FoM curve is very small which is a consequence of the definition of the FoM. Not much variation is to be expected, except when the second solution is close enough to the first one, so that (due to the noise added to the solutions) the ranking changes. There is very little variation between angle=90 and angle=180, which is clearly not desired, since an ambiguity of 180 degrees is much preferred above an ambiguity of 90 degrees (for low and moderate wind speeds). The  $\text{score}_r$  curve has a sharp increase between angles of 10 and 40 degrees. This is a desired property since it will give an extra penalty to solutions with angles close to the first rank solution.

- When this second solution is given a random angle compared to the first solution, and  $Q=0.5$ , this yields:  $\text{score}_u=0.772$ ,  $\text{score}_v=0.770$ , and  $\text{score}_r=0.985$ , so  $\text{FoM}=0.814$ . The  $\text{score}_r$  changes to 0.370 for  $Q=1$ , and FoM to 0.691.

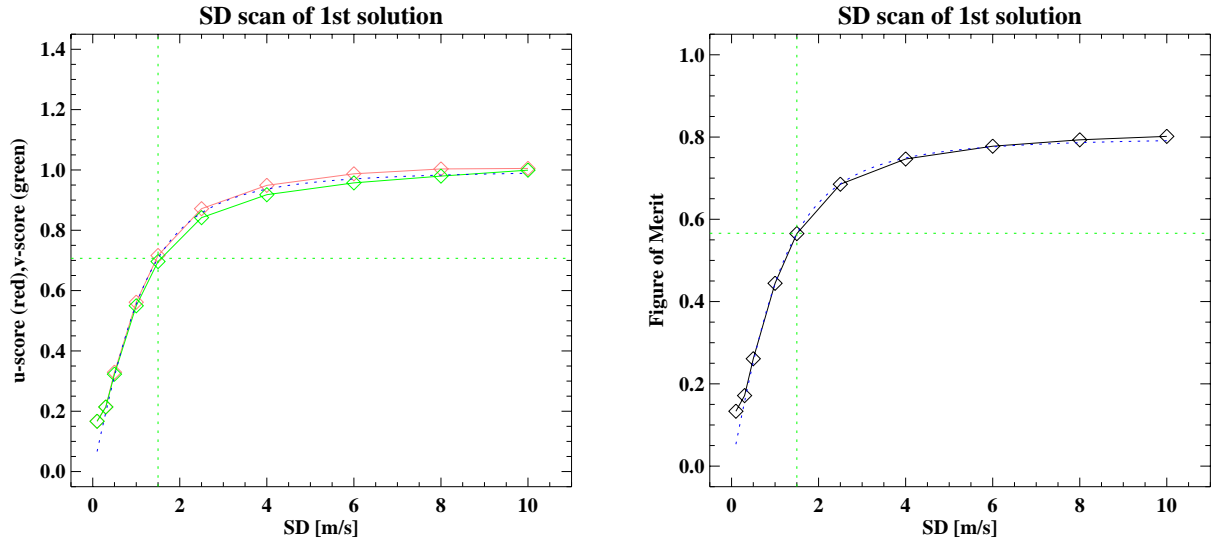


Figure 1: *FoM* results for different *SD* values for the synthetic wind solution (case with 1 solution). The left plot shows the  $\text{score}_u$  and  $\text{score}_v$ . The right plot shows the *FoM*. The blue dotted line in both plots is the exact curve using equation 17. The green crossing lines indicate the result with  $SD=1.5$  m/s.

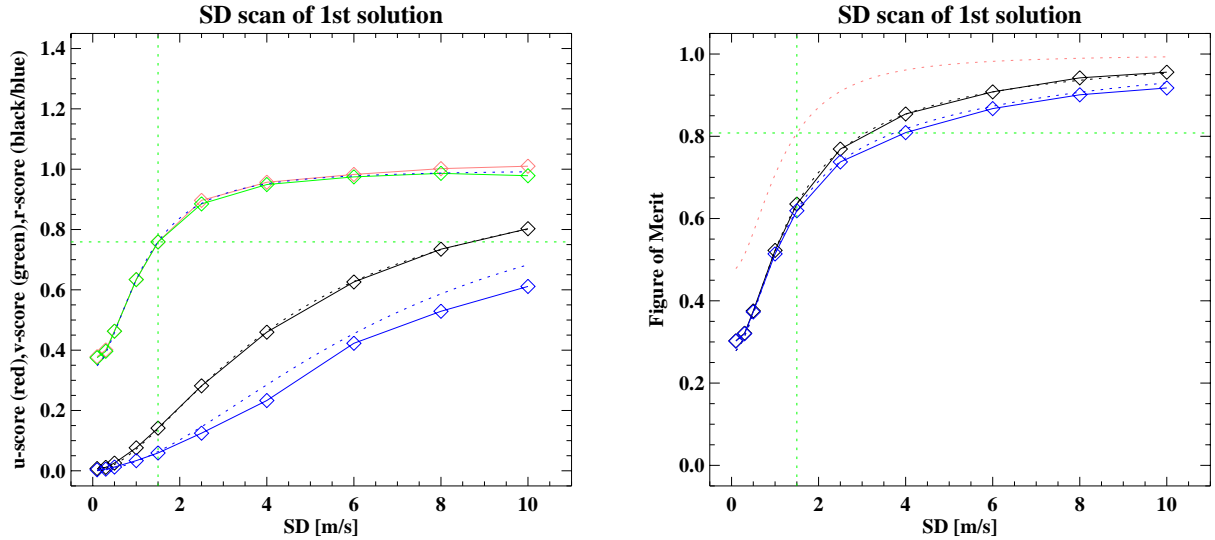


Figure 2: *FoM* results for different *SD* values for the synthetic wind solution (case with 2 solutions with 180 degree ambiguity) for implementation 1. The left plot shows the  $\text{score}_u$  (red),  $\text{score}_v$  (green) and  $\text{score}_r$  (black). The right plot shows the *FoM*. The dotted lines in the left plot are the exact calculation of the scores using eqs 26, 30 and 32. The red dotted line in the right plot is the exact *FoM* calculation for  $Q=0.5$ , and the black dotted line is the exact *FoM* calculation for  $Q=1$ . The green crossing lines indicate the result with  $SD=1.5$  m/s and  $Q=0.5$ . The blue solid and dotted lines are the results for implementation 3, which will be discussed in section 4.3.5.

- When this second solution taken completely uncorrelated to the first solution, and  $Q=0.5$ , this yields:  $\text{score}_u=0.773$ ,  $\text{score}_v=0.778$ , and  $\text{score}_r=0.981$ , so  $\text{FoM}=0.816$ .

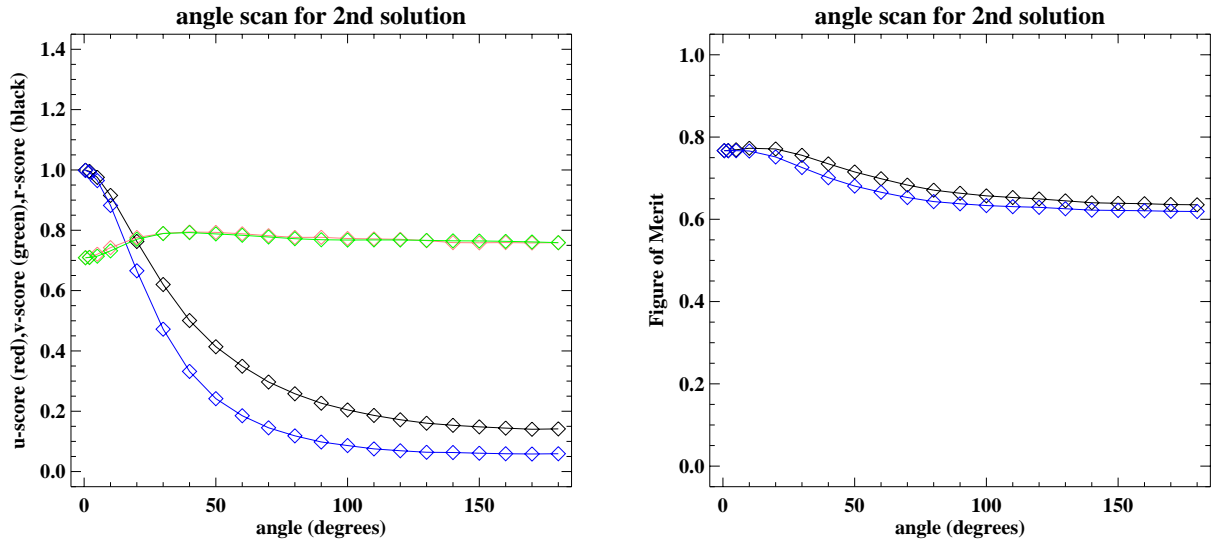


Figure 3: *FoM* results for different angles between the 1st and 2nd synthetic wind solution (case with 2 solutions and  $Q=1$ ). The left plot shows the  $\text{score}_u$  (red),  $\text{score}_v$  (green) and  $\text{score}_r$  (black). The right plot shows the *FoM*. The blue solid and dotted lines are the results for implementation 3, which will be discussed in section 4.3.5.

This again is very close to the expected values of:  $\text{score}_u = 0.773$ ,  $\text{score}_v = 0.773$ ,  $\text{score}_r = 1.0$ , and  $\text{FoM} = 0.818$ . For  $Q=1$   $\text{score}_r = 0.226$  and  $\text{FoM} = 0.665$ , which again matches the expected theoretical values of  $\text{score}_r = 0.233$  and  $\text{FoM} = 0.665$  very well.

- For the case with 3 solutions, with an ambiguity of  $\pm 125$  degrees between the 1st and both other solutions, and  $Q=0.5$ , the numbers are:  $\text{score}_u = 0.805$ ,  $\text{score}_v = 0.799$ , and  $\text{score}_r = 1.003$ , so  $\text{FoM} = 0.842$ . For  $Q=1$  the scores are:  $\text{score}_u = 0.807$ ,  $\text{score}_v = 0.798$ , and  $\text{score}_r = 0.407$ , so  $\text{FoM} = 0.723$ .
- For the case with 3 solutions (with random angle between the 1st rank solution and both other solutions), and  $Q=0.5$ , the numbers are:  $\text{score}_u = 0.805$ ,  $\text{score}_v = 0.804$ , and  $\text{score}_r = 1.080$ , so  $\text{FoM} = 0.859$ . For  $Q=1$  the scores are:  $\text{score}_u = 0.810$ ,  $\text{score}_v = 0.801$ , and  $\text{score}_r = 0.555$ , so  $\text{FoM} = 0.755$ .
- For the case with 3 solutions taken completely uncorrelated to each other, and  $Q=0.5$ , the numbers are:  $\text{score}_u = 0.809$ ,  $\text{score}_v = 0.816$ , and  $\text{score}_r = 1.061$ , so  $\text{FoM} = 0.862$ . The  $\text{score}_u$  and  $\text{score}_v$  are very close to the expected values of  $\text{score}_u = 0.814$ ,  $\text{score}_v = 0.814$ . However the  $\text{score}_r$  and the *FoM* deviates from the expected values of  $\text{score}_r = 1.150$ , and  $\text{FoM} = 0.881$ . For  $Q=1$  the scores are:  $\text{score}_u = 0.815$ ,  $\text{score}_v = 0.817$ , and  $\text{score}_r = 0.533$ , so  $\text{FoM} = 0.759$ . Now all scores and the *FoM* are again very close to the expected theoretical values of:  $\text{score}_u = 0.814$ ,  $\text{score}_v = 0.814$ ,  $\text{score}_r = 0.532$  and  $\text{FoM} = 0.758$ .
- For the case with 4 solutions (with random angle between the 1st rank solution and the other solutions), and  $Q=0.5$ , the numbers are:  $\text{score}_u = 0.826$ ,  $\text{score}_v = 0.822$ , and  $\text{score}_r = 1.159$ , so  $\text{FoM} = 0.891$ . For  $Q=1$  the scores are:  $\text{score}_u = 0.827$ ,  $\text{score}_v = 0.823$ , and  $\text{score}_r = 0.734$ , so  $\text{FoM} = 0.807$ .

- For the case with 4 solutions taken completely uncorrelated to each other, and  $Q=0.5$ , the numbers are:  $\text{score}_u = 0.838$ ,  $\text{score}_v = 0.842$ , and  $\text{score}_r = 1.175$ , so  $\text{FoM} = 0.907$ . The  $\text{score}_u$  and  $\text{score}_v$  are very close to the expected values of  $\text{score}_u = 0.841$ ,  $\text{score}_v = 0.841$ . However the  $\text{score}_r$  and the FoM deviate significantly from the expected values of  $\text{score}_r = 1.353$ , and  $\text{FoM} = 0.943$ . For  $Q=1$  the scores are:  $\text{score}_u = 0.836$ ,  $\text{score}_v = 0.848$ , and  $\text{score}_r = 0.793$ , so  $\text{FoM} = 0.832$ . These scores again are very close to the expected values for  $Q=1$  of  $\text{score}_u = 0.841$ ,  $\text{score}_v = 0.841$ ,  $\text{score}_r = 0.796$  and  $\text{FoM} = 0.832$ .

#### 4.1.6 Comparison with ERS data

The FoM for the ERS geometry was also tested by applying it to real ERS data. To do this, data was taken from 6 to 17 February 1996. Only the data contained in the inner area of the Hirlam model was selected (to avoid the edge regions, which are known to have a poor background). The node-averaged mean results are:  $\text{score}_u = 0.761$ ,  $\text{score}_v = 0.742$ ,  $\text{score}_r = 0.768$  and  $\text{FoM} = 0.755$ . The  $\text{score}_u$  and  $\text{score}_v$  are close to the expected theoretical value of  $\text{score}_u = 0.759$ . The  $\text{score}_r$  is clearly better than the expected value of 1.0. This can be explained by the fact that 62 % of all first rank solutions also are the closest solution (and not 50 % as was assumed in the theoretical calculation). If this is entered into the synthetic solution ERS-case then a  $\text{score}_r = 0.789$  is found, which is again close to the found value.

#### 4.1.7 Comparison with Seawinds data

The FoM for the Seawinds geometry was also tested by applying it to real Seawinds data. To do this, global data was taken from 2 to 4 February 2002. As reference wind an ECMWF forecast was used. The node-averaged mean results are:  $\text{score}_u = 0.725$ ,  $\text{score}_v = 0.701$ ,  $\text{score}_r = 0.416$  and  $\text{FoM} = 0.653$ . The  $\text{score}_u$  and  $\text{score}_v$  are significantly below the results from the synthetic calculation, for both 3- and 4 solutions. This can be explained by the fact that the used ECMWF background windfield is of better quality, then the assumed  $\sigma_b = 1.5$  m/s.

For example if a case with 3 uncorrelated solutions is considered, and reasonable values of  $\sigma_b = 0.92$  m/s and  $Q = 0.93$  are used, the resulting scores from the synthetic vector calculation are:  $\text{score}_u = 0.699$ ,  $\text{score}_v = 0.704$ ,  $\text{score}_r = 0.424$ , and a  $\text{FoM} = 0.646$ , which is already close to the observed values. This is very close to published values for the error in the ECMWF reference wind, of 0.90 m/s for the u-component and 0.95 m/s for the v-component [9].

The difference between  $\text{score}_u$  and  $\text{score}_v$  can probably be explained by the distortion of the wind component distribution by the tradewinds.

The Seawinds data consists mostly of a combination of nodes with 2, 3 and 4 solutions, so the combined numbers will be different. The idea remains the same, however, that by taking similar values of  $\sigma_b$  and  $Q$  the results can be explained.

## 4.2 Implementation 2

Giving different weights to the different solutions, based on the angle-sector associated with each solution, only makes a difference for 3 or 4 solutions. These weights also complicate the



Table 2: Values for the FoM for implementation 2.

$n_{\text{sol}}$	$\text{score}_u$	$\text{score}_v$	$\text{score}_r$ ( $Q = \frac{1}{2}$ )	$\text{score}_r$ ( $Q = 1$ )	FoM ( $Q = \frac{1}{2}$ )	FoM ( $Q = 1$ )
$Q = \frac{1}{2}$						
3 (125 deg.amb.)	0.802	0.798	1.000		0.840	
3 (random angle)	0.810	0.805	1.080		0.862	
3 (uncorrelated)	0.810	0.814	1.063		0.862	
4 (random angle)	0.802	0.795	1.128		0.865	
4 (uncorrelated)	0.839	0.840	1.182		0.908	
$Q = 1$						
3 (125 deg.amb.)	0.804	0.792		0.369		0.712
3 (random angle)	0.808	0.810		0.554		0.758
3 (uncorrelated)	0.816	0.819		0.537		0.761
4 (random angle)	0.803	0.798		0.661		0.773
4 (uncorrelated)	0.835	0.847		0.809		0.835
Real data						
var (real Seawinds data)	0.728	0.703	0.422		0.657	

exact calculation of the FoM so this has not been attempted for this implementation. The check with synthetic solutions has been done, as is described below, and also a comparison with real Seawinds data is performed.

#### 4.2.1 Synthetic solutions

Also this second implementation of the FoM calculation was tested by generating artificial wind vector solutions, and using them as input for the FoM routine. 5001 windvectors were generated in each example. The results are summarised in table 2.

- First a case with 3 solutions is simulated. The first solution has a SD of 1.5 m/s around the true wind. Then the second and third solution are constructed by adding and subtracting 125 degrees to the first solution. (from the real Seawinds data, it was found that for the cases with 3 solutions, the average angle difference between the first, and the second or third solution, is 125 degrees). It was not attempted to simulate the ranking of the solutions ( $Q=1$  was taken). This should not influence the results very much, since the closest solution for real Seawinds data is equal to the first rank solution in 90-95 % of all cases (except for the outer and nadir nodes). When these synthetic solutions are used as input for the FoM calculation this yields:  $\text{score}_u=0.804$ ,  $\text{score}_v=0.792$ , and  $\text{score}_r=0.369$ , so  $\text{FoM} = 0.712$ . These results are close to the synthetic results for implementation 1, except for  $\text{score}_r$  which is somewhat lower.
- For the case with 3 solutions (with random angle between the 1st rank solution and both other solutions) and  $Q=1$  the numbers are:  $\text{score}_u=0.808$ ,  $\text{score}_v=0.810$ , and  $\text{score}_r=0.554$ , so  $\text{FoM} = 0.758$ . These results are all very close to the synthetic results for implementation 1.

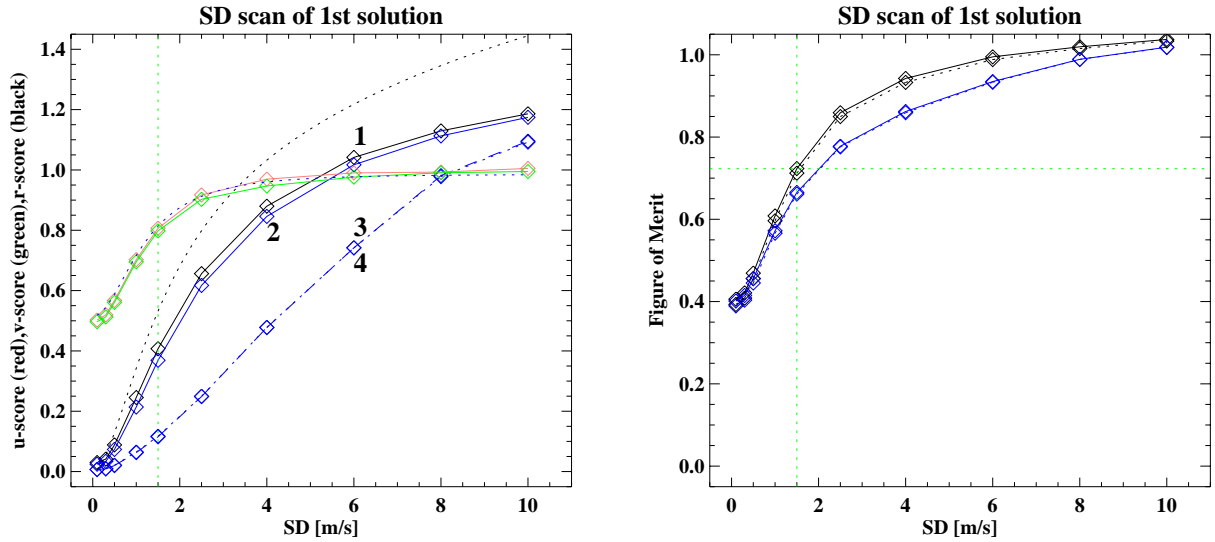


Figure 4: FoM results for different SD values for the synthetic wind solution (case with 3 solutions with 125 degrees between the first rank and the other solutions, and  $Q=1$ ). The left plot shows the score<sub>u</sub> (red), score<sub>v</sub> (green) and score<sub>r</sub> (black) for implementation 1. The blue lines are the curves for score<sub>r</sub> for implementations 2, 3 and 4. The black dotted lines in the left plot are the exact calculations of the scores (for FoM implementation 1) for the case of 3 uncorrelated solutions and  $Q=1$ , using eq. 41, 48 and 49. The fact that only for 3 uncorrelated solutions an exact curve was determined is the reason why the exact score<sub>r</sub> curve is above the synthetic curve (with 125 degree angles between the first rank and other solutions). The synthetic results for score<sub>u</sub> and score<sub>v</sub> for implementations 2, 3 and 4 coincide almost precisely with the results for implementation 1, so they have not been drawn. The right plot shows the FoM for implementation 1 (black solid line) and implementation 2 (black dotted line). The green crossing lines indicate the result with  $SD=1.5$  m/s and  $Q=1$ . The blue lines give the results for score<sub>r</sub> for implementation 3 and 4 and will be further discussed in sections 4.3 and 4.4. Note that for an angle of 120 degrees the scores 1 and 2, and the scores 3 and 4 should be identical by definition.

- For the case with 3 solutions taken completely uncorrelated to each other, and  $Q=1$ , the numbers are: score<sub>u</sub> = 0.816, score<sub>v</sub> = 0.819, and score<sub>r</sub> = 0.537, so FoM = 0.761. Also these results are all very close to the synthetic results for implementation 1.
- For the case of 3 solutions, a scan of the SD was done. In the calculations for this scan the swapping was switched off ( $Q=1$ ) and an angle between the first rank and both other solutions of 125 degrees was taken. The results are given in Fig. 4. The found values for score<sub>u</sub> and score<sub>v</sub> are very similar to the theoretical values for uncorrelated solutions. The found values for score<sub>r</sub> differ clearly from the theoretical result. This is caused by the fact that the synthetic result uses an angle difference of 125 degrees, which is in the angle region where score<sub>r</sub> is very low (compare with Fig. 3). For uncorrelated windsolutions also many cases with small angle differences will occur (for which score<sub>r</sub> is much larger), so this explains the higher theoretical curve for uncorrelated solutions.

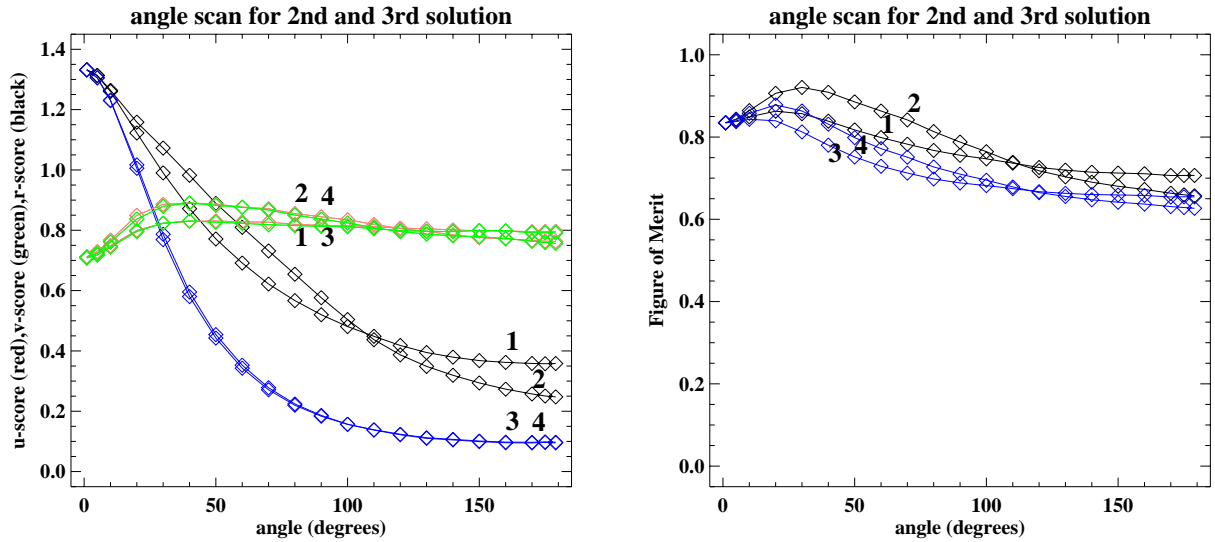


Figure 5: *FoM* results for different angles between the 1st rank solution and the other 2 solutions (case with 3 synthetic wind solutions). The 4 implementations are indicated by the numbers 1-4. The green and red lines are the curves for  $\text{score}_u$  and  $\text{score}_v$ . The black and blue lines are the curves for  $\text{score}_r$ . Note that for an angle of 120 degrees all 3 solutions will have identical weights, so the curves for implementation 1 and 2 should cross each other at that angle, as should the curves of implementation 3 and 4. The fact that this does not happen here pointed us to another bug in the implementation of the *FoM* implementation. However, the effect of this bug is very small (see section 4.2.3).

- Then for this 3 solution case, the angle between the 1st rank and the other 2 solutions was scanned. The results are given in Fig. 5. The found values for  $\text{score}_u$  and  $\text{score}_v$  vary more than in the 2 solution case, but still not very much. The largest effect comes from the  $\text{score}_r$ , despite of its small weight of only 0.2.
- For the case with 4 solutions (with random angle between the 1st rank solution and the other solutions) and  $Q=1$  the numbers are:  $\text{score}_u=0.803$ ,  $\text{score}_v=0.798$ , and  $\text{score}_r=0.661$ , so  $\text{FoM}=0.773$ .
- For the case with 4 solutions taken completely uncorrelated to each other, and  $Q=1$ , the numbers are:  $\text{score}_u=0.835$ ,  $\text{score}_v=0.847$ , and  $\text{score}_r=0.809$ , so  $\text{FoM}=0.835$ .
- Finally a case with 4 solutions was simulated. The angle between the 1st rank and the 2nd and 3rd solution was scanned. The angle between the 1st rank and the 4th solution was taken to be 180 degrees, and  $Q=1$  was taken. The results are given in Fig. 6. The results are very similar compared to the 3 solution case. For an angle of 125 degrees the values are:  $\text{score}_u=0.804$ ,  $\text{score}_v=0.790$ , and  $\text{score}_r=0.530$ , so  $\text{FoM}=0.744$ . This is very close to the values for 3 solutions, except for  $\text{score}_r$ .

#### 4.2.2 Comparison with Seawinds data

The *FoM* for the Seawinds geometry was also tested by applying it to real Seawinds data. For this the same data is used as was described in section 4.1.7. The node-averaged mean

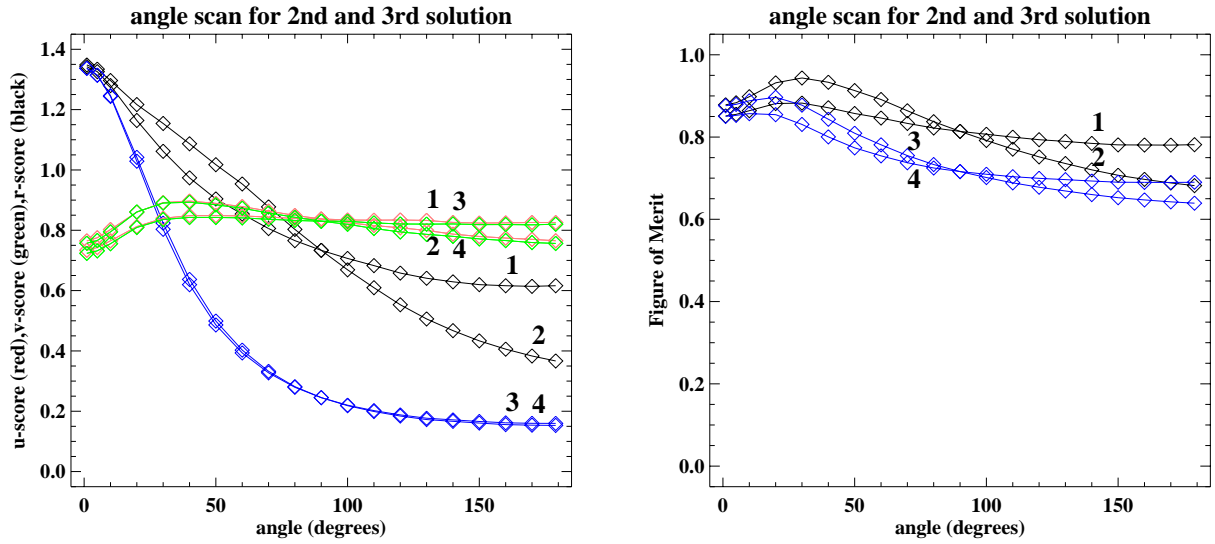


Figure 6: *FoM* results for different angles between the 1st rank solution and the 2nd and 3rd solution. (case with 4 synthetic wind solutions). The ambiguity of the 4th solution is fixed to 180 degrees.

Note that in this case the score<sub>r</sub> curves do cross at 90 degrees, as expected.

results are: score<sub>u</sub> = 0.728, score<sub>v</sub> = 0.703, score<sub>r</sub> = 0.422, FoM = 0.657.

These scores are very close to the ones found for implementation 1. From figures 4, 5 and 6 it can be seen that the score<sub>u</sub> and score<sub>v</sub> are very similar for implementation 1 and 2.

However, the curve for score<sub>r</sub> differs clearly in figure 5. For angle differences between 20 and 110 degrees implementation 2 gives a larger penalty. For angle differences between 110 and 180 degrees implementation 1 gives a larger penalty. An ideal score would give a large penalty around 0 degrees, and this behaviour is present in both implementations.

### 4.2.3 Another small bug

The 3 solution case for angles between the 3 solutions of 120 degrees should give equal weights for the 3 solutions for FoM implementations 2 and 4. Therefore the FoM result should be identical to the FoM result of implementation 1 or 3. The fact that this is not the case in the results shown in Fig. 5 pointed us to yet another bug in the FoM code.

The main problem was that the routine that calculates the weights assumed that the solutions were sorted in counter-clock-wise direction, which was not always the case.

The code in Appendix A has been corrected for this bug, and the correction can be switched on by setting the switch “apply\_correction\_to\_weights = true”. All synthetic results have been recalculated with this bug fix, which showed that the differences are very small.

- For 1 and 2 solutions there are no differences.
- For 3 solutions the only noticeable change is that the score<sub>r</sub> curve for implementation 2 in Fig. 5 shifts a little up, causing the curves for implementations 1 and 2 to intersect now correctly at an angle of 120 degrees. A second effect is that the score<sub>r</sub> curves become somewhat noisier, which indicates that the buggy version of the FoM caused some extra smoothing.

- For 4 solutions no shift in  $\text{score}_r$  is visible anymore and the only effect is the increased noisiness of the  $\text{score}_r$  curves in Fig. 5.

The differences in the SD scan shown in Fig. 4 are small and are within the variation caused by the statistical noise on the curves.

### 4.3 Implementation 3

#### 4.3.1 1 solution

For the 1 solution case, the  $\text{score}_r$  is always zero, so there is no difference with implementation 1.

#### 4.3.2 2 solutions, 180 degrees apart, exact calculation

The new ranking score for this case is calculated by replacing  $P_o$  by  $P_a$  in eqs. 9 and 10. So:

$$P_{a,2}(\Delta u, \Delta v) = P_a(\Delta u)P_a(\Delta v) = P_o(\Delta u) \times P_b(\Delta u) \times P_o(\Delta v) \times P_b(\Delta v) \quad (53)$$

The probing functions  $P_{p,1}$  and  $P_{p,2,3,4}$  remain the same.

Using this the  $r_1$  score becomes:

$$\begin{aligned} r_1 &= \int_{\Delta u=-\infty}^{\infty} \int_{\Delta v=-\infty}^{\infty} P_{p,1}(\Delta u, \Delta v) \times P_{a,2}(\Delta u, \Delta v) d\Delta v d\Delta u \\ &= \mathbf{n}_{\text{nodes}} \times f_{\text{norm},1}^2 \times \\ &\left( \int_{\Delta u=-\infty}^{\infty} [QP_{o,1}(\Delta u) + (1-Q)P_{o,2}(\Delta u)] \times [P_{o,1}(\Delta u) + P_{o,2}(\Delta u)] \times P_b(\Delta u) d\Delta u \right)^2 \\ &= \mathbf{n}_{\text{nodes}} \times f_{\text{norm},1}^2 \times \\ &\left( \int_{\Delta u=-\infty}^{\infty} [QP_{o,1}^2(\Delta u) + (1-Q)P_{o,2}^2(\Delta u) + P_{o,1}(\Delta u)P_{o,2}(\Delta u)] \times P_b(\Delta u) d\Delta u \right)^2 \\ &= \mathbf{n}_{\text{nodes}} \times f_{\text{norm},1}^2 \times \left( \int_{\Delta u=-\infty}^{\infty} \left[ Qc_1^2 \exp\left(\frac{-\Delta u^2}{\sigma_1^2} + \frac{-\Delta u^2}{2\sigma_b^2}\right) + \right. \right. \\ &\quad \left. \left. (1-Q)c_2^2 \exp\left(\frac{-\Delta u^2}{\sigma_2^2} + \frac{-\Delta u^2}{2\sigma_b^2}\right) + \right. \right. \\ &\quad \left. \left. c_1c_2 \exp\left(\frac{-\Delta u^2}{2\sigma_1^2} + \frac{-\Delta u^2}{2\sigma_2^2} + \frac{-\Delta u^2}{2\sigma_b^2}\right) \right] d\Delta u \right)^2 \\ &= \mathbf{n}_{\text{nodes}} \times f_{\text{norm},1}^2 \times \left( Qc_1^2\sigma_{1,b}\sqrt{\pi} + (1-Q)c_2^2\sigma_{2,b}\sqrt{\pi} + c_1c_2\sigma_{1,2,b}\sqrt{2\pi} \right)^2 \quad (54) \end{aligned}$$

with  $\sigma_{1,b} = \frac{1}{\sqrt{\frac{1}{\sigma_1^2} + \frac{1}{2\sigma_b^2}}}$ , and  $\sigma_{2,b} = \frac{1}{\sqrt{\frac{1}{\sigma_2^2} + \frac{1}{2\sigma_b^2}}}$ , and  $\sigma_{1,2,b} = \frac{1}{\sqrt{\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} + \frac{1}{\sigma_b^2}}}$ .

So, after inserting the values for  $c_1$ ,  $c_2$  and  $f_{\text{norm},1}$  this becomes:

$$\begin{aligned} r_1 &= \frac{\mathbf{n}_{\text{nodes}}}{4\pi\mathbf{n}_{\text{sol}}^2 [Q + (1-Q)(\mathbf{n}_{\text{sol}} - 1)]^2} \times \\ &\left( \frac{Q\sigma_{1,b}}{\sigma_1^2} + \frac{(1-Q)(\mathbf{n}_{\text{sol}} - 1)^2\sigma_{2,b}}{\sigma_2^2} + \frac{\sqrt{2}(\mathbf{n}_{\text{sol}} - 1)\sigma_{1,2,b}}{\sigma_1\sigma_2} \right)^2 \quad (55) \end{aligned}$$

(the normalisation constant for  $P_b$  was left out, since it is of no importance, and will be deleted again in the division that leads to the score<sub>r</sub>)

Similarly,  $r_r$  follows from:

$$\begin{aligned}
r_r &= \int_{\Delta u=-\infty}^{\infty} \int_{\Delta v=-\infty}^{\infty} P_{p,2,3,4}(\Delta u, \Delta v) \times P_{a,2}(\Delta u, \Delta v) d\Delta v d\Delta u \\
&= n_{\text{nodes}} \times (n_{\text{sol}} - 1) \times f_{\text{norm},2}^2 \times \\
&\left( \int_{\Delta u=-\infty}^{\infty} [(1-Q)P_{o,1}(\Delta u) + (n_{\text{sol}} - 1 + Q - 1)P_{o,2}(\Delta u)] \times \right. \\
&\quad \left. [P_{o,1}(\Delta u) + P_{o,2}(\Delta u)] P_b(\Delta u) d\Delta u \right)^2 \\
&= n_{\text{nodes}} \times (n_{\text{sol}} - 1) \times f_{\text{norm},2}^2 \times \\
&\left( \int_{\Delta u=-\infty}^{\infty} [(1-Q)P_{o,1}^2(\Delta u) + (n_{\text{sol}} - 1 + Q - 1)P_{o,2}^2(\Delta u) + \right. \\
&\quad \left. (n_{\text{sol}} - 1)P_{o,1}(\Delta u)P_{o,2}(\Delta u)] P_b(\Delta u) d\Delta u \right)^2 \\
&= n_{\text{nodes}} \times (n_{\text{sol}} - 1) \times f_{\text{norm},2}^2 \times \\
&\left( \int_{\Delta u=-\infty}^{\infty} \left[ (1-Q)c_1^2 \exp\left(\frac{-\Delta u^2}{\sigma_1^2} + \frac{-\Delta u^2}{2\sigma_b^2}\right) + \right. \right. \\
&\quad \left. \left. (n_{\text{sol}} - 1 + Q - 1)c_2^2 \exp\left(\frac{-\Delta u^2}{\sigma_2^2} + \frac{-\Delta u^2}{2\sigma_b^2}\right) + \right. \right. \\
&\quad \left. \left. (n_{\text{sol}} - 1)c_1c_2 \exp\left(\frac{-\Delta u^2}{2\sigma_1^2} + \frac{-\Delta u^2}{2\sigma_2^2} + \frac{-\Delta u^2}{2\sigma_b^2}\right) \right] d\Delta u \right)^2 \\
&= n_{\text{nodes}} \times (n_{\text{sol}} - 1) \times \frac{n_{\text{sol}}^2}{[(1-Q) + (n_{\text{sol}} - 1 + Q - 1)(n_{\text{sol}} - 1)]^2} \times \\
&\left( (1-Q)c_1^2\sigma_{1,b}\sqrt{\pi} + (n_{\text{sol}} - 1 + Q - 1)c_2^2\sigma_{2,b}\sqrt{\pi} + (n_{\text{sol}} - 1)^2c_1c_2\sigma_{1,2,b}\sqrt{2\pi} \right)^2 \tag{56}
\end{aligned}$$

So, after inserting the values for  $c_1$ ,  $c_2$  and  $f_{\text{norm},1}$  this becomes:

$$\begin{aligned}
r_r &= \frac{n_{\text{nodes}}(n_{\text{sol}} - 1)}{4\pi n_{\text{sol}}^2 \times [(1-Q) + (n_{\text{sol}} - 1 + Q - 1)(n_{\text{sol}} - 1)]^2} \times \\
&\left( \frac{(1-Q)\sigma_{1,b}}{\sigma_1^2} + \frac{(n_{\text{sol}} - 1 + Q - 1)(n_{\text{sol}} - 1)^2\sigma_{2,b}}{\sigma_2^2} + \frac{(n_{\text{sol}} - 1)^2\sqrt{2}\sigma_{1,2,b}}{\sigma_1\sigma_2} \right)^2 \tag{57}
\end{aligned}$$

In an ideal case,  $Q=1$  and  $n_{\text{sol}}=2$ , which gives:  $r_1 = 0.315 \times \frac{n_{\text{nodes}}}{16\pi}$  and  $r_r = 0.010 \times \frac{n_{\text{nodes}}}{16\pi}$ , so score<sub>r</sub> = 0.064.

However, for the ERS case ( $n_{\text{sol}}=2$ ) both ranks have (almost) equal probability (i.e.  $Q = 0.5$ ). From this it follows that  $r_1 = r_r$ . So equation 11 gives score<sub>r</sub> = 1.0. Combining the scores for 2 solutions, the FoM becomes  $0.8 \times 0.759 + 0.2 \times 1.0 = 0.807$  for  $Q=0.5$ , and FoM=0.620 for  $Q=1$ .

### 4.3.3 2 solutions, general case, exact calculation

A similar calculation for 2 solutions, in which the second solution is uncorrelated to the first rank solution, uses a SD of  $\sigma_2 = \sqrt{2}\sigma_w = 7.778$  m/s. This gives again for  $Q=1$  and  $n_{\text{sol}}=2$ :  $r_1 = 0.359 \times \frac{n_{\text{nodes}}}{16\pi}$  and  $r_r = 0.023 \times \frac{n_{\text{nodes}}}{16\pi}$ , so score<sub>r</sub> = 0.121.

However, for the ERS case ( $n_{\text{sol}}=2$ ) both ranks have (almost) equal probability (i.e.  $Q = 0.5$ ). From this it follows that  $r_1 = r_r$ . So equation 11 gives  $\text{score}_r = 1.0$ .

In section 4.1.2 a value of  $\text{score}_u = 0.773$  was found, which is not changed by the changing definition of  $\text{score}_r$ . Combining the scores for 2 uncorrelated solutions, the FoM becomes  $0.8 \times 0.773 + 0.2 \times 1.0 = 0.818$  for  $Q=0.5$  and  $\text{FoM} = 0.643$  for  $Q=1$ .

#### 4.3.4 3 and more solutions, exact calculation

Also in the case of three or more ambiguous solutions the exact calculation for implementation 3 only differs in the  $\text{score}_r$ .

So again in case of 3 solutions  $\text{score}_u = 0.814$ .

To calculate the ranking score for this case, equations 55 and 57 are used, with  $Q=0.5$  and  $n_{\text{sol}}=3$ :

$$r_1 = \frac{n_{\text{nodes}}}{4\pi \times 9 \times \left(\frac{3}{2}\right)^2} \times \left( \frac{\frac{1}{2}\sigma_{1,b}}{\sigma_1^2} + \frac{4\sigma_{2,b}}{\sigma_2^2} + \frac{2\sqrt{2}\sigma_{1,2,b}}{\sigma_1\sigma_2} \right)^2 \quad (58)$$

and:

$$r_r = \frac{n_{\text{nodes}} \times 2}{4\pi \times 9 \times \left(\frac{7}{2}\right)^2} \times \left( \frac{\frac{1}{2}\sigma_{1,b}}{\sigma_1^2} + \frac{6\sigma_{2,b}}{\sigma_2^2} + \frac{4\sqrt{2}\sigma_{1,2,b}}{\sigma_1\sigma_2} \right)^2 \quad (59)$$

Now we fill in the values for  $\sigma_1$  and  $\sigma_2$  and get  $\text{score}_r = 1.002$  for  $Q=0.5$ , and  $\text{score}_r = 0.210$  for  $Q=1$ . Combining the scores for the 3 solution case, the FoM becomes  $0.8 \times 0.814 + 0.2 \times 1.002 = 0.852$  for  $Q=0.5$ , and  $\text{FoM} = 0.693$  for  $Q=1$ .

In case of 4 solutions  $\text{score}_u = 0.841$  still remains the same.

To calculate the ranking score for this case, equations 55 and 57 are used, with  $Q=0.5$  and  $n_{\text{sol}}=4$ :

$$r_1 = \frac{n_{\text{nodes}}}{4\pi \times 16 \times 2^2} \times \left( \frac{\frac{1}{2}\sigma_{1,b}}{\sigma_1^2} + \frac{\frac{1}{2} \times 3^2\sigma_{2,b}}{\sigma_2^2} + \frac{3\sqrt{2}\sigma_{1,2,b}}{\sigma_1\sigma_2} \right)^2 \quad (60)$$

and:

$$r_r = \frac{n_{\text{nodes}} \times 3}{4\pi \times 16 \times 8^2} \times \left( \frac{\frac{1}{2}\sigma_{1,b}}{\sigma_1^2} + \frac{\frac{5}{2} \times 3^2\sigma_{2,b}}{\sigma_2^2} + \frac{3^2\sqrt{2}\sigma_{1,2,b}}{\sigma_1\sigma_2} \right)^2 \quad (61)$$

From this we obtain  $\text{score}_r = 1.138$ . Combining the 3 scores for this case, the FoM becomes  $0.8 \times 0.841 + 0.2 \times 1.138 = 0.900$  for  $Q=0.5$ . In a similar way we obtain  $\text{score}_r = 0.284$ , so  $\text{FoM} = 0.8 \times 0.841 + 0.2 \times 0.284 = 0.730$  for  $Q=1$ .

#### 4.3.5 Synthetic solutions

Also implementation 3 of the FoM calculation was tested by generating artificial windvector solutions, and using them as input for the FoM routine. The results are summarised in table 3.

- the 1 solution case is independant of the  $\text{score}_r$  so is identical to implementation 1.

Table 3: Values for the FoM for implementation 3.

$n_{\text{sol}}$	$\text{score}_u$	$\text{score}_v$	$\text{score}_r$ ( $Q = \frac{1}{2}$ )	$\text{score}_r$ ( $Q = 1$ )	FoM ( $Q = \frac{1}{2}$ )	FoM ( $Q = 1$ )
Exact values						
1	0.707	0.707	0.0	0.0	0.566	0.566
2 (180 deg.amb.)	0.759	0.759	1.0	0.064	0.807	0.620
2 (uncorrelated)	0.773	0.773	1.0	0.121	0.818	0.643
3 (uncorrelated)	0.814	0.814	1.002	0.210	0.852	0.693
4 (uncorrelated)	0.841	0.841	1.138	0.284	0.900	0.730
Values resulting from synthetic windvectors						
1	0.716	0.697	0.0	0.0	0.565	0.565
2 (180 deg.amb.)	0.759	0.759	0.992	0.059	0.806	0.619
2 (random angle)	0.772	0.770	0.981	0.280	0.813	0.673
2 (uncorrelated)	0.773	0.778	0.963	0.103	0.813	0.641
$Q = \frac{1}{2}$						
3 (125 deg.amb.)	0.805	0.799	0.902		0.822	
3 (random angle)	0.805	0.804	1.006		0.845	
3 (uncorrelated)	0.809	0.816	0.953		0.841	
4 (random angle)	0.826	0.822	1.002		0.860	
4 (uncorrelated)	0.838	0.842	0.953		0.863	
$Q = 1$						
3 (125 deg.amb.)	0.807	0.798		0.117		0.665
3 (random angle)	0.810	0.801		0.326		0.710
3 (uncorrelated)	0.815	0.817		0.220		0.697
4 (random angle)	0.827	0.823		0.361		0.732
4 (uncorrelated)	0.836	0.848		0.311		0.736
Real data						
2 (real ERS data)	0.761	0.742		0.743		0.750
var (real Seawinds data)	0.725	0.701		0.235		0.617

- the case with 2 solutions, with 180 degree ambiguity and 50 % swapping ( $Q=0.5$ ) yielded:  $\text{score}_u = 0.759$ ,  $\text{score}_v = 0.759$ , and  $\text{score}_r = 0.992$ , so  $\text{FoM} = 0.806$ . Which is very close to both the analytic results and the results for implementation 1. For  $Q=1$  these numbers become:  $\text{score}_u = 0.759$ ,  $\text{score}_v = 0.759$ , and  $\text{score}_r = 0.059$ , so  $\text{FoM} = 0.619$ . So as expected the only change is in the  $\text{score}_r$ . Again this is very close to the expected analytic result, but now a clear deviation is seen from implementation 1. This is also clearly seen in figure 2. Implementation 3 has a lower  $\text{score}_r$  for all values of the SD of the closest wind solution, which can be understood by the fact that  $P_a$  has a narrower PDF than  $P_o$ . Therefore implementation 3 has lower values for  $r_r$  than implementation 1, since this is dominated by the solution at 180 degrees. This effect is also clearly seen in the angle scan shown in figure 3.

The dotted blue line in figure 2 is the analytical result for  $\text{score}_r$  for implementation 3, based on equations 55 and 57. It does not exactly match the synthetic results, but



is still pretty close.

- When this second solution is given a random angle compared to the first solution, and  $Q=0.5$ , this yields:  $\text{score}_u = 0.772$ ,  $\text{score}_v = 0.770$ , and  $\text{score}_r = 0.981$ , so  $\text{FoM} = 0.813$ . Again this is almost identical to the results of implementation 1, and to the analytic results. For  $Q=1$  this changes to  $\text{score}_r = 0.280$  and  $\text{FoM} = 0.673$ .
- When this second solution is taken completely uncorrelated to the first solution, and  $Q=0.5$ , this yields:  $\text{score}_u = 0.773$ ,  $\text{score}_v = 0.778$ , and  $\text{score}_r = 0.963$ , so  $\text{FoM} = 0.813$ . This again is very close to the expected values of:  $\text{score}_u = 0.773$ ,  $\text{score}_v = 0.773$ ,  $\text{score}_r = 1.0$ , and  $\text{FoM} = 0.818$ . For  $Q=1$   $\text{score}_r = 0.103$  and  $\text{FoM} = 0.641$ , which again is close to the expected theoretical values of  $\text{score}_r = 0.121$  and  $\text{FoM} = 0.643$ .
- For the case with 3 solutions, with an ambiguity of  $\pm 125$  degrees between the 1st and both other solutions, and  $Q=0.5$ , the numbers are:  $\text{score}_u = 0.805$ ,  $\text{score}_v = 0.799$ , and  $\text{score}_r = 0.902$ , so  $\text{FoM} = 0.822$ . For  $Q=1$  the scores are:  $\text{score}_u = 0.807$ ,  $\text{score}_v = 0.798$ , and  $\text{score}_r = 0.117$ , so  $\text{FoM} = 0.665$ .
- For the case with 3 solutions (with random angle between the 1st rank solution and both other solutions), and  $Q=0.5$ , the numbers are:  $\text{score}_u = 0.805$ ,  $\text{score}_v = 0.804$ , and  $\text{score}_r = 1.006$ , so  $\text{FoM} = 0.845$ . For  $Q=1$  the scores are:  $\text{score}_u = 0.810$ ,  $\text{score}_v = 0.801$ , and  $\text{score}_r = 0.326$ , so  $\text{FoM} = 0.710$ .
- For the case with 3 solutions taken completely uncorrelated to each other, and  $Q=0.5$ , the numbers are:  $\text{score}_u = 0.809$ ,  $\text{score}_v = 0.816$ , and  $\text{score}_r = 0.953$ , so  $\text{FoM} = 0.841$ . The  $\text{score}_u$  and  $\text{score}_v$  are very close to the expected values of  $\text{score}_u = 0.814$ ,  $\text{score}_v = 0.814$ , and are identical to the results of implementation 1. However the  $\text{score}_r$  and the  $\text{FoM}$  deviate somewhat from the expected values of  $\text{score}_r = 1.001$ , and  $\text{FoM} = 0.852$  and are clearly below the values of implementation 1. For  $Q=1$  the scores are:  $\text{score}_u = 0.815$ ,  $\text{score}_v = 0.817$ , and  $\text{score}_r = 0.220$ , so  $\text{FoM} = 0.697$ . Now all scores and the  $\text{FoM}$  are again very close to the expected theoretical values of:  $\text{score}_u = 0.814$ ,  $\text{score}_v = 0.814$ ,  $\text{score}_r = 0.210$  and  $\text{FoM} = 0.693$ . Again the  $\text{score}_r$  and the  $\text{FoM}$  are clearly below the values of implementation 1.
- For the case with 4 solutions (with random angle between the 1st rank solution and the other solutions), and  $Q=0.5$ , the numbers are:  $\text{score}_u = 0.826$ ,  $\text{score}_v = 0.822$ , and  $\text{score}_r = 1.002$ , so  $\text{FoM} = 0.860$ . For  $Q=1$  the scores are:  $\text{score}_u = 0.827$ ,  $\text{score}_v = 0.823$ , and  $\text{score}_r = 0.361$ , so  $\text{FoM} = 0.732$ . Compared to implementation 1 the value of  $\text{score}_r$  has dropped by ca. 50 %.
- For the case with 4 solutions taken completely uncorrelated to each other, and  $Q=0.5$ , the numbers are:  $\text{score}_u = 0.838$ ,  $\text{score}_v = 0.842$ , and  $\text{score}_r = 0.953$ , so  $\text{FoM} = 0.863$ . The  $\text{score}_u$  and  $\text{score}_v$  are very close to the expected values of  $\text{score}_u = 0.841$ ,  $\text{score}_v = 0.841$ . However the  $\text{score}_r$  and the  $\text{FoM}$  deviate significantly from the expected values of  $\text{score}_r = 1.138$ , and  $\text{FoM} = 0.900$ . For  $Q=1$  the scores are:  $\text{score}_u = 0.836$ ,  $\text{score}_v = 0.848$ , and  $\text{score}_r = 0.311$ , so  $\text{FoM} = 0.736$ . These scores again are close to the expected values for  $Q=1$  of  $\text{score}_u = 0.841$ ,  $\text{score}_v = 0.841$ ,  $\text{score}_r = 0.284$  and  $\text{FoM} = 0.730$ . Again, it is clear the  $\text{score}_r$  has dropped by more than 50 % compared to implementation 1. This effect on  $\text{score}_r$  is also very clear in figure 6.

### 4.3.6 Comparison with ERS data

For implementation 3 the FoM for the ERS geometry was also tested by applying it to real ERS data. For this the same data is used as was described in section 4.1.6. The node-averaged mean results are:  $\text{score}_u = 0.761$ ,  $\text{score}_v = 0.742$ ,  $\text{score}_r = 0.743$ ,  $\text{FoM} = 0.750$ . From this it is clear that the result for implementation 3 only differs for  $\text{score}_r$ . However, then effect is very small, which is not surprising due to the large mixing percentage for this case.

### 4.3.7 Comparison with Seawinds data

For implementation 3 the FoM for the Seawinds geometry was also tested by applying it to real Seawinds data. For this the same data is used as was described in section 4.1.7. The node-averaged mean results are:  $\text{score}_u = 0.725$ ,  $\text{score}_v = 0.701$ ,  $\text{score}_r = 0.235$  and  $\text{FoM} = 0.617$ . Now a large effect is seen on  $\text{score}_r$ , but due to the small weighting factor the effect on the FoM value is still modest.

## 4.4 Implementation 4

Table 4: Values for the FoM for implementation 4.

$n_{\text{sol}}$	$\text{score}_u$	$\text{score}_v$	$\text{score}_r$ ( $Q = \frac{1}{2}$ )	$\text{score}_r$ ( $Q = 1$ )	FoM ( $Q = \frac{1}{2}$ )	FoM ( $Q = 1$ )
	$Q = \frac{1}{2}$					
3 (125 deg.amb.)	0.802	0.798	0.901		0.820	
3 (random angle)	0.810	0.805	1.006		0.847	
3 (uncorrelated)	0.810	0.814	0.953		0.840	
4 (random angle)	0.802	0.795	1.001		0.839	
4 (uncorrelated)	0.839	0.840	0.953		0.862	
	$Q = 1$					
3 (125 deg.amb.)	0.804	0.792		0.115		0.661
3 (random angle)	0.808	0.810		0.326		0.713
3 (uncorrelated)	0.816	0.819		0.221		0.698
4 (random angle)	0.803	0.798		0.357		0.712
4 (uncorrelated)	0.835	0.847		0.310		0.735
Real data						
var (real Seawinds data)	0.728	0.703		0.236		0.620

Again giving different weights to the different solutions, based on the angle-sector associated with each solution, only makes a difference for 3 or 4 solutions. These weights also complicate the exact calculation of the FoM so this has not been attempted for this implementation. The check with synthetic solutions has been done, as is described below, and also a comparison with real Seawinds data is performed.

#### 4.4.1 Synthetic solutions

Also this fourth implementation of the FoM calculation was tested by generating artificial windvector solutions, and using them as input for the FoM routine. 5001 windvectors were generated in each example. The results are summarised in table 4.

- First a case with 3 solutions is simulated. The first solution has a SD of 1.5 m/s. Then the second and third solution are constructed by adding and subtracting 125 degrees to the first solution. and  $Q=1$  was taken. When these synthetic solutions are used as input for the FoM calculation this yields:  $score_u = 0.804$ ,  $score_v = 0.792$ , and  $score_r = 0.115$ , so  $FoM = 0.661$ . Again the large difference in  $score_r$  compared to implementation 2 is very clear
- For the case with 3 solutions (with random angle between the 1st rank solution and both other solutions) and  $Q=1$  the numbers are:  $score_u = 0.808$ ,  $score_v = 0.810$ , and  $score_r = 0.326$ , so  $FoM = 0.713$ . These results are all very close to the synthetic results for implementation 3, but again the  $score_r$  differs clearly when compared to the result for implementation 2.
- For the case with 3 solutions taken completely uncorrelated to each other, and  $Q=1$ , the numbers are:  $score_u = 0.816$ ,  $score_v = 0.819$ , and  $score_r = 0.221$ , so  $FoM = 0.698$ . Also these results are all very close to the synthetic results for implementation 3 and again the  $score_r$  differs clearly when compared to the result for implementation 2.
- For the case of 3 solutions, a scan of the SD was done. In the calculations for this scan the swapping was switched off ( $Q=1$ ) and an angle between the first rank and both other solutions of 125 degrees was taken The results are given in Fig. 4 on page 26. From this figure it can clearly be seen that implementations 3 and 4 give very similar results, but they differ clearly from implementations 1 and 2 in the  $score_r$ .
- Also for this 3 solution case, the angle between the 1st rank and the other 2 solutions was scanned (for  $Q=1$ ). The results are given in Fig. 5 on page 27. It is clear from this figure that the  $score_r$  is much more peaked around the firstrank solution for implementations 3 and 4, than it is for implementations 1 and 2. Another noticeable effect is that the  $score_r$  is almost identical for implementations 3 and 4, so giving different weights to the different solutions seems to be a waste of effort.
- For the case with 4 solutions (with random angle between the 1st rank solution and the other solutions) and  $Q=1$  the numbers are:  $score_u = 0.803$ ,  $score_v = 0.798$ , and  $score_r = 0.357$ , so  $FoM = 0.712$ . Again the same behaviour is seen, a much smaller  $score_r$  compared to implementation 2, but just a small difference with implementation 3.
- For the case with 4 solutions taken completely uncorrelated to each other, and  $Q=1$ , the numbers are:  $score_u = 0.835$ ,  $score_v = 0.847$ , and  $score_r = 0.310$ , so  $FoM = 0.735$ , so again a clear effect in the  $score_r$ .
- Finally a case with 4 solutions was simulated. The angle between the 1st rank and the 2nd and 3rd solution was scanned. The angle between the 1st rank and the 4th

solution was taken to be 180 degrees, and  $Q=1$  was taken. The results are given in Fig. 6 on page 28, and show the same effects as for the 3 solution case described above.

#### **4.4.2 Comparison with Seawinds data**

The FoM for the Seawinds geometry for implementation 4 was also tested by applying it to real Seawinds data. For this the same data is used as was described in section 4.1.7. The node-averaged mean results are:  $\text{score}_u = 0.728$ ,  $\text{score}_v = 0.703$ ,  $\text{score}_r = 0.236$ ,  $\text{FoM} = 0.620$ . These scores are very close to the ones found for implementation 3, but the  $\text{score}_r$  is clearly different from the one found in implementations 1 and 2.

## 5 Conclusions

### 5.1 implementations 1 and 2

For 1 and 2 solutions the first implementation of the new Figure of Merit is very well understood. The exact calculations and the synthetic approach agree very well with each other. Also the real ERS data agrees very well with the results.

The 3 components clearly have some desired properties. The  $\text{score}_u$  and  $\text{score}_v$  become worse (i.e. higher) for larger SD or larger angle differences. The  $\text{score}_r$  clearly becomes worse for a higher number of solutions, which is also a good feature.

Also, the  $\text{score}_r$  improves for a larger angle difference which partly counteracts the behaviour of the  $\text{score}_u$  and  $\text{score}_v$ .

From figure 1 it is clear that the current FoM is not fully optimised, since most of the variation in  $\text{score}_u$  and  $\text{score}_v$  occurs for a SD between 0 and 1.5 m/s. More desirable would be a perfect score of 0 when the SD of the solutions equals the (for this resolution of 50 km) expected SD of 1.5 m/s, and a degradation of the FoM for higher values of the SD.

Note also that a FoM is usually defined in such a way that a value of 1 is very good, and 0 is very bad. To get this the FoM is rewritten as:  $FoM' = 1 - FoM$ , which is the form used in [3].

For the case of 3 and 4 solutions the same behaviour is seen. The  $\text{score}_u$  and  $\text{score}_v$  increase for larger SD. They also increase for a larger angle difference, but go down again for angle differences larger than 50 degrees. This effect is a little more pronounced in implementation 2, but still the range in which these scores vary is rather small.

The  $\text{score}_r$  again improves for a larger angle difference which partly counteracts the behaviour of the  $\text{score}_u$  and  $\text{score}_v$ .

The analytic results for implementation 1 agree very well with the synthetic (simulated) results. For  $Q=1$  (no mixing of 1st rank with the other solutions) the results are in perfect agreement. For  $Q=0.5$  (50 % of all 1st rank solutions is swapped with one of the other solutions) the  $\text{score}_r$  shows some differences, especially for the 3 and 4 solution case.

Implementation 2 of the FoM has only been tested with synthetic solutions. The differences between implementations 1 and 2 are very small. The trends of all the curves stay the same, and only for  $\text{score}_r$  in the 4 solution case a significant difference is seen for large angle differences (between 130-180 degrees).

This difference might be increased by taking the solution probability after inversion (residual) instead of the sectors between the solutions as weights.

Implementation 2 has also been applied to real Seawinds data. The results can be explained by assuming a better background wind quality than was used in the calculations. This is explained by the fact that the background ECMWF wind is of better quality, than was initially assumed in the calculation [9].

### 5.2 implementations 3 and 4

For implementations 3 and 4 there is no difference in  $\text{score}_u$  and  $\text{score}_v$ , so all remarks above remain valid.

The  $\text{score}_r$  does show some differences. It does not increase as fast with SD, as it does in implementations 1 and 2, and it drops much faster with the angle difference between the

solutions. Both effects can be understood when keeping in mind the different widths of the  $P_o$  and the  $P_a$  distributions.

Another clear effect is that  $\text{score}_r$  is almost identical in implementations 3 and 4, so the different weights that are given to the solutions in implementation 4 have almost no effect.

However, the general trends of all the curves remains the same. Therefore all four implementations are equally suitable for use in optimising the parameters for scatterometer performance.

### 5.3 overall conclusion

From the angle scans for 2, 3 and 4 solutions it can be seen that the amount of variation of the FoM is not large, and is dominated by the  $\text{score}_r$  behaviour.

The SD scans however do show a large variation, so this effect probably dominates when these measures are used for optimisation of an instrument.

Interpreting the absolute numbers of one of these FoM implementations remains difficult. The results should only be used to compare to a FoM value for a known situation and to decide if it is better or worse.

The overall conclusion is thus that these Figure of Merit implementations are usefull, but not completely optimised, and there certainly is room for improvement/tuning of the FoM.

It should also be noted that the optimal FoM is about 0.6, for a reference wind with SD of 1.5 m/s. Improvement of the instrument will not yield better results than this.

Looking at the fact that the results are almost identical for implementations 3 and 4, giving different weights to the different solutions seems to be not very effective. Using the solution probability as weight in the FoM implementation would probably improve the FoM, but is not feasible at this moment.

## 6 Appendix A: FoM program code

The following code, suitable for PVWAVE and IDL, was used to calculate the new FoM:

```

;-----
; implemented by: Jos de Kloe
; last change: 20-02-2003
;
; version 5
;
; reference implementation of the FoM for all four versions
; (not optimised on runtime or memory usage)
;-----
; interface of the ads_fom_all subroutine:
;
; function call:
;   ads_fom_all, FoM, score_r, score_u, score_v, $
;               max_nr_of_wvcs, nws, wvcarr, inp_u, inp_v, $
;               found_u1, found_v1, found_u2, found_v2, $
;               found_u3, found_v3, found_u4, found_v4, $
;               dirs, numsol, resolution, implementation
;
; outputs:
; 4 real arrays with dimension [0..max_nr_of_wvcs-1]
;   FoM      : the FoM value for each WVC
;   score_r  : the score_r for each WVC
;   score_u  : the score_u for each WVC
;   score_v  : the score_v for each WVC
;
; inputs:
; 2 integer numbers
;   max_nr_of_wvcs : number of WVC's present in the data
;   nws            : number of nodes to process
;
; an integer array with dimension [0..nws-1]
;   wvcarr        : listt the WVC number for each node
;
; 10 real arrays with dimension [0..nws-1]
;   inp_u, inp_v   : the true/background windcomponents for each node
;   found_u1, found_v1 : the found windcomponents for solution 1 for each node
;   found_u2, found_v2 : the found windcomponents for solution 2 for each node
;   found_u3, found_v3 : the found windcomponents for solution 3 for each node
;   found_u4, found_v4 : the found windcomponents for solution 4 for each node
;
; a two dimensional real arrays with dimension [0..nws-1][0..3]
;   dirs          : the direction of the windvector for each node

```

```

;               for each solution
;
; an integer array with dimension [0..nws-1]
;   numsol           : number of found solutions for each node
;
; one real number
;   resolution       : the measurement resolution in km
;
; one integer number
;   implementation   : the implementation [1,2,3 or 4]
;-----
;   two further switches can be set to false/true to
;   return to the versions that contain the two described bugs:
;
;   -apply_correction_to_implementation_2
;   -apply_correction_to_weights
;-----

function FoM_calc_weight,numsol,sol,dir1,dir2,dir3,dir4,implementation

  true = 1 eq 1 & false = 1 eq 0

  IF ( (implementation eq 1) OR $
        (implementation eq 3)      ) THEN equal_weights = true

  IF ( (implementation eq 2) OR $
        (implementation eq 4)      ) THEN equal_weights = false

  weight = 1.0

  apply_correction_to_weights = true
;  apply_correction_to_weights = false

  IF (numsol eq 1) THEN weight = 1.0
  IF (numsol eq 2) THEN weight = 0.5

  IF (numsol eq 3) THEN $
  BEGIN
    IF (equal_weights) THEN $
    BEGIN
      weight = 1./3
    END ELSE $
    BEGIN
      IF (apply_correction_to_weights) THEN $
      BEGIN ; sort the 3 directions

```



```
dirs = [dir1,dir2,dir3]
dirs = dirs      *(dirs ge  0.)*(dirs lt 360.) + $
          (dirs-360.)*(dirs ge 360.) + $
          (dirs+360.)*(dirs lt  0.)

sort_result = sort(dirs)
sdir1 = dirs(sort_result(0))
sdir2 = dirs(sort_result(1))
sdir3 = dirs(sort_result(2))

IF (sol eq 1) THEN $
BEGIN
  IF (dirs(0) eq sdir1) THEN ssol=1
  IF (dirs(0) eq sdir2) THEN ssol=2
  IF (dirs(0) eq sdir3) THEN ssol=3
END
IF (sol eq 2) THEN $
BEGIN
  IF (dirs(1) eq sdir1) THEN ssol=1
  IF (dirs(1) eq sdir2) THEN ssol=2
  IF (dirs(1) eq sdir3) THEN ssol=3
END
IF (sol eq 3) THEN $
BEGIN
  IF (dirs(2) eq sdir1) THEN ssol=1
  IF (dirs(2) eq sdir2) THEN ssol=2
  IF (dirs(2) eq sdir3) THEN ssol=3
END

END ELSE $
BEGIN
  sdir1 = dir1
  sdir2 = dir2
  sdir3 = dir3
  ssol = sol
END

IF (ssol eq 1) THEN $
BEGIN
  IF (apply_correction_to_weights) THEN $
  BEGIN
    sector =      sdir2-sdir3
    IF (sector gt 360.0) THEN sector = sector - 360.
    IF (sector le  0.0) THEN sector = sector + 360.
    sector = 0.5*sector
  END ELSE $
```

```

BEGIN
  sector = 0.5*(sdir2-sdir3)
  IF (sector ge 360.0) THEN sector = sector - 360.
  IF (sector lt  0.0) THEN sector = sector + 360.
END
END
IF (ssol eq 2) THEN $
BEGIN
  IF (apply_correction_to_weights) THEN $
  BEGIN
    sector =      sdir3-sdir1
    IF (sector gt 360.0) THEN sector = sector - 360.
    IF (sector le  0.0) THEN sector = sector + 360.
    sector = 0.5*sector
  END ELSE $
  BEGIN
    sector = 0.5*(sdir3-sdir1)
    IF (sector ge 360.0) THEN sector = sector - 360.
    IF (sector lt  0.0) THEN sector = sector + 360.
  END
END
IF (ssol eq 3) THEN $
BEGIN
  IF (apply_correction_to_weights) THEN $
  BEGIN
    sector =      sdir1-sdir2
    IF (sector gt 360.0) THEN sector = sector - 360.
    IF (sector le  0.0) THEN sector = sector + 360.
    sector = 0.5*sector
  END ELSE $
  BEGIN
    sector = 0.5*(sdir1-sdir2)
    IF (sector ge 360.0) THEN sector = sector - 360.
    IF (sector lt  0.0) THEN sector = sector + 360.
  END
END
  weight = sector/360.0

END ; end of un-equal weights case
END ; end of code for 3 solutions

IF (numsol eq 4) THEN $
BEGIN
  IF (equal_weights) THEN $
  BEGIN
    weight = 1./4

```

```

END ELSE $
BEGIN

  IF (apply_correction_to_weights) THEN $
  BEGIN ; sort the 4 directions
    dirs = [dir1,dir2,dir3,dir4]
    dirs = dirs      *(dirs ge  0)*(dirs lt 360) + $
              (dirs-360)*(dirs ge 360) + $
              (dirs+360)*(dirs lt  0)

    sort_result = sort(dirs)
    sdir1 = dirs(sort_result(0))
    sdir2 = dirs(sort_result(1))
    sdir3 = dirs(sort_result(2))
    sdir4 = dirs(sort_result(3))

    IF (sol eq 1) THEN $
    BEGIN
      IF (dirs(0) eq sdir1) THEN ssol=1
      IF (dirs(0) eq sdir2) THEN ssol=2
      IF (dirs(0) eq sdir3) THEN ssol=3
      IF (dirs(0) eq sdir4) THEN ssol=4
    END
    IF (sol eq 2) THEN $
    BEGIN
      IF (dirs(1) eq sdir1) THEN ssol=1
      IF (dirs(1) eq sdir2) THEN ssol=2
      IF (dirs(1) eq sdir3) THEN ssol=3
      IF (dirs(1) eq sdir4) THEN ssol=4
    END
    IF (sol eq 3) THEN $
    BEGIN
      IF (dirs(2) eq sdir1) THEN ssol=1
      IF (dirs(2) eq sdir2) THEN ssol=2
      IF (dirs(2) eq sdir3) THEN ssol=3
      IF (dirs(2) eq sdir4) THEN ssol=4
    END
    IF (sol eq 4) THEN $
    BEGIN
      IF (dirs(3) eq sdir1) THEN ssol=1
      IF (dirs(3) eq sdir2) THEN ssol=2
      IF (dirs(3) eq sdir3) THEN ssol=3
      IF (dirs(3) eq sdir4) THEN ssol=4
    END
  END

```

; remaining problem (20-3-2003):

```
; in the case that 2 or 3 solutions have identical directions
; the ordering of the solutions is wrong.
; example: if dir2=dir3=dir4 then ssol will become 4
;         independant of the value of sol, for sol=2,3 or 4.
; however, since this never could occur in real scatterometer
; data, I have not yet tried to fix this.
```

```
END ELSE $
```

```
BEGIN
```

```
  sdir1 = dir1
```

```
  sdir2 = dir2
```

```
  sdir3 = dir3
```

```
  sdir4 = dir4
```

```
  ssol = sol
```

```
END
```

```
IF (ssol eq 1) THEN $
```

```
BEGIN
```

```
  difs = [sdir2-sdir1,sdir3-sdir1,sdir4-sdir1]
```

```
  IF (apply_correction_to_weights) THEN $
```

```
    leftneighbours = difs*(difs ge 0.0)*(difs le 360) + $
```

```
                    (difs+360.0)*(difs lt  0.0) + $
```

```
                    (difs-360.0)*(difs gt 360.0) $
```

```
  ELSE $
```

```
    leftneighbours = difs*(difs ge 0.0) + (360.0+difs)*(difs lt 0.0)
```

```
  leftneighbour  = min(leftneighbours)
```

```
  difs = [sdir1-sdir2,sdir1-sdir3,sdir1-sdir4]
```

```
  IF (apply_correction_to_weights) THEN $
```

```
    rightneighbours = difs*(difs ge 0.0)*(difs le 360) + $
```

```
                    (difs+360.0)*(difs lt  0.0) + $
```

```
                    (difs-360.0)*(difs gt 360.0) $
```

```
  ELSE $
```

```
    rightneighbours = difs*(difs ge 0.0) + (360.0+difs)*(difs lt 0.0)
```

```
  rightneighbour = min(rightneighbours)
```

```
  sector = 0.5*(leftneighbour+rightneighbour)
```

```
END
```

```
IF (ssol eq 2) THEN $
```

```
BEGIN
```

```
  difs = [sdir1-sdir2,sdir3-sdir2,sdir4-sdir2]
```

```
  IF (apply_correction_to_weights) THEN $
```

```

    leftneighbours = difs*(difs ge 0.0)*(difs le 360) + $
                    (difs+360.0)*(difs lt 0.0) + $
                    (difs-360.0)*(difs gt 360.0) $
ELSE $
    leftneighbours = difs*(difs ge 0.0) + (360.0+difs)*(difs lt 0.0)

leftneighbour = min(leftneighbours)
difs = [sdir2-sdir1,sdir2-sdir3,sdir2-sdir4]

IF (apply_correction_to_weights) THEN $
    rightneighbours = difs*(difs ge 0.0)*(difs le 360) + $
                    (difs+360.0)*(difs lt 0.0) + $
                    (difs-360.0)*(difs gt 360.0) $
ELSE $
    rightneighbours = difs*(difs ge 0.0) + (360.0+difs)*(difs lt 0.0)

rightneighbour = min(rightneighbours)
sector = 0.5*(leftneighbour+rightneighbour)
END

IF (ssol eq 3) THEN $
BEGIN
    difs = [sdir1-sdir3,sdir2-sdir3,sdir4-sdir3]

    IF (apply_correction_to_weights) THEN $
        leftneighbours = difs*(difs ge 0.0)*(difs le 360) + $
                        (difs+360.0)*(difs lt 0.0) + $
                        (difs-360.0)*(difs gt 360.0) $
    ELSE $
        leftneighbours = difs*(difs ge 0.0) + (360.0+difs)*(difs lt 0.0)

    leftneighbour = min(leftneighbours)
    difs = [sdir3-sdir1,sdir3-sdir2,sdir3-sdir4]

    IF (apply_correction_to_weights) THEN $
        rightneighbours = difs*(difs ge 0.0)*(difs le 360) + $
                        (difs+360.0)*(difs lt 0.0) + $
                        (difs-360.0)*(difs gt 360.0) $
    ELSE $
        rightneighbours = difs*(difs ge 0.0) + (360.0+difs)*(difs lt 0.0)

    rightneighbour = min(rightneighbours)
    sector = 0.5*(leftneighbour+rightneighbour)
END

IF (ssol eq 4) THEN $

```

```

BEGIN
  difs = [sdir1-sdir4,sdir2-sdir4,sdir3-sdir4]

  IF (apply_correction_to_weights) THEN $
    leftneighbours = difs*(difs ge 0.0)*(difs le 360) + $
                    (difs+360.0)*(difs lt 0.0) + $
                    (difs-360.0)*(difs gt 360.0) $
  ELSE $
    leftneighbours = difs*(difs ge 0.0) + (360.0+difs)*(difs lt 0.0)

  leftneighbour = min(leftneighbours)
  difs = [sdir4-sdir1,sdir4-sdir2,sdir4-sdir3]

  IF (apply_correction_to_weights) THEN $
    rightneighbours = difs*(difs ge 0.0)*(difs le 360) + $
                     (difs+360.0)*(difs lt 0.0) + $
                     (difs-360.0)*(difs gt 360.0) $
  ELSE $
    rightneighbours = difs*(difs ge 0.0) + (360.0+difs)*(difs lt 0.0)

  rightneighbour = min(rightneighbours)
  sector = 0.5*(leftneighbour+rightneighbour)
END

weight = sector/360.0

  END ; end of un-equal weights case
END ; end of 4 solutions case

return, weight
end

;-----
pro ads_fom_all, FoM, score_r, score_u, score_v, $
    max_nr_of_wvcs, nws, wvcarr, inp_u, inp_v, $
    found_u1, found_v1, found_u2, found_v2, $
    found_u3, found_v3, found_u4, found_v4, $
    dirs, numsol, resolution, implementation

true = 1 eq 1 & false = 1 eq 0

apply_correction_to_implementation_2 = false

expected_SD = (1.0*resolution/50)^(1.0/3.0) * 1.5
; = 1.5 [m/s] for 50 km resolution

```

```

; vars used for storing the distribution of solutions
nrbins = 201
startval = -50.0
stepval = 0.5
binned_u = fltarr(nrbins); -50 to +50, step 0.5
binned_v = fltarr(nrbins); -50 to +50, step 0.5

; used for output of the results
FoM = fltarr(max_nr_of_wvcs)
score_u = fltarr(max_nr_of_wvcs)
score_v = fltarr(max_nr_of_wvcs)
score_r = fltarr(max_nr_of_wvcs)

; repeat the procedure for each satellite node
FOR wvc = 0,max_nr_of_wvcs-1 DO $
BEGIN
  binned_u(*) = 0.
  binned_v(*) = 0.

; loop over all windvectors and produce the distribution
; function for the diff in components for u and v
FOR i=0L,nws-1 DO $
  IF (wvcarr(i) eq wvc) THEN $
  BEGIN

    IF (numsol(i) ge 1) THEN $
    BEGIN
      weight = FoM_calc_weight(numsol(i),1,$
                                dirs(i,0), dirs(i,1), $
                                dirs(i,2), dirs(i,3), $
                                implementation )

      vector_diff_u1 = inp_u(i)-found_u1(i)
      index = FIX((vector_diff_u1-startval)/stepval)
      index = max([0,min([index,nrbins-1]])]
      binned_u(index) = binned_u(index)+weight

      vector_diff_v1 = inp_v(i)-found_v1(i)
      index = FIX((vector_diff_v1-startval)/stepval)
      index = max([0,min([index,nrbins-1]])]
      binned_v(index) = binned_v(index)+weight
    END

  IF (numsol(i) ge 2) THEN $
  BEGIN
    weight = FoM_calc_weight(numsol(i),2,$

```

```

                                dirs(i,0),  dirs(i,1),  $
                                dirs(i,2),  dirs(i,3),  $
                                implementation      )

vector_diff_u2 = inp_u(i)-found_u2(i)
index = FIX((vector_diff_u2-startval)/stepval)
index = max([0,min([index,nrbins-1])])
binned_u(index) = binned_u(index)+weight

vector_diff_v2 = inp_v(i)-found_v2(i)
index = FIX((vector_diff_v2-startval)/stepval)
index = max([0,min([index,nrbins-1])])
binned_v(index) = binned_v(index)+weight
END

IF (numsol(i) ge 3) THEN $
BEGIN
  weight = FoM_calc_weight(numsol(i),3,$
                                dirs(i,0),  dirs(i,1),  $
                                dirs(i,2),  dirs(i,3),  $
                                implementation      )

  vector_diff_u3 = inp_u(i)-found_u3(i)
  index = FIX((vector_diff_u3-startval)/stepval)
  index = max([0,min([index,nrbins-1])])
  binned_u(index) = binned_u(index)+weight

  vector_diff_v3 = inp_v(i)-found_v3(i)
  index = FIX((vector_diff_v3-startval)/stepval)
  index = max([0,min([index,nrbins-1])])
  binned_v(index) = binned_v(index)+weight
END

IF (numsol(i) ge 4) THEN $
BEGIN
  weight = FoM_calc_weight(numsol(i),4,$
                                dirs(i,0),  dirs(i,1),  $
                                dirs(i,2),  dirs(i,3),  $
                                implementation      )

  vector_diff_u4 = inp_u(i)-found_u4(i)
  index = FIX((vector_diff_u4-startval)/stepval)
  index = max([0,min([index,nrbins-1])])
  binned_u(index) = binned_u(index)+weight

  vector_diff_v4 = inp_v(i)-found_v4(i)

```



```

        index = FIX((vector_diff_v4-startval)/stepval)
        index = max([0,min([index,nrbins-1])])
        binned_v(index) = binned_v(index)+weight
    END

END ; end IF (wvcarr(i) eq wvc)

; let the distributions have an integral of 1
IF (total(binned_u) ne 0.0) THEN $
    obs_prob_u = 1.0*binned_u/total(binned_u) $
ELSE  obs_prob_u = 0.0*binned_u

; let the distributions have an integral of 1
IF (total(binned_v) ne 0.0) THEN $
    obs_prob_v = 1.0*binned_v/total(binned_v) $
ELSE obs_prob_v = 0.0*binned_v

; init some new vars
anal_prob_u = 0.*binned_u
anal_prob_v = 0.*binned_v
sum_weighted_mean_squares_u = 0.0
sum_weighted_mean_squares_v = 0.0
sum_anal_prob_u              = 0.0
sum_anal_prob_v              = 0.0

; produce a new (smaller) distribution by multiplying the
; found distribution by a Gaussian background distribution.
exp_factor = -1.0/(2.0*expected_SD^2)
; = -1.0/4.5 if expected_SD = 1.5
; (4.5 = 2*(1.5^2) so halfwidth = 1.5 m/s )

vector_diff = startval + stepval*(indgen(nrbins)+0.5)
backgr_prob = exp(exp_factor*vector_diff^2)

anal_prob_u = backgr_prob * obs_prob_u
anal_prob_v = backgr_prob * obs_prob_v

sum_weighted_mean_squares_u = total(anal_prob_u*vector_diff^2)
sum_weighted_mean_squares_v = total(anal_prob_v*vector_diff^2)

sum_anal_prob_u = total(anal_prob_u)
sum_anal_prob_v = total(anal_prob_v)

; compute the rms from this distribution
IF (sum_anal_prob_u ne 0.0) THEN $
    rms_u = sqrt(sum_weighted_mean_squares_u / sum_anal_prob_u) $

```

```

ELSE rms_u = 0.0

; compute the rms from this distribution
IF (sum_anal_prob_v ne 0.0) THEN $
  rms_v = sqrt(sum_weighted_mean_squares_v / sum_anal_prob_v) $
ELSE rms_v = 0.0

score_u(wvc) = rms_u/expected_SD
score_v(wvc) = rms_v/expected_SD

; Compute a ranking score using Pa
rr = 0.
r1 = 0.
nvectors = 0L
FOR i=0L,nws-1 DO $
  IF (wvcarr(i) eq wvc) THEN $
  BEGIN

  IF (apply_correction_to_implementation_2) THEN $
  BEGIN
    IF (numsol(i) ge 1) THEN $
    BEGIN
      nvectors = nvectors + 1
      vector_diff_u = abs(inp_u(i)-found_u1(i))
      vector_diff_v = abs(inp_v(i)-found_v1(i))
      index_u = FIX((vector_diff_u-startval)/stepval)
      index_v = FIX((vector_diff_v-startval)/stepval)
      index_u = max([0,min([index_u,nrbins-1]])]
      index_v = max([0,min([index_v,nrbins-1]])]
      IF ( (implementation eq 1) OR (implementation eq 2) ) THEN $
        r1 = r1 + obs_prob_u(index_u)*obs_prob_v(index_v)
      IF ( (implementation eq 3) OR (implementation eq 4) ) THEN $
        r1 = r1 + anal_prob_u(index_u)*anal_prob_v(index_v)
    END
  END ELSE $
  BEGIN
    IF (numsol(i) ge 2) THEN $
    BEGIN
      nvectors = nvectors + 1
      vector_diff_u = abs(inp_u(i)-found_u1(i))
      vector_diff_v = abs(inp_v(i)-found_v1(i))
      index_u = FIX((vector_diff_u-startval)/stepval)
      index_v = FIX((vector_diff_v-startval)/stepval)
      index_u = max([0,min([index_u,nrbins-1]])]
      index_v = max([0,min([index_v,nrbins-1]])]
      IF ( (implementation eq 1) OR (implementation eq 2) ) THEN $

```

```

        r1 = r1 + obs_prob_u(index_u)*obs_prob_v(index_v)
    IF ( (implementation eq 3) OR (implementation eq 4) ) THEN $
        r1 = r1 + anal_prob_u(index_u)*anal_prob_v(index_v)
    END
END

IF (numsol(i) ge 2) THEN $
BEGIN
    vector_diff_u = abs(inp_u(i)-found_u2(i))
    vector_diff_v = abs(inp_v(i)-found_v2(i))
    index_u = FIX((vector_diff_u-startval)/stepval)
    index_v = FIX((vector_diff_v-startval)/stepval)
    index_u = max([0,min([index_u,nrbins-1])])
    index_v = max([0,min([index_v,nrbins-1])])
    IF ( (implementation eq 1) OR (implementation eq 2) ) THEN $
        rr = rr + obs_prob_u(index_u)*obs_prob_v(index_v)
    IF ( (implementation eq 3) OR (implementation eq 4) ) THEN $
        rr = rr + anal_prob_u(index_u)*anal_prob_v(index_v)
    END
END

IF (numsol(i) ge 3) THEN $
BEGIN
    vector_diff_u = abs(inp_u(i)-found_u3(i))
    vector_diff_v = abs(inp_v(i)-found_v3(i))
    index_u = FIX((vector_diff_u-startval)/stepval)
    index_v = FIX((vector_diff_v-startval)/stepval)
    index_u = max([0,min([index_u,nrbins-1])])
    index_v = max([0,min([index_v,nrbins-1])])
    IF ( (implementation eq 1) OR (implementation eq 2) ) THEN $
        rr = rr + obs_prob_u(index_u)*obs_prob_v(index_v)
    IF ( (implementation eq 3) OR (implementation eq 4) ) THEN $
        rr = rr + anal_prob_u(index_u)*anal_prob_v(index_v)
    END
END

IF (numsol(i) ge 4) THEN $
BEGIN
    vector_diff_u = abs(inp_u(i)-found_u4(i))
    vector_diff_v = abs(inp_v(i)-found_v4(i))
    index_u = FIX((vector_diff_u-startval)/stepval)
    index_v = FIX((vector_diff_v-startval)/stepval)
    index_u = max([0,min([index_u,nrbins-1])])
    index_v = max([0,min([index_v,nrbins-1])])
    IF ( (implementation eq 1) OR (implementation eq 2) ) THEN $
        rr = rr + obs_prob_u(index_u)*obs_prob_v(index_v)
    IF ( (implementation eq 3) OR (implementation eq 4) ) THEN $
        rr = rr + anal_prob_u(index_u)*anal_prob_v(index_v)

```

```
END

END ; end IF (wvcarr(i) eq wvc)

IF (r1+rr ne 0.0) THEN score_r(wvc) = (2*rr) / (r1 + rr) $
      ELSE score_r(wvc) = 0.0

FoM(wvc) = 0.2*score_r(wvc) + 0.4*score_u(wvc) + 0.4*score_v(wvc)

END ; loop over wvc

END ; end of this subroutine
```