Prepared for:

RWS RIKZ (Rijkswaterstaat - Rijksinstituut voor Kust en Zee)

User Manual Calibration Instrument SWAN (Version build 122/08.01.30)

version 1.0

Report

January 2008

WL | delft hydraulics

H4918.67

Prepared for:

RWS RIKZ (Rijkswaterstaat - Rijksinstituut voor Kust en Zee)

User Manual Calibration Instrument SWAN (Version build 122/08.01.30)

version 1.0

Ivo Wenneker, Juzer Dhondia, Stef Hummel, Martin Verlaan, Herman Gerritsen

Report

January 2008

WL | delft hydraulics

Client:	RWS RIKZ (Rijkswaterstaat - Rijksinstituut voor Kust en Zee)
Title:	User Manual Calibration Instrument SWAN (Version build 122/08.01.30)

Abstract:

This document is a concise User Manual for the Calibration Instrument SWAN (C-I), version build 122 d.d. 08 01 30. It is intended for the experienced SWAN user who wants to calibrate a SWAN case, i.e. a SWAN model for a given application. It describes the steps that a user needs to take in order to perform a calibration for a SWAN case.

The implementation of the Calibration Instrument has been shown to work on Linux and Windows platforms. As calibration methods, the optimisation techniques DUD, Simplex and Powell have been implemented. The Beji-Battjes case was used as primary test case for the software tests and the interpretation of the results. It is also used to illustrate the practical application of the calibration instrument in the User Manual. For all three methods, experiments have been performed , both without and including constraints on the allowed variation of the parameters from the initial guess during the optimisation process. A Beji-Battjes twin experiment was performed successfully as well. For the more practically relevant application Friesche Zeegat, calibration experiments were initialised to investigate whether the system initialises properly, starts the optimisation correctly and starts to converge. This is the case for all three calibration methods. Full optimisation results for the Friesche Zeegat case will be reported in the next phase of the present project.

The results of the test experiments are reported in Appendix B.
Appendix C provides a short description of the Beji-Battjes test case and the Friesche Zeegat case.
Appendix D provides a list of tests and additional functionalities which will be realised in the next phase of the SBW Waddenzee project (2008).

Finally, Appendix E provides the external reviewer's report on the present software and User Manual. The follow up on the review comments and recommendations, in as far as they have not yet be taken into account in the present document or software, will be part of the next phase of the project. A new software release and new version of the User Manual is foreseen as part of the next project.

References:	H4918; SBW Activity 8.4 – Realisation Calibration Instrument dated 31 July 2007

Ver	Author		Date	Remarks	Review		Approved by	
0.3	Wenneker,Dhondia,		19 December 2007	draft	H. Gerritsen		M. van Gent	
	Hummel,Verlaan.							
	Gerritsen							
1.0	Idem		31 January 2008	final	G.Ph.van Vledder		W. Tilmans	

Project number:	H4918.67
Keywords:	automated model calibration, parameter estimation techniques, Simplex, Powell, DUD, SWAN
Number of pages:	46
Classification:	None
Status:	Final

Contents

Appendices

1 Introduction

This document contains a concise User Manual of the Calibration Instrument (C-I), version 1.0, version build 122 d.d. 08 01 30. It is intended for the experienced SWAN user who wants to calibrate a *SWAN case*, i.e. a SWAN model for a given application. It is not intended to give the user more information than strictly necessary on calibration techniques itself; we refer to WL (2006) and WL (2007a) for more on this. It is also not intended to give the user more information than strictly necessary on the software aspects; we refer to WL (2007b,c,d) for more on this, nor on the SWAN model for which we refer to the SWAN manual (DUT, 2007).

In Chapter 2, some essentials on SWAN calibration are given. The installation of the C-I and the set-up, execution and outcomes of the calibration for a SWAN case are discussed in Chapter 3.
In Appendix A, conventions on SWAN file names and file formats are given. An overview of the test results for calibration of the Beji Battjes test case plus a first conclusion on the software testing for the Friesche Zeegat case are given in Appendix B. A background description of the two SWAN cases is given in Appendix C. Appendix D presents a list of new functionalities and further test activities which are the subject of the next phase of the present project, foreseen to be completed by July 2008.

Finally, Appendix E presents the reviewer report of the present software and User Manual, prepared by external reviewer dr. Gerbrant Ph. van Vledder of Alkyon.

2 Calibration of SWAN – some essentials

This chapter provides some essential aspects concerning SWAN calibration with the C-I.

In the C-I, three semi-automated parameter estimation methods – DUD, Powell, Simplex – have been implemented. By structured variation of the SWAN model parameters that need to be optimised, a so-called Goodness-of-Fit (GoF) – see Section 2.1 – is minimised. The parameter values that correspond to the minimum value of the GoF are considered the optimum parameter values for the given optimisation experiment and the selected optimisation method. The SWAN model parameters are discussed in Section 2.2. The C-I works only properly if certain conventions related to the SWAN file names and the SWAN file formats are adhered to. These conventions are discussed in Section 2.3.

2.1 Goodness-of-Fit (GoF) or cost function

In the C-I, the so-called Goodness-of-Fit (GoF) or cost function is minimised under certain constraints (selected method, criteria for stopping further optimisation, constraints on allowed variation of parameters from the initial estimate, etc.). For each set of parameter values during parameter variation, the GoF provides a measure indicating how close the SWAN model results for that set are to observational data.

2.1.1 General form of the Goodness-of-Fit criterion

Based on general practice and availability of reliable data sets, the GoF will in principle be based on a formulation in terms of the integral parameters significant wave height (H_{m0}), (smoothed) peak period (T_p) and mean wave direction $\overline{\theta}$ (all with weight terms), but could also provide information on the spectral form. Information on the low and high end of the spectrum are represented by the spectral wave periods $T_{m-1,0}$ and T_{m02}. As an example of a GoF or least squares criterion, we refer to Alkyon (2003). This formula could be adapted for the present case to read:

$$
\begin{aligned}
GoF = {} & \frac{1}{2}\frac{\sum_{i=1}^{N_i}\left[w^i_{H_{m0}}\left(H^i_{m0,obs}-H^i_{m0,sim}\right)/\sigma_{H^i_{m0,obs}}\right]^2}{\sum_{i=1}^{N_i} {w^i_{H_{m0}}}^2} + \\
& + \frac{1}{2}\frac{\sum_{i=1}^{N_i}\left[w^i_{T_{m-1,0}}\left(T^i_{m-1,0,obs}-T^i_{m-1,0,sim}\right)/\sigma_{T^i_{m-1,0,obs}}\right]^2}{\sum_{i=1}^{N_i} {w^i_{T_{m-1,0}}}^2} + \frac{1}{2}\frac{\sum_{i=1}^{N_i}\left[w^i_{T_p}\left(T^i_{p,obs}-T^i_{p,sim}\right)/\sigma_{T^i_{p,obs}}\right]^2}{\sum_{i=1}^{N_i} {w^i_{T_p}}^2} + \\
& + \frac{1}{2}\frac{\sum_{i=1}^{N_i}\left[w^i_{T_{m02}}\left(T^i_{m02,obs}-T^i_{m02,sim}\right)/\sigma_{T^i_{m02,obs}}\right]^2}{\sum_{i=1}^{N_i} {w^i_{T_{m02}}}^2} + \frac{1}{2}\frac{\sum_{i=1}^{N_i}\left[w^i_{\overline{\theta}}\left(\overline{\theta}^i_{obs}-\overline{\theta}^i_{sim}\right)/\sigma_{\overline{\theta}^i_{obs}}\right]^2}{\sum_{i=1}^{N_i} {w^i_{\overline{\theta}}}^2},
\end{aligned}
$$

in which N_i is the number of spatial calibration data points (possibly summed over a number of applications), w_{φ}^{i} is a weight function (for the present moment taken to be equal to 1), and $\sigma^{i}_{\varphi_{obs}}$ the standard deviation of the observed parameter in the particular location, accounting for the uncertainty or quality of the respective observations. An appropriate directional measure would be the angle from the mean wave direction $\overline{\theta}$ (ensuring proper accounting for directional periodicity, e.g. using the formula $\Delta\theta$=180-|180-|θ_1-θ_2||, but losing its direction: clockwise or counter clockwise).

In the present implementation of the C-I, the factor ½, which is needed for reasons of Maximum Likelihood theory and the estimation of uncertainty measures in the parameters, is still absent. This does not affect the calibration process or the best estimates found. (the factor ½ will be added in the next software release, when calculation of the uncertainty bands around the parameters will be added). The user has the freedom to choose which quantities are included in the calibration process. These may also be other quantities than the ones just mentioned, such as directional spreading. The only – obvious – restriction is that observational data of this quantity should be available, and SWAN should be able to output this quantity.
The factor 1/2 in the GoF will be added in a later version of the C-I; its absence has no influence on the calibration process.

2.1.2 Formulation of (soft) constraints on parameter ranges

During a calibration cycle in which a number of parameters are jointly optimised, individual parameters may gradually drift away from the range that is considered acceptable from a physical point of view. The variation of individual parameters can be constrained by adding a so-called soft constraint term (SC) to the GoF in the previous section. This SC term has the form:

$$SC = +\frac{1}{2}\sum_{p=1}^{P} w_p \left(\frac{\alpha_p - \alpha_p^{ref}}{\sigma_p^{ref}}\right)^2,$$

in which α_p is a parameter that is being calibrated, α_p^{ref} is an initial best guess for α_p, and σ_p^{ref} is a measure for the variation that is allowed in α_p^{ref} ; w_p is a weight factor with value 0 or 1, to indicate whether a constraint for parameter α_p is applied. Furthermore, p = 1, …., P is the index for the parameter considered.

2.2 Overview of model parameters

The following table presents an overview of the model parameters that may be taken into account in the SWAN model calibration. The table also includes input parameters (representing boundary forcing or volume forcing in the model) that may be included in the calibration process. The C-I poses no limits on the number of model parameters, so the list of model parameters may be extended in the future.

The subsequent columns in Table 2.1 contain the parameterisations of the physical process involved, the symbol of the parameter, its acronym in the SWAN command file, its default value in SWAN version 40.51A, and values indicating the uncertainty in the parameters. For a detailed description of the nature of the parameters, see the SWAN User Manual (DUT, 2007) and WL (2006). Distributions are used to express the uncertainty of the model parameters. The values for the standard deviation in case of normal distribution are taken from Table 4 of WL (2007e). In the C-I, the use of a uniform distribution for the uncertainty of the parameters is not possible. For these parameters, we have obtained the standard deviation by applying the rule that the confidence range spans the default value plus and minus twice the standard deviation. For some other parameters, no information on the uncertainty has been given in WL (2007e) at all. For these, we have come up with uncertainty values ourselves:

- In WL (2007e), the uncertainties in the whitecapping parameters are given for the saturation-based whitecapping formulation by Van der Westhuysen et al. (2007), whereas in the present report the default whitecapping formulation by Komen et al. (1994) is used. In the present report, we assume that the uncertainty in the parameter $C_{ds,wc}$ for the Komen formulation satisfies a normal distribution and has, percent wise, the same amount of uncertainty as the parameter $C_{ds,wc}$ for the Van der Westhuysen formulation.
- For parameter q, a value equal to 2 is advised in virtually all cases. We suggest to put the standard deviation for this parameter equal to a small value, e.g. 0.2 (viz., 10%).
- For parameter r, values equal to 0 (SWAN default value) and 1 (for the computation of the Hydraulic Boundary Conditions in 2006) have been used in the past. This suggests a standard deviation for the uncertainty of 0.5, with a base value equal to 0.5. However, putting a negative value for r is physically not realistic, therefore we leave the issue of uncertainty open for this parameter. The user is advised to use his best knowledge and try several values to see their influence of the parameter optimisation process.
- For parameter $f_{\max,EB}$, no uncertainty values are given in WL (2007e). The default value for $f_{\max,EB}$ has changed from 2.2 (SWAN version 40.41) and 5.0 (SWAN version 40.51) to 2.5 (SWAN version 40.51A). This gives an indication for the range of uncertainty in this parameter. For the moment, we taken the standard deviation in the uncertainty for this parameter equal to 1.0.

We note that the uncertainty in the parameters nor their distribution are not well-known. The problem is even deeper. Assigning uncertainties to the parameters of these formulations suggests that the formulations are known (conceptual model), whereas in reality large uncertainties exist about the formulations itself. Therefore, the uncertainty values in Table 2.1 can only be regarded as guidelines on the basis of expert opinion.

Table 2.1 SWAN model parameters

Phys. process	**Symbol**	**Acronym**	**Default value 40.51A**	**Standard devation**
Wave growth	α_{in}	---	---	---
Whitecapping	$C_{ds,wc}$	cds2	2.36E-5	0.17E-5

	q	powst	2	0.2
	r	powk	0	---
Quadruplets	C_{nl4}	Cnl4	3.0E7	0.25E7
	λ_{nl4}	lambda	0.25	0.025
Bottom friction	$C_{ds,bot}$	cfjon	0.067	0.015
Depth-induced breaking	α_{BJ}	alpha	1.0	0
	γ_{BJ}	gamma	0.73	0.15
Triads	α_{EB}	trfac	0.05	0.0125
	$f_{\max,EB}$	cutfr	2.5	1.0
Boundary condition	α_{DWS}	Adws	1.0	---
Wind speed	$\alpha_{windspeed}$	Awspeed	1.0	---
Wind direction	$a_{winddir}$	Awdir	0.0	---

In the above table, α_{DWS} stands for the uncertainty factor with which the spectrum (measured or assumed uniform) that is prescribed at the open boundary will be multiplied. Furthermore, $\alpha_{windspeed}$ stands for the uncertainty factor in the wind speed, and $a_{winddir}$ stands for the uncertainty in wind direction, expressed in radians or degrees.

Remark 1: It is noted that not all parameters need to be selected for a calibration cycle; in general, a sub-set will be selected, based on prior knowledge on behaviour and relevance for the SWAN case under study. This subset will be called *SWAN case parameters*. In other words, the user needs to decide which processes are included in the calibration cycle and which ones not. The input parameters α_{DWS}, $\alpha_{windspeed}$ and $a_{winddir}$ are not taken into account initially, as we generally assume that these are sufficiently accurately known. If, however, the analysis of the simulation results shows internal inconsistencies which, based on physical reasoning can be related to the model inputs, parameters α_{DWS}, $\alpha_{windspeed}$ and $a_{winddir}$ can be included in a further cycle of the calibration.

Remark 2: Parameter α_{in} cannot be calibrated via the standard SWAN command file. Parameter α_{DWS} cannot be calibrated via the SWAN command file when a non-parametric spectrum such as a measured spectrum is applied at the boundary. Parameters $\alpha_{windspeed}$ and $a_{winddir}$ cannot be calibrated via the SWAN command file in case of a non-uniform wind field.

Remark 3: Parameter α_{BJ} must be kept fixed and equal to 1.0, see the arguments in WL (2006).

Remark 4: Note that we do not include bathymetric uncertainties in the calibration. If there is uncertainty about the bathymetry, this often is formulated in terms of a migration of a channel or similar systematic effect. In such cases, one should consider simulations with an appropriately adjusted bathymetry as such. The same holds for the water level.

2.3 Conventions on nomenclature and file formats

The C-I adheres to the conventions on nomenclature and file formats as used for SHiVa (SWAN Hindcast and Validation instrument). These conventions are discussed in detail in Chapter 4 of Xi (2007b), which is included in Appendix A.

Adherence to these conventions requires some actions from the user. There are three situations possible:

- The user downloads the SWAN case from the SHiVa website; see Section 2.3.1.
- The user creates the SWAN case by him/herself; see Section 2.3.2.
- The SWAN case is already included in an operational version of the C-I. This is the present situation, with SWAN case l21triad. In this situation, no additional action with respect to conventions on nomenclature and file formats is required by the user.

Note: The C-I software includes the SWAN executable (at present: version 40.51A). A dedicated SWAN wrapping facilitates the easy exchange of SWAN model parameter settings and SWAN model results between SWAN and the C-I.

2.3.1 Settings of model parameters – SWAN case is downloaded from SHiVa server

As long as the SWAN case, i.e. the test case which is calibrated, is downloaded from the SHiVa website, the SHiVa conventions on filenames are automatically adhered to. The only thing the user needs to do is to make some adaptations to the SWAN command file.

The SWAN command file, with extension .swn, is the SWAN steering file. It is an ASCII file. It contains, among others, the model parameters mentioned in Table 2.1. The adopted convention in SHiVa is that the model parameters and some related parameters are stated in the SWAN command file as follows:

```
$ --- Begin of settings physical process parameters
$
GEN3
<#WCAPON#>WCAP KOM cds2=<#cds2#> stpm=<#stpm#> powst=<#powst#> delta=<#delta#> powk=<#powk#>
<#WCAPOFF#>OFF WCAP
<#QUADON#>QUAD iquad=<#iquad#> lambda=<#lambda#> Cnl4=<#Cnl4#>
<#QUADON#>LIMITER ursell=<#ursell#> qb=<#qb#>
<#QUADOFF#>OFF QUAD
<#FRICON#>FRIC JONSWAP cfjon=<#cfjon#>
<#FRICOFF#>$
<#BREAON#>BREA CON alpha=<#alpha#> gamma=<#gamma#>
<#BREAOFF#>OFF BREA
<#TRIADON#>TRIAD trfac=<#trfac#> cutfr=<#cutfr#>
<#TRIADOFF#>$
$
$ --- End of settings physical process parameters
```

There are various so-called placement holder strings <#..#> to steer the calibration process automatically. Their syntax and usage will be explained in the SHiVa manual, see for example Appendix A. In addition to specifying parameter settings, physical processes can be activated or deactivated using the SWAN command OFF. The strings <#WCAPON#> and <#WCAPOFF#> refer to whitecapping on respectively off; the strings <#QUADON#> and <#QUADOFF#> refer to quadruplets on respectively off; the strings <#FRICON#> and <#FRICOFF#> refer to bottom friction on respectively off; the strings <#BREAON#> and <#BREAOFF#> refer to depth-induced breaking on respectively off, and the strings <#TRIADON#> and <#TRIADOFF#> refer to triads on respectively off. The meaning of the other placement holders will become clear further on.

In the above given part of the SWAN command file, the user needs to make certain adjustments manually. This will be illustrated for the SWAN case l021triad001 (l21triad in the ONR Testbed).

Step 1. Include / exclude physical processes

The user needs to decide which physical processes are to be included in / excluded from the model calibration. This decision may be based on information in the xml-file in directory model_io, which comes with the SWAN case after download from the SHiVa website. This information is based on earlier model runs, and may be regarded as a good starting point. However, the user may also deviate from this. This implies that the user has all freedom in inclusion / exclusion of physical processes in the model calibration.

Sticking to the information in the xml-directory, this leads for the SWAN case l21triad to an inclusion of the processes whitecapping, bottom friction, breaking and triads, and an exclusion of quadruplets. The user needs to remove manually the placement holders strings that refer to statements that are ‘true’, and he needs to remove the entire lines that refer to ‘false’ statements. For the SWAN case l21triad, this leads to:

```
$ --- Begin of settings physical process parameters
$
GEN3
WCAP KOM cds2=<#cds2#> stpm=<#stpm#> powst=<#powst#> delta=<#delta#> powk=<#powk#>
OFF QUAD
FRIC JONSWAP cfjon=<#cfjon#>
BREA CON alpha=<#alpha#> gamma=<#gamma#>
TRIAD trfac=<#trfac#> cutfr=<#cutfr#>
$
$ --- End of settings physical process parameters
```

Step 2. Insert numerical values for parameters that are not calibrated

After step 1, the user needs to insert numerical values for the parameters that need not be calibrated. This is done in the SWAN input file or in the parameters.xda file (see Chapter 3 below). This set of parameters can be divided into the following groups:

- The group of parameters that do not belong to the model parameters mentioned in Table 2.1.
- The group of parameters that the user, based on knowledge of the physical processes involved in the SWAN case, wants to exclude from model calibration.

The first group consists of parameters that are present in the SWAN command file, but that may not be taken into account in model calibration. These parameters are not mentioned in Table 2.1. In Table 2.2, the variable names as appear in the SWAN command file and their default value (SWAN version 40.51A) are given. The user may choose other values than the default values. We have included variable alpha (α_{BJ}) in this table, since this parameter is kept fixed, see also WL (2006). All parameters in Table 2.2 must be kept fixed, since they are not included as calibration parameters in the calibration process.

Table 2.2 Name and default value of parameters that are not included in the calibration process.

Variable name	Default value
delta	0.0
stpm	3.02E-3
iquad	2
ursell	10
qb	1
alpha	1

The second group consists of parameters that the user, based on knowledge of the physical processes involved in the SWAN case, wants to exclude from model calibration. Based on a priori knowledge of the SWAN case (e.g. obtained from a sensitivity analysis), it may be expected that certain physical processes will play only a minor role in the considered SWAN case. It may then be wise, for example to save computing resources, to exclude these processes from the calibration cycle. This is realized by inserting numerical values for the parameters pertaining to these processes. For example, for a shallow water case like l21triad the process of whitecapping does not have a large influence, so that changes in the parameters pertaining to whitecapping do not lead to significantly different results. Based on the type of application (deep water, shallow water), a parameter categorization is given in Section 5.1 of WL (2006) from which the ‘application-dependent’ and ‘dust-bin’ physical processes can be deduced. Leaving out a certain parameter will speed up the calibration process because it reduces the dimension of optimization problem.

We note that the user has also the option to fix a subset of parameters pertaining to the dominating physical processes. For example, the user may choose to fix the value for parameter cutfr in the triad-interaction process, while leaving parameter trfac as a calibration parameter. Fixing a parameter can also be realized by setting its range to zero, but this requires unwanted CPU. In the next release, an error message will be generated in case the range is set to zero, to avoid accidental errors. The user should then explicitly exclude the parameter from the parameter set that needs to be estimated.

Returning to our example l21triad, assuming that we are excluding the whitecapping process from model calibration and that we do not fix any other parameter, we arrive at:

```
$ --- Begin of settings physical process parameters
$
GEN3
WCAP KOM cds2=2.36E-5 stpm=3.02E-3 powst=2 delta=0.0 powk=0
OFF QUAD
FRIC JONSWAP cfjon=<#cfjon#>
BREA CON alpha=1.0 gamma=<#gamma#>
TRIAD trfac=<#trfac#> cutfr=<#cutfr#>
$
$ --- End of settings physical process parameters
```

In the present version, the placement holders need to be replaced by strings of the form ‘PAR_*NAME*’, with *NAME* the name of the parameter. For the example given above, this leads to:

```
$ --- Begin of settings physical process parameters
$
GEN3
WCAP KOM cds2=2.36E-5 stpm=3.02E-3 powst=2 delta=0.0 powk=0
OFF QUAD
FRIC JONSWAP cfjon=PAR_CFJON
BREA CON alpha=1.0 gamma=PAR_GAMMA
TRIAD trfac=PAR_TRFAC cutfr=PAR_CUTFR
$
$ --- End of settings physical process parameters
```

Concluding remark

In the SWAN command file, for the SWAN parameters that will be calibrated, placement holders / PAR_*NAME* strings remain (four in the example given above). The calibration process aims at optimizing the parameters pertaining to these variables. The calibration algorithm selects suitable initial numerical values for these parameters, and the SWAN wrapper replaces the placement holders with these values before SWAN execution takes place.

2.3.2 Settings of physical parameters – SWAN case created by the user

In the situation that the user wants to calibrate a SWAN case that is created by him/herself, then the following must be taken into account:

- The SHiVa conventions on nomenclature and file formats must be adhered to, see Appendix A.
- All SWAN case parameters (the model parameters that need to be calibrated) must be included in the SWAN command file by means of placement holders / PAR_*NAME* strings. See Section 2.3.1 for this.

3 Installation of C-I, set-up, execution and outcomes of calibration process

3.1 Introductory remarks

In this chapter we discuss installation of the calibration instrument, set-up, execution and outcomes of the calibration process for a SWAN case. In the present section, some introductory remarks are made.

- This chapter is written for the experienced SWAN user who wants to calibrate a SWAN case, which is a SWAN model for a given application. It is not intended to give the user more information than strictly necessary on the software aspects of the C-I or on SWAN.
- We have made several choices, e.g. concerning directory structure and file names, to make this chapter more readable. These choices pose no restrictions on the use of the C-I, so we recommend that the user adheres to our choices. The C-I offers the user the flexibility to deviate from our choices.
- Only relative path names ought to be used, since this facilitates the use of the C-I considerably.

The set-up of the calibration process for a particular SWAN case will be discussed by means of the example l21triad, which was already introduced in the previous chapter. Setting up the calibration process for other SWAN cases goes in a similar fashion.

3.2 Installation

Installation of the C-I goes simply by copying the C-I from the CD-ROM directory to the user’s local system. The directory structure must be kept the same. Since only relative path names are used in the C-I, there are no restrictions on the location of the root directory of the C-I.

The entire directory structure of the C-I, including the most important files for the user, is shown in Figure 3.1. This is discussed in more detail below.

3.3 Description of main directory structure

In [C-I root], the root directory of the C-I, four directories are present:

- \bin
- \jre
- \model (*advised alternative name*: \tests)
- \xmlschemas

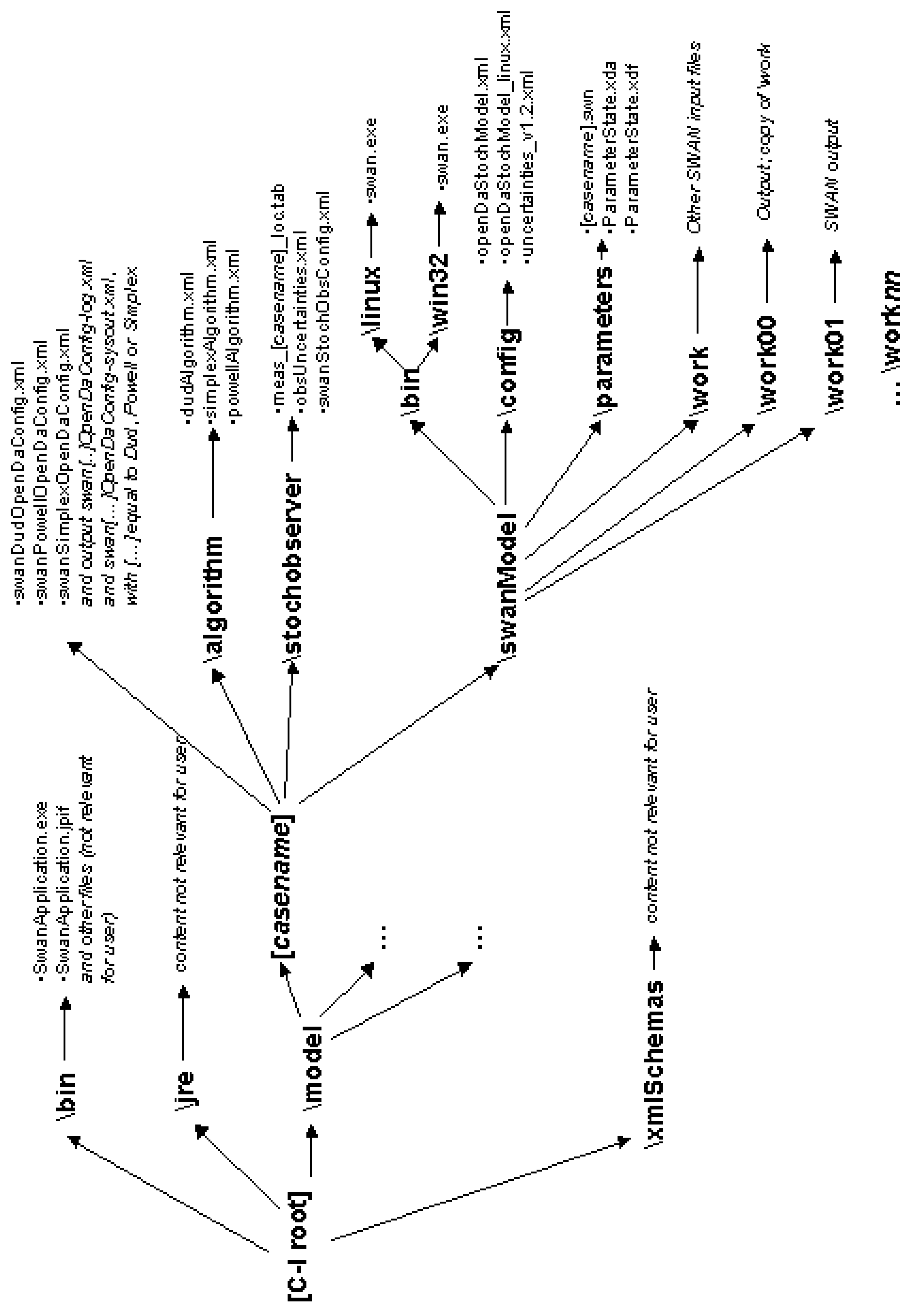


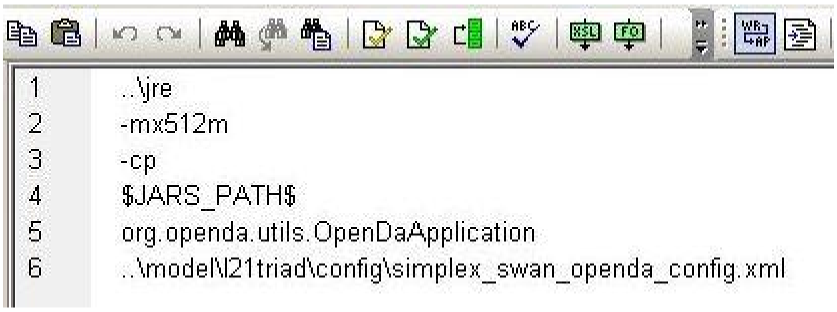
Figure 3.1 Directory structure of C-I, including the most important files for the user.

3.3.1 Directory \bin

In the directory \bin, the executables of the C-I are stored. *The user is advised to check this in detail when installing.* To facilitate the execution of the C-I, the following steps need to be performed (if not done already):

- Copy the file Application.exe to a file named SwanApplication.exe.

- Copy the file Application.jpif to a file named SwanApplication.jpif, and adapt its content at two places: line 1 is changed, and line 6 is added. The change in line 1 concerns the reference to the java run file environment. Line 6 refers to the Xml-file in the SWAN case directory that contains the configuration of the SWAN case. From its name, the selected calibration algorithm can be deduced. In other words, this is where the user need to select the SWAN case and the calibration algorithm. In the example as shown in Figure 3.2, the SWAN case is l21triad, and the selected calibration algorithm is Simplex.



```
1    ..\jre
2    -mx512m
3    -cp
4    $JARS_PATH$
5    org.openda.utils.OpenDaApplication
6    ..\model\l21triad\config\simplex_swan_openda_config.xml
```

Figure 3.2 Content of SwanApplication.jpif as shown in the C-I tool

3.3.2 Directory \jre

This directory contains the Java runtime environment. There is no need for the user to go into this directory.

3.3.3 Directory \model (*advised name:* \tests)

In this directory (preferred name: \tests), the user needs to create subdirectories: one per SWAN case. This is discussed in Section 3.4.

3.3.4 Directory \xmlSchemas

This directory contains xml-files[1] that validate the user-made xml files. There is no need for the user to go into this directory.

3.4 Set-up calibration process for a SWAN case

All information related to a particular SWAN case is stored in the directory [C-I root]\model*casename*, where [C-I root] is the root directory of the C-I and *casename* refers to the name of the case. In our example, *casename* is l21triad. In this directory, there are three subdirectories:

- \algorithm

[1] xml-files can be read and written with xml-editors like Altova XML spy, but also with simple ASCII text editors like Textpad or Notepad

- \stochobserver
- \swanModel

In each of these directories, the user needs to perform the actions that are described below.

In the directory [C-I root]\model*casename*, there are also three configuration files:

- swanDudOpenDaConfig.xml – calibration algorithm is SuperDUD;
- swanPowellOpenDaConfig.xml – calibration algorithm is Powell;
- swanSimplexOpenDaConfig.xml – calibration algorithm is Simplex.

These files contain the configuration of the stochastic observer, the SWAN model and the calibration algorithm. There is no need for the user to go in these files.

After execution of the SWAN calibration, files are created and stored in directory [C-I root]\model*casename*. Their content is discussed in Section 3.6.

Remark

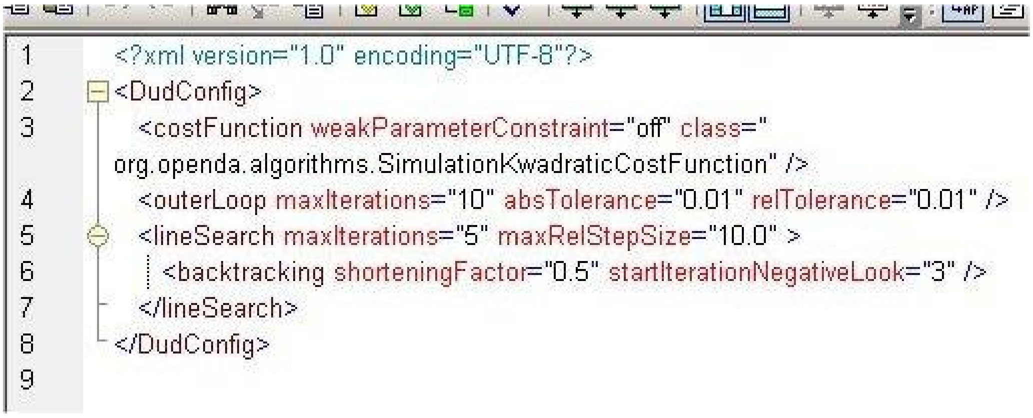
According to us, a good working practice for setting up a new SWAN calibration case is to copy the entire directory [C-I root]\model*casename* to a new directory called [C-I root]\model*casenameOther*. Here, *casename* is the existing SWAN calibration case (here: l21triad), and *casenameOther* is the new SWAN calibration case. In this way existing files needed for the calibration can be re-used, and the user does not need to delve into the detailed syntax of these files. Hereafter, the user has to make various adjustments in the files contained in the directory [C-I root]\model*casenameOther*, as is described below.

3.4.1 Subdirectory \algorithm

This directory contains the configuration files of the calibration algorithms:

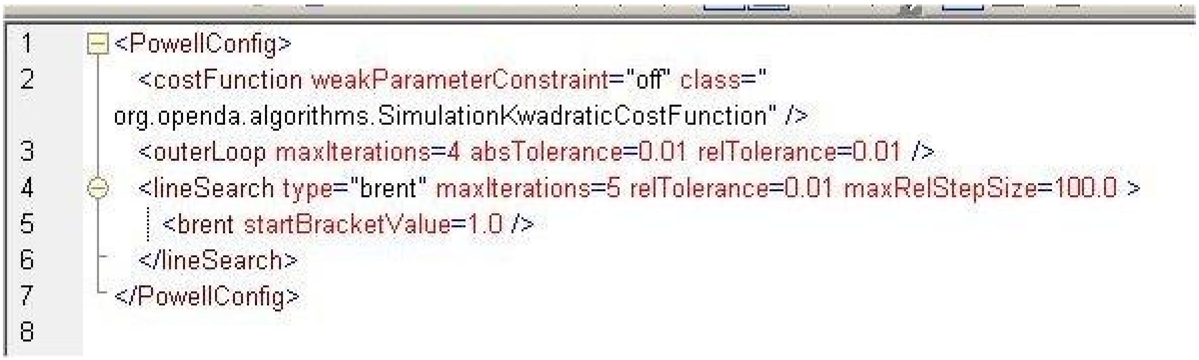
- dudAlgorithm.xml
- powellAlgorithm.xml
- simplexAlgorithm.xml

The first part of the file names correspond to the calibration algorithm involved: DUD, Powell and Simplex. In these files, the user can specify the settings for the calibration algorithms. The content of these files, as used during the tests described further on in this report, is included below.



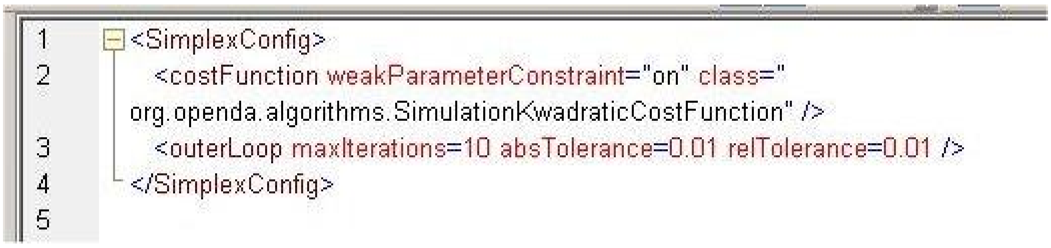
```
<?xml version="1.0" encoding="UTF-8"?>
<DudConfig>
  <costFunction weakParameterConstraint="off" class="
org.openda.algorithms.SimulationKwadraticCostFunction" />
  <outerLoop maxIterations="10" absTolerance="0.01" relTolerance="0.01" />
  <lineSearch maxIterations="5" maxRelStepSize="10.0" >
    <backtracking shorteningFactor="0.5" startIterationNegativeLook="3" />
  </lineSearch>
</DudConfig>
```

Figure 3.3 Content of the file dudAlgorithm.xml



```
<PowellConfig>
  <costFunction weakParameterConstraint="off" class="
org.openda.algorithms.SimulationKwadraticCostFunction" />
  <outerLoop maxIterations=4 absTolerance=0.01 relTolerance=0.01 />
  <lineSearch type="brent" maxIterations=5 relTolerance=0.01 maxRelStepSize=100.0 >
    <brent startBracketValue=1.0 />
  </lineSearch>
</PowellConfig>
```

Figure 3.4 Content of the file powellAlgorithm.xml



```
<SimplexConfig>
  <costFunction weakParameterConstraint="on" class="
org.openda.algorithms.SimulationKwadraticCostFunction" />
  <outerLoop maxIterations=10 absTolerance=0.01 relTolerance=0.01 />
</SimplexConfig>
```

Figure 3.5 Content of the file simplexAlgorithm.xml

3.4.2 Subdirectory \stochobserver

This directory contains the observational data and the uncertainties associated with them. The software associated with this is called the Stochastic Observer. The following three files are present:

- meas_*casename*_loc.tab, where *casename* is the name of the SWAN case (l21triad in our example)
- obsUncertainties.xml
- swanStochObsConfig.xml

The file meas_*casename*_loc.tab contains the observational data. Its file format follows the SHiVa conventions, i.e. is the SWAN table format. See also Appendix A. In particular, we note the following:

- The first two columns must contain the coordinates Xp and Yp of the measurement location.
- The remaining columns must contain the quantities against which calibration is performed.

Concerning the quantities against which calibration is performed, the following remarks are made:

- The user needs to decide which are the quantities against which calibration is performed. Only these quantities must be included in the file meas_*casename*_loc.tab; the others ones should be removed manually. This requires user insight in the relevant processes. Typical quantities are the significant wave height (H_{m0}), peak period (T_p), mean wave direction $\overline{\theta}$ and spectral wave periods $T_{m-1,0}$ and T_{m02}. See also Section 2.1.
- It may, as stated above, be the case that calibration is performed against only a part of the available observed quantities. Then it is handy to make a copy of the original meas_*casename*_loc.tab file (for example, under the name meas_*casename*_loc_alldata.tab), and to remove the not considered columns from the meas_*casename*_loc.tab file.
- Directional periodicity is not properly accounted for yet. This means that in the present version the mean wave direction should be excluded from the quantities against which calibration is performed. In the next release, this will be properly accounted for.
- The names of the quantities in the file meas_*casename*_loc.tab must be identical to the names in the SWAN output. So: Hsig for H_{m0}, TPsmoo for T_p, Dir for $\overline{\theta}$, Tm_10 for $T_{m-1,0}$, Tm02 for T_{m02}, and so on.

The file obsUncertainties.xml contains the uncertainties associated with all observational data points in the file meas_*casename*_loc.tab. We recall that the uncertainties in the file obsUncertainties.xml are related to the variable $\sigma^{i}_{\varphi_{obs}}$, which appears in the definition of the GoF (see Section 2.1). A normal probability distribution function (pdf) is assumed for the uncertainty. The standard deviation should be defined in absolute or relative values:

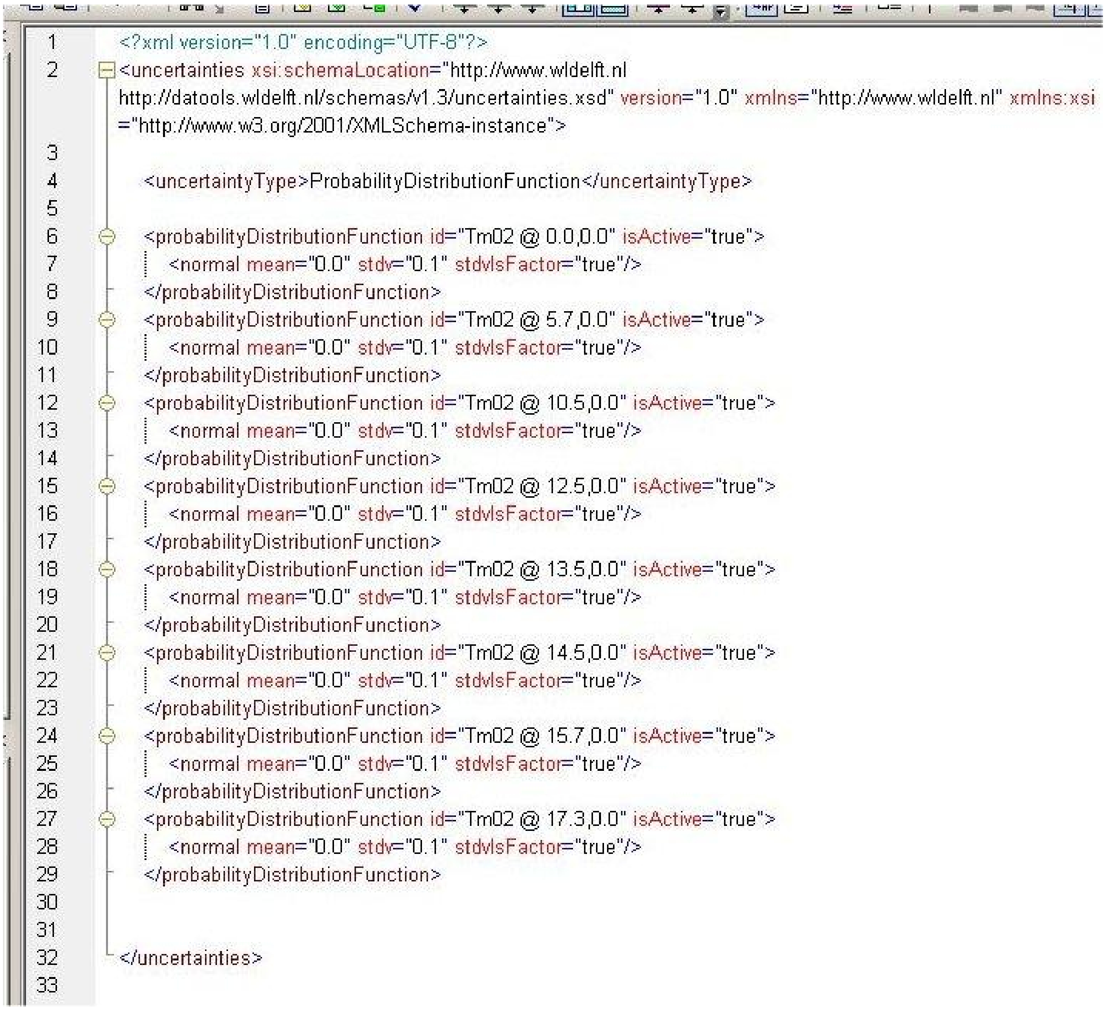
- ‘stdvIsFactor = false’ means that absolute values are used.
- ‘stdvIsFactor = true’ means that relative values are used, i.e. as a fraction of the observation.

An offset can be added to the observed quantity by associating this value with the parameter ‘mean’. This only functions of absolute values are used, i.e. if ‘stdvIsFactor = false’. An observational data point can be omitted from the calibration procedure by putting ‘isActive=false’.

An example of this file is shown in Figure 3.6. A part of the content of this file is repeated here and explained:

```
<probabilityDistributionFunction id="Hsig @ 5.7,0.0" isActive="true">
    <normal mean="0.0" stdv="0.1" stdvIsFactor="true"/>
</probabilityDistributionFunction>
```

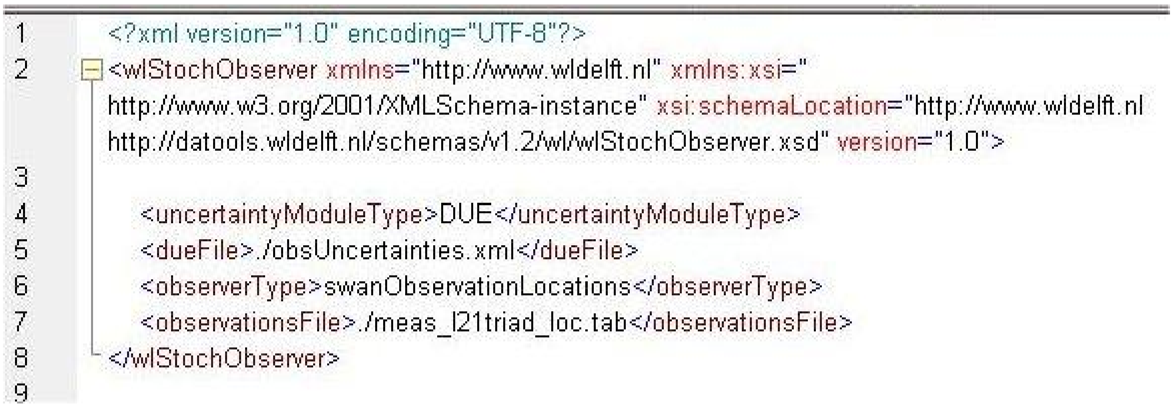
Here, the normal distribution is applied for the quantity Hsig at location (x,y) = (5.7,0.0). The spread is 0.1 times the observation value.



```xml
<?xml version="1.0" encoding="UTF-8"?>
<uncertainties xsi:schemaLocation="http://www.wldelft.nl
http://datools.wldelft.nl/schemas/v1.3/uncertainties.xsd" version="1.0" xmlns="http://www.wldelft.nl" xmlns:xsi
="http://www.w3.org/2001/XMLSchema-instance">

    <uncertaintyType>ProbabilityDistributionFunction</uncertaintyType>

    <probabilityDistributionFunction id="Tm02 @ 0.0,0.0" isActive="true">
        <normal mean="0.0" stdv="0.1" stdvIsFactor="true"/>
    </probabilityDistributionFunction>
    <probabilityDistributionFunction id="Tm02 @ 5.7,0.0" isActive="true">
        <normal mean="0.0" stdv="0.1" stdvIsFactor="true"/>
    </probabilityDistributionFunction>
    <probabilityDistributionFunction id="Tm02 @ 10.5,0.0" isActive="true">
        <normal mean="0.0" stdv="0.1" stdvIsFactor="true"/>
    </probabilityDistributionFunction>
    <probabilityDistributionFunction id="Tm02 @ 12.5,0.0" isActive="true">
        <normal mean="0.0" stdv="0.1" stdvIsFactor="true"/>
    </probabilityDistributionFunction>
    <probabilityDistributionFunction id="Tm02 @ 13.5,0.0" isActive="true">
        <normal mean="0.0" stdv="0.1" stdvIsFactor="true"/>
    </probabilityDistributionFunction>
    <probabilityDistributionFunction id="Tm02 @ 14.5,0.0" isActive="true">
        <normal mean="0.0" stdv="0.1" stdvIsFactor="true"/>
    </probabilityDistributionFunction>
    <probabilityDistributionFunction id="Tm02 @ 15.7,0.0" isActive="true">
        <normal mean="0.0" stdv="0.1" stdvIsFactor="true"/>
    </probabilityDistributionFunction>
    <probabilityDistributionFunction id="Tm02 @ 17.3,0.0" isActive="true">
        <normal mean="0.0" stdv="0.1" stdvIsFactor="true"/>
    </probabilityDistributionFunction>


</uncertainties>

```

Figure 3.6 Content of the file obsUncertainties.xml

The file swanStochObsConfig.xml is a configuration file for the Stochastic Observer. An example of its content is shown in Figure 3.7. The user needs to insert the correct name of the observational data file in line 7.



```xml
1  <?xml version="1.0" encoding="UTF-8"?>
2  <wlStochObserver xmlns="http://www.wldelft.nl" xmlns:xsi="
   http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.wldelft.nl
   http://datools.wldelft.nl/schemas/v1.2/wl/wlStochObserver.xsd" version="1.0">
3
4      <uncertaintyModuleType>DUE</uncertaintyModuleType>
5      <dueFile>./obsUncertainties.xml</dueFile>
6      <observerType>swanObservationLocations</observerType>
7      <observationsFile>./meas_l21triad_loc.tab</observationsFile>
8  </wlStochObserver>
9
```

Figure 3.7 Content of the file swanStochObsConfig.xml

3.4.3 Subdirectory \swanModel

In this subdirectory, the SWAN model, the SWAN model parameters and the SWAN results after execution of the C-I are stored. This subdirectory contains, before execution of the C-I, four subdirectories:

- bin
- config
- parameters
- work

The content of these directories is discussed in the present section.

After execution of the C-I, additional subdirectories are created with names:

- work00
- work01
- work02
- ...
- work*xx* (*xx* is a sequence number associated with the calibration process, see Section 3.6)

Their content is discussed in Section 3.6.

Subdirectory \bin

This directory contains the SWAN executable, e.g. swan4051A.exe, for a Windows platform (directory \win32) and a Linux platform (directory \linux).

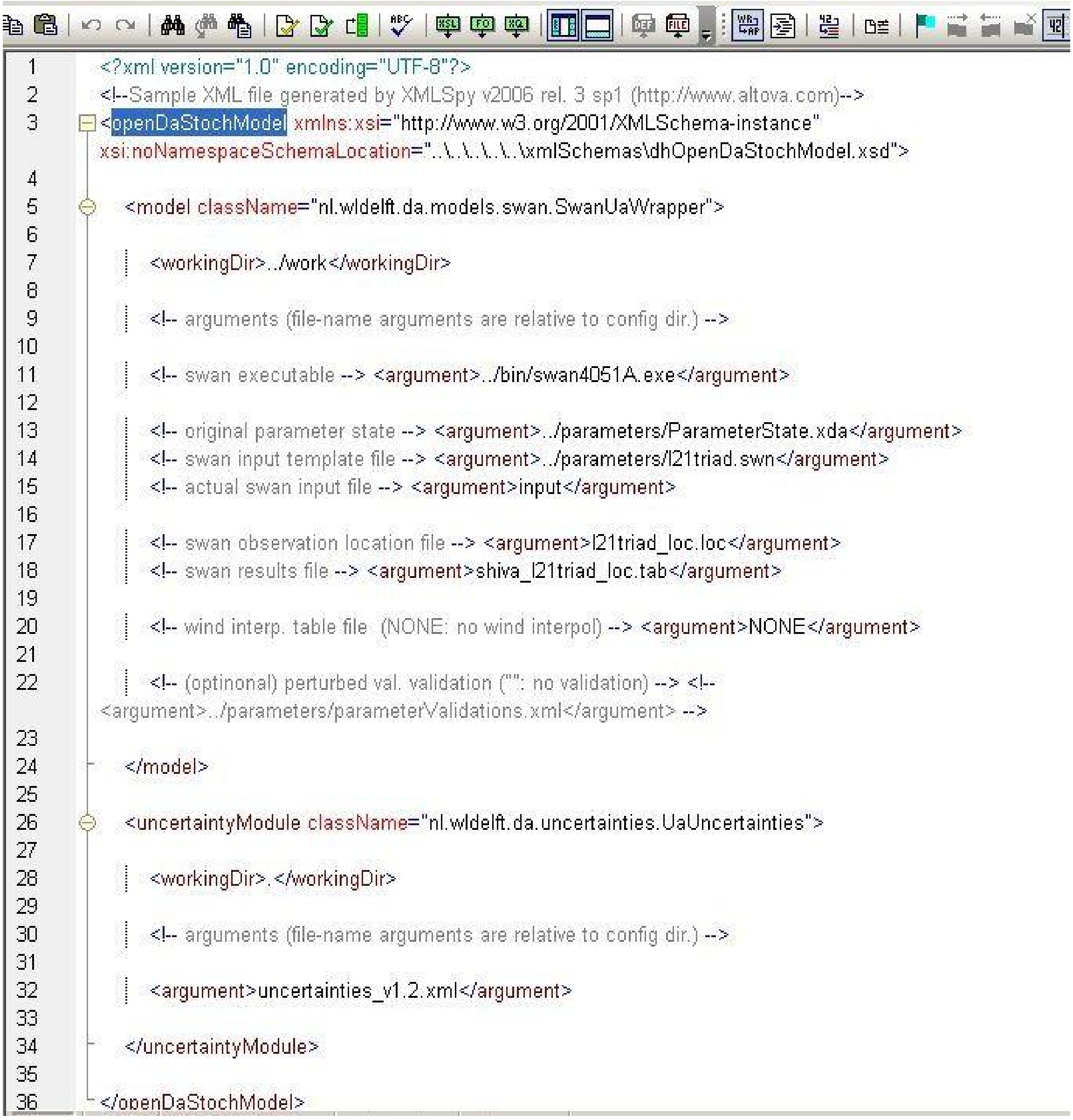
Subdirectory \config

This directory contains three files:

- openDaStochModel.xml
- openDaStochModel_linux.xml
- uncertainties_v1.2.xml

The file openDaStochModel.xml contains information on the execution of the SWAN model and on the model parameters. The user needs to replace the correct name of the SWAN case (here: l21triad) in lines 14, 17 and 18, see also Figure 3.8, while keeping other parts of the filename fixed.

The file openDaStochModel_linux.xml is similar to the file openDaStochModel.xml, but is used on a Linux platform.



```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2006 rel. 3 sp1 (http://www.altova.com)-->
<openDaStochModel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="..\..\..\..\xmlSchemas\dhOpenDaStochModel.xsd">

    <model className="nl.wldelft.da.models.swan.SwanUaWrapper">

        <workingDir>../work</workingDir>

        <!-- arguments (file-name arguments are relative to config dir.) -->

        <!-- swan executable --> <argument>../bin/swan4051A.exe</argument>

        <!-- original parameter state --> <argument>../parameters/ParameterState.xda</argument>
        <!-- swan input template file --> <argument>../parameters/l21triad.swn</argument>
        <!-- actual swan input file --> <argument>input</argument>

        <!-- swan observation location file --> <argument>l21triad_loc.loc</argument>
        <!-- swan results file --> <argument>shiva_l21triad_loc.tab</argument>

        <!-- wind interp. table file  (NONE: no wind interpol) --> <argument>NONE</argument>

        <!-- (optinonal) perturbed val. validation ("": no validation) --> <!-- <argument>../parameters/parameterValidations.xml</argument> -->

    </model>

    <uncertaintyModule className="nl.wldelft.da.uncertainties.UaUncertainties">

        <workingDir>.</workingDir>

        <!-- arguments (file-name arguments are relative to config dir.) -->

        <argument>uncertainties_v1.2.xml</argument>

    </uncertaintyModule>

</openDaStochModel>
```

Figure 3.8 Content of the file openDaStochModel.xml

The file uncertainties_v1.2.xml contains information on the uncertainties associated with the SWAN model parameters, cf. Table 2.1. These are used to determine the initial step size in the search mechanism in the calibration algorithm. The uncertainties are also used in the option in which the GoF is extended with weak or soft constraints. In that case they control the contribution of the constraint term to the GoF, see Section 2.1, paragraph 2.1.2.
Only the SWAN case parameters, i.e. the model parameters that are calibrated in the SWAN case under consideration, must be present in this file. Only a normal distribution of the uncertainty is allowed. The stdv (standard deviation) should be defined in absolute or relative values:

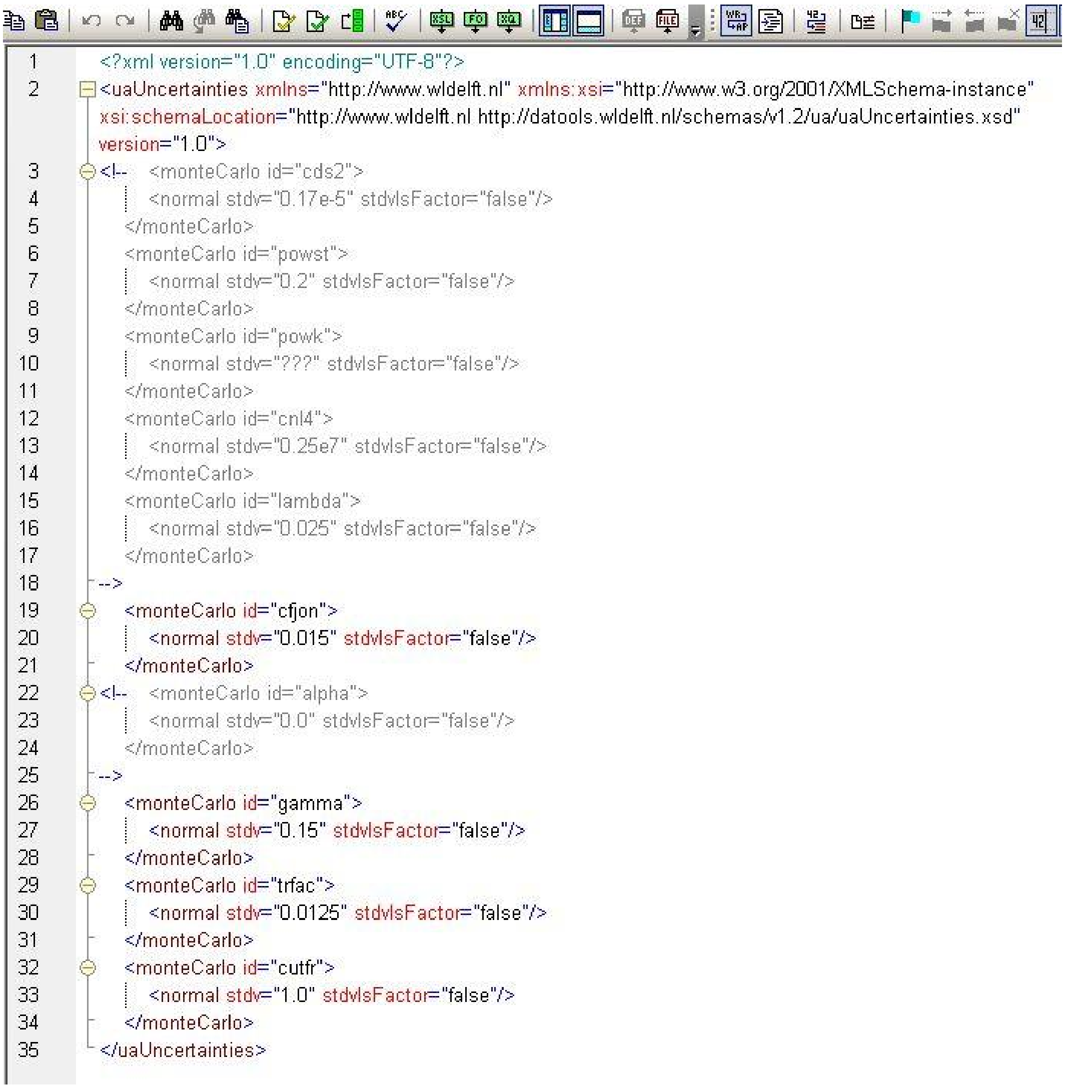
- ‘stdvIsFactor = false’ means that absolute values are used.
- ‘stdvIsFactor = true’ means that relative values are used, i.e. as a fraction of the base value (see below).

Example:

```
<monteCarlo id="cds2">
    <normal stdv="0.17E-5" stdvIsFactor="false"/>
</monteCarlo>
```

In this example the normal distribution is applied to the quantity cds2 with a standard deviation equal to 0.17E-5.

The content of the file uncertainties_v1.2.xml for our example of Chapter 2 is shown in Figure 3.9. It is recommended to keep the other parameters in this file in the form of a comment statement as is visible in the figure; this facilitates the work of the user.



```
<?xml version="1.0" encoding="UTF-8"?>
<uaUncertainties xmlns="http://www.wldelft.nl" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wldelft.nl http://datools.wldelft.nl/schemas/v1.2/ua/uaUncertainties.xsd"
version="1.0">
<!--   <monteCarlo id="cds2">
         <normal stdv="0.17e-5" stdvIsFactor="false"/>
     </monteCarlo>
     <monteCarlo id="powst">
         <normal stdv="0.2" stdvIsFactor="false"/>
     </monteCarlo>
     <monteCarlo id="powk">
         <normal stdv="???" stdvIsFactor="false"/>
     </monteCarlo>
     <monteCarlo id="cnl4">
         <normal stdv="0.25e7" stdvIsFactor="false"/>
     </monteCarlo>
     <monteCarlo id="lambda">
         <normal stdv="0.025" stdvIsFactor="false"/>
     </monteCarlo>
-->
     <monteCarlo id="cfjon">
         <normal stdv="0.015" stdvIsFactor="false"/>
     </monteCarlo>
<!--   <monteCarlo id="alpha">
         <normal stdv="0.0" stdvIsFactor="false"/>
     </monteCarlo>
-->
     <monteCarlo id="gamma">
         <normal stdv="0.15" stdvIsFactor="false"/>
     </monteCarlo>
     <monteCarlo id="trfac">
         <normal stdv="0.0125" stdvIsFactor="false"/>
     </monteCarlo>
     <monteCarlo id="cutfr">
         <normal stdv="1.0" stdvIsFactor="false"/>
     </monteCarlo>
</uaUncertainties>
```

Figure 3.9 Content of the file uncertainties_v1.2.xml

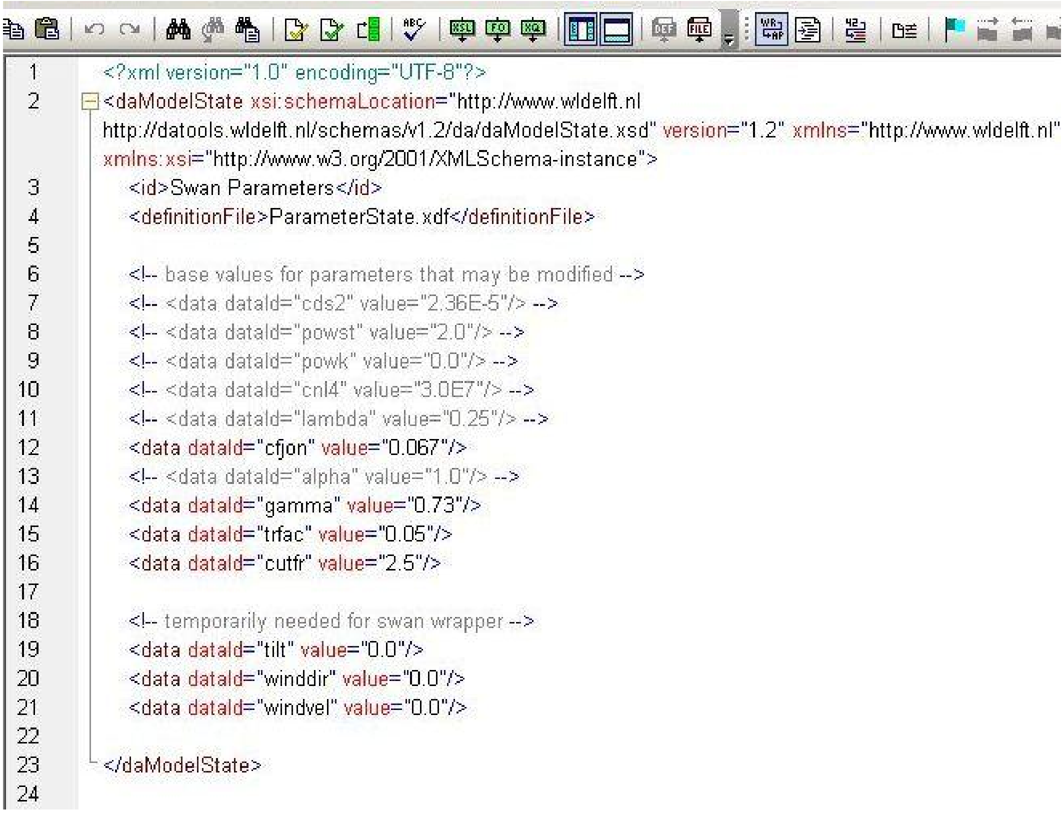
Subdirectory \parameters

This directory contains three files:

- *casename*.swn, where *casename* is the name of the SWAN case (l21triad in our example)
- ParameterState.xda
- ParameterState.xdf

The file *casename*.swn is the SWAN command file. In this file, placement holders are present for the values of the SWAN case parameters (the model parameters that are calibrated in the SWAN case under consideration), see the discussion in Section 2.3.

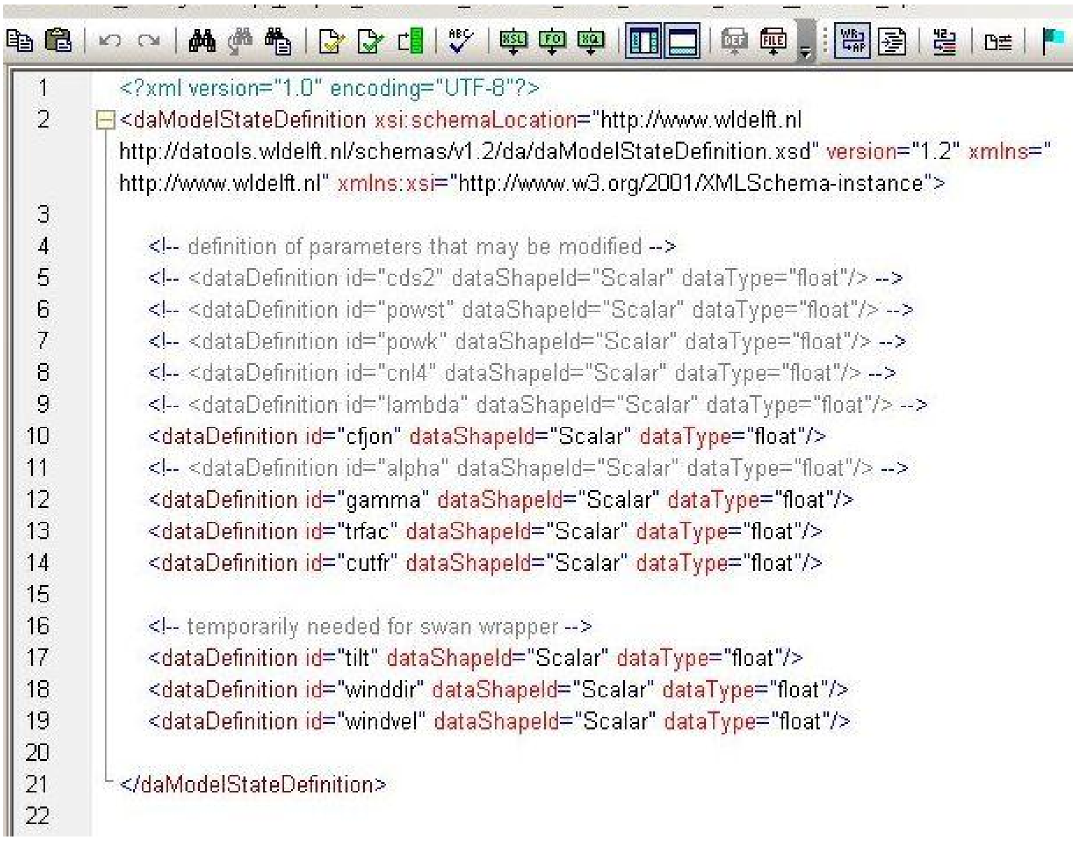
The file ParameterState.xda contains the base values of the SWAN case parameters. The content of this file for our example of Chapter 2 is shown in Figure 3.10. It is recommended to keep the other parameters in this file in the form of a comment statement as is visible in the figure; this facilitates the work of the user.



```
<?xml version="1.0" encoding="UTF-8"?>
<daModelState xsi:schemaLocation="http://www.wldelft.nl
http://datools.wldelft.nl/schemas/v1.2/da/daModelState.xsd" version="1.2" xmlns="http://www.wldelft.nl"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <id>Swan Parameters</id>
    <definitionFile>ParameterState.xdf</definitionFile>

    <!-- base values for parameters that may be modified -->
    <!-- <data dataId="cds2" value="2.36E-5"/> -->
    <!-- <data dataId="powst" value="2.0"/> -->
    <!-- <data dataId="powk" value="0.0"/> -->
    <!-- <data dataId="cnl4" value="3.0E7"/> -->
    <!-- <data dataId="lambda" value="0.25"/> -->
    <data dataId="cfjon" value="0.067"/>
    <!-- <data dataId="alpha" value="1.0"/> -->
    <data dataId="gamma" value="0.73"/>
    <data dataId="trfac" value="0.05"/>
    <data dataId="cutfr" value="2.5"/>

    <!-- temporarily needed for swan wrapper -->
    <data dataId="tilt" value="0.0"/>
    <data dataId="winddir" value="0.0"/>
    <data dataId="windvel" value="0.0"/>

</daModelState>
```

Figure 3.10 Content of the file ParameterState.xda

The file ParameterState.xdf contains the definition of the SWAN case parameters. The content of this file for our example in Chapter 2 is shown in Figure 3.11. It is recommended to keep the other parameters in this file in the form of a comment statement as is visible in the figure; this facilitates the work of the user.



```
<?xml version="1.0" encoding="UTF-8"?>
<daModelStateDefinition xsi:schemaLocation="http://www.wldelft.nl
http://datools.wldelft.nl/schemas/v1.2/da/daModelStateDefinition.xsd" version="1.2" xmlns="
http://www.wldelft.nl" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

    <!-- definition of parameters that may be modified -->
    <!-- <dataDefinition id="cds2" dataShapeId="Scalar" dataType="float"/> -->
    <!-- <dataDefinition id="powst" dataShapeId="Scalar" dataType="float"/> -->
    <!-- <dataDefinition id="powk" dataShapeId="Scalar" dataType="float"/> -->
    <!-- <dataDefinition id="cnl4" dataShapeId="Scalar" dataType="float"/> -->
    <!-- <dataDefinition id="lambda" dataShapeId="Scalar" dataType="float"/> -->
    <dataDefinition id="cfjon" dataShapeId="Scalar" dataType="float"/>
    <!-- <dataDefinition id="alpha" dataShapeId="Scalar" dataType="float"/> -->
    <dataDefinition id="gamma" dataShapeId="Scalar" dataType="float"/>
    <dataDefinition id="trfac" dataShapeId="Scalar" dataType="float"/>
    <dataDefinition id="cutfr" dataShapeId="Scalar" dataType="float"/>

    <!-- temporarily needed for swan wrapper -->
    <dataDefinition id="tilt" dataShapeId="Scalar" dataType="float"/>
    <dataDefinition id="winddir" dataShapeId="Scalar" dataType="float"/>
    <dataDefinition id="windvel" dataShapeId="Scalar" dataType="float"/>

</daModelStateDefinition>

```

Figure 3.11 Content of the file ParameterState.xdf

Subdirectory \work

In this directory, the other SWAN input files (related to bathymetry, grid, boundary conditions, wind field, current field) need to be put. In our example, these are the files l21triad.bnd (boundary conditions) and l21triad.bot (bathymetry).

3.5 Execution of the calibration process

The user needs to go to the directory [C-I root]\Bin, and – if not done already – create the files SwanApplication.exe and SwanApplication.jpif as discussed in Section 3.3.1.

Execution of the C-I is done by either mouse clicking the executable SwanApplication.exe, or by typing SwanApplication.exe in the command line of the Total Commander. We note that typing SwanApplication.exe and then pressing Enter in a DOS box does not work; for some reason DOS does not go along with the jpif-file format.

A simple progress monitor showing the progress of the SWAN computations is present. In the next software release, key user information on the optimisation progress will be added to it in order to provide the user information on convergence and intermediate values of the parameters. This will allow the user to take decisions on interrupting the process at an early stage, e.g. in case of divergence, which will enhance efficiency in case of optimisation experiments with large applications, which generally require many hours of computing time.

3.6 Outcome of the calibration process

After execution of the C-I, the directories \work00, \work01, \work02, ..., \work*xx* (*xx* is a number, which may in occurring cases become larger than 99, resulting in \work100, \work101 etc.) are created as subdirectories in the directory [C-I root]\model*casename*\swanModel\. The directory \work00 is a copy of the directory \work. The directories \work01, \work02, ..., \work*xx* contain the SWAN input and output files for a SWAN run. In each of these runs, numerical values for the SWAN model parameters have been inserted in the SWAN command file, now named INPUT. The calibration algorithm chooses the numerical values for the SWAN parameters judiciously.

After execution of the C-I, two logfiles are created in directory [C-I root]\model*casename*:

- swan[*CalAlg*]OpenDaConfig-log.txt
- swan[*CalAlg*]OpenDaConfig-sysout.txt

where [*CalAlg*] is a string referring to the selected calibration algorithm (Dud, Powell or Simplex).

In case of a calibration attempt that is unsuccessful, e.g. due to errors in the input Xml-files, an error message is displayed in the file swan[*CalAlg*]OpenDaConfig-log.txt. In case of a successful attempt, this file is empty.

The file swan[*CalAlg*]OpenDaConfig-sysout.txt contains information on the calibration process and on the SWAN case parameters. Below a typical part of the information is given (there exist slight variations for the different calibration algorithms, mainly due to additional information that is specific for that particular calibration algorithm):

```
%%===========================================================
%%evaluation number
%no=23
%   parameters = [0.08768094851044683,0.642971813678741 5,0.04528953202500241,3.1581368446350098]
Execution of D:\CalInstr\bin\..\model\l21triad\config\..\swanModel\.\config\..\work23\..\bin\swan4051A.exe
successful.
%   predicted=[1.2787,0.02321,1.6019,1.2694,0.02266,1.5947,...,...];
%   observed=[1.265,0.023,1.603,1.265,0.023,1.603,1.312,0.025,...,...];
%   residuals=[-0.013700000000000045,-2.100000000000185E-4,0.0010999999999998789,...,...];
%   cost=3.4151035884728294;
%%===========================================================
```

This information pertains to one evaluation, i.e. one SWAN run. Here, the evaluation number is 23, so the corresponding data is stored in directory \work23. The employed values of the SWAN model parameters are shown here (four in this case). The vectors ‘predicted’ and ‘observed’ correspond to respectively the SWAN values[2] and observed values[3] for the quantities against which calibration is performed. The vector ‘residuals’ equals the difference between the vectors observed and predicted. The scalar value ‘cost’ is the GoF.

[2] In file [C-I root]\model*casename*\swanModel\work*xx*\shiva_*casename*_loc.tab

[3] In file [C-I root]\model\casename\stochobserver\meas_*casename*_loc.tab

At the end of the file openDaApplication-SystemOutlog.txt, the following is written:

```
%==============================================================
%
% SimulationKwadraticCostfunction :
%
% number of evaluations
n = 30;
% all cost values
costs = [3.600314273782221,3.6037176137982843,3.600314273782221,...,...];
% all parameter values
parameters=[0.06700000166893005,0.08700000122189522,0.06700000166893005,...,...];
% number of observations
nobs=24;
% best cost
costOpt = 3.1538362655661123;
% best parameters
pOpt =[0.09213189426514745,0.4918911457061767 6,0.05028823689055795,2.9513607025146484];
%==============================================================
```

First, the total number of evaluations (SWAN runs) is given. Then, the collection of all costs and SWAN model parameters for all evaluations are summarized into two long vectors ‘costs’ and ‘parameters’. The total number of observations, which is typically equal to the number of observational locations times the number of quantities against which validation is performed, is given under ‘nobs’. The best (smallest) value for the GoF is given under ‘costOpt’. The optimal set of parameters is given in the vector ‘pOpt’. In case some observations are missing, the number ‘nobs’ is lower.

4 References

Alkyon (2003). Calibration of SWAN 40.20 for field cases Petten, Slotermee and Westerschelde. Alkyon report A1168 prepared for Rijkswaterstaat/RIKZ, December 2006.

DUT (2007). Delft University of Technology. SWAN User Manual. Version 40.51A, SWAN Cycle III, 2007. Can be downloaded from the SWAN website: http://vlm089.citg.tudelft.nl/swan/index.htm

Komen, G.J., S. Hasselmann, K. Hasselmann (1984). On the existence of a fully developed wind-sea spectrum. J. of Phys. Oceanogr. 14, 1271-1285.

SHiVa website. http://demo.xi-alles.nl/shiva_download/ (password required)

Van der Westhuysen, A.J., M. Zijlema, J.A. Battjes (2007). Nonlinear saturation-based whitecapping dissipation in SWAN for deep and shallow water. Coast. Eng. 54, 151-170.

WL, 2006. Gerritsen, H., A.J. van der Westhuysen, G. Ph. van Vledder, J. Groeneweg, G. Ruessink en H.F.P. van den Boogaard. Calibration tool for SWAN. Activity 8.1 and 8.2 of SBW project Waddenzee. Part 1: analysis of requirements. WL|Delft Hydraulics, Report X0346, 31 pp.

WL, 2007a. Van den Boogaard, H.F.P., G. Ruessink, H. Gerritsen, G. Ph. van Vledder, A.J. van der Westhuysen, J. Groeneweg. Calibration tool for SWAN. Activity 8.2 of SBW project Waddenzee. Part II: Selection of appropriate calibration techniques. WL | Delft Hydraulics, Report X0346, 54 pp.

WL, 2007b. Gerritsen, H., S. Hummel, M. Verlaan. Calibration tool for SWAN – Detailed Specification Calibration Instrument. Activity 8.3 of SBW project Waddenzee. WL | Delft Hydraulics, Report H4918.65, 57 pp.

WL, 2007c. Hummel, S., M. Mens. DaTools User Manual. WL | Delft Hydraulics, Draft Report, 33 pp.

WL, 2007d. Weerts, A. DATools. WL | Delft Hydraulics, Report Q3944 & X0354.

WL, 2007e. Beckers, J.V.L and P.F.C. van Geer. Uncertainty analysis of Hydraulic Boundary Conditions of the Wadden Sea – results of the uncertainty analysis of the HBC chain and SWAN. Draft report H4918.32.

Xi, 2007a. Dekker, P.J., F. Kleissen, E. Maliepaard. SHiVa – Design and Prototype. Xi. Version 1.0.

Xi, 2007b. Dekker, P.J., F. Kleissen, E. Maliepaard. SHiVa – GUI User Manual. Xi. Version 1.0.

A SHiVa conventions – nomenclature and format of files

This chapter contains an integral copy of Chapter 4 of the SHiVa GUI User Manual v0.1, see Xi (2007b).

4 Input and output files: nomenclature

4.1 Introduction

For each SWAN case a number of input and output files are used respectively generated. The content of these files differs largely: from textual input files to numerical output data in matrix format.

SHiVa can only execute automated actions if certain conventions are adhered to. These conventions relate to the filenames (Section 4.2), certain parts of the SWAN command file (Section 4.2.1) and the format of the observational data files (Section 4.2.3). For reasons of completeness, the format of SWAN output is briefly mentioned in Section 4.4. The relation between observational locations, SWAN output locations and observational data files is discussed in Section 4.6.

Adhering to these conventions also ensures that SHiVa is able to find all possible data files (input and output) in the case directories.

Identification of a file is based on the meta–information related to the content. This meta–information can be stored in either the filename or the file itself. For SWAN related files both methods are used.

4.2 Filenames

Three file categories can be distinguished in SHiVa:

- **SWAN input files**
  SWAN input files are necessary to run SWAN. Each case has a number of SWAN input files. This number varies from case to case. Each case has one SWAN command file. This file contains instructions of the user to SWAN with respect to physical and numerical settings, names of the model input files, output settings, etc.. The model input files (if present) contain information on the computational grid, bathymetry, water level, current, friction, wind and wave boundary conditions. The SWAN input files are provided by the user or are supplied with the cases that are retrieved from the server. SHiVa does not create SWAN input files, though SHiVa offers the option to adjust physical parameter settings, see Section 4.3.
- **SWAN output files**
  SWAN output files are created by SWAN during execution. Output files contain data to be used by SHiVa for comparison against observational data or against other SWAN output data and for presentation purposes.
- **Observational data files**
  Observational data files contain observational data. SWAN is validated against this data. Observational data files are provided by the user or are supplied with the cases that are retrieved from the server.

In SHiVa, a validation case is identified by a unique code which must be contained in all filenames related to that case. The code name is build up as:

$$\begin{vmatrix} a \\ l \\ f \end{vmatrix} \begin{vmatrix} iii \end{vmatrix} \begin{vmatrix} ccccc \end{vmatrix} \begin{vmatrix} iii \end{vmatrix}$$

where

Table 4.1 Code description

item	description
1st block	a = analytical case, l = laboratory case, f = field case
2nd block	iii = case identifying code (3 integers)
3rd block	ccccc = case identifying code (5 characters)
4th block	iii = subcase identifying code (3 integers)

for example: f051fries003 or l012triad001.

The filename itself can have any length but must consist of alphanumeric characters (0…9,a…z). Additional parts in the filenames, if present, need to be separated by an underscore. Furthermore, each filename can be split in a name and an extension. The extension is also used for identification. An example is:

shiva_f051fries001_loc01.sp1

where `f051fries001` is the unique case code, `shiva` and `loc01` are additional parts to be explained in the following sections, and `sp1` the extension.

The following sections describe the conventions for filenames which need to be adhered to.

4.2.1 SWAN input files

In the following, **code** refers to the codename as described above.

Table 4.2 SWAN input files

FILENAME	CONTENT	ACTION
INPUT	SWAN input.	Is a copy of ***code**.swn*, with the placement holders replaced by the selected parameter settings. This copy is generated by the SHiVa GUI.
swan.exe* (*: wildcard, identifying SWAN version number, eg 4051)	SWAN executable.	Is part of the case. Executables are stored in the directory **swan_executables**. Must be copied to case run directory.
***code**.swn*	SWAN command file.	Certain parameters can be adjusted by user through the SHiVa GUI.
***code**_loc.loc*	File containing the observation locations for validation.	See remark below.
runswan.bat	Launches SWAN.	Must be present in the case run directory.

Remarks:

- The table above does not contain all SWAN input files. For the SWAN input files not present in the table, no strict filename conventions hold. This poses no problems for SHiVa, since their names are not used in the SHiVa operations. However, it is, of course, recommendable to use convenient names and extensions. For example, the wind field file may be named *code.wnd* and the wind grid file may be named *code_wnd.grd*.
- The file *code_loc.loc* contains the SWAN output coordinates of all observational locations for which validation is done. Note that SWAN to SWAN comparison is done in the locations given in file

code_loc.loc as well, although (strictly speaking) these locations are not observational locations, but rather 'validation locations'.

4.2.2 SWAN output files

Table 4.3 SWAN output files

FILENAME	CONTENT	ACTION
PRINT	Log file	SHiVa checks for warnings and convergence
norm_end	Normal end of run	SHiVa checks for existence of this file
swaninit	SWAN initialisation file	Is generated by the SWAN executable. No further action required.
*shiva_**code**.cuv*	SWAN output of wave parameters in table format as a predefined curve (1D representation).	Used for function plots of wave parameter versus a coordinate or distance from origin
*shiva_**code**.mat*	SWAN block output of wave parameters in matlab format for all computational grid points. Only for 2D cases.	Used for area plots of wave parameter versus (x,y)–coordinate
*shiva_**code**_loc.sp1*	SWAN 1D–spectral output in all observational locations given in file ***code**_loc.loc*	Plot versus observed 1D spectra (stored in files *meas_**code**_loc*.sp1*)
*shiva_**code**_loc.sp2*	SWAN 2D–spectral output in all observational locations given in file ***code**_loc.loc*	Plot versus observed 2D spectra (stored in files *meas_**code**_loc*.sp2*)
*shiva_**code**_loc.tab*	SWAN output of wave parameters in table format in all observational locations given in file ***code**_loc.loc*	Plot and statistical analysis versus observed wave parameters (stored in file *meas_**code**_loc.tab*)

Remarks:

- SWAN must write the desired data to files with the correct names. This imposes requirements on the SWAN command file, see Section 4.3.
- SWAN output files containing wave data for validation all start with '*shiva_*'.
- SWAN can only generate block data (in Matlab binary format) for 2D cases. For 1D cases, the table format is used (the shiva ***code***.cuv files).
- The curve data (shiva_**code**.cuv) are mainly used for 1D cases. For some 2D cases, like a011refra and a021shoal, the use of 1D output along a curve is useful. For this selected set of 2D cases, both curve data and mat data needs to be output.
- If the user wishes to output data in other locations than the observational locations, this data should be stored in files with names different from the ones given above.

4.2.3 Observational data files

Table 4.4 Observational data files

FILENAME	CONTENT	ACTION
*meas_**code**_loc*.sp1*	Observed 1D spectra for the location identified by '*'	Plot versus computed 1D spectra (stored in file *shiva_**code**_loc.sp1*)

FILENAME	CONTENT	ACTION
*meas_**code**_loc*.sp2*	Observed 2D spectra for the location identified by '*'	Plot versus computed 2D spectra (stored in file *shiva_**code**_loc.sp2*)
*meas_**code**_loc.tab*	Observed integral wave parameters	Plot and statistical analysis versus computed integral wave parameters (stored in file *shiva_**code**_loc.tab*)

Remarks:

- The symbol '*' in the table represents two integer numbers, so '01', '02', '03', etc. Note that '00' is not used. These numbers refer to the position (line number) of the output location in ***code**_loc.loc* file. In Section 4.6, this issue is elaborated on further.
- Note that labels in the filenames do not contain a reference to the name of the observational location (eg, ELD or SCW) or to the time of measurement. The reason for this is twofold:
  1) not all observational locations have a name, and
  2) some users simply may not like this. Note, however, that it is possible to store this sort of meta information in the comment header of the files.

4.3 Format of the SWAN command file

Each case has one SWAN command file that contains instructions of the user to SWAN. Some parts of the command file must adhere to certain conventions in order to make use of all SHiVa functionalities. These conventions relate to the following issues:

- Settings of physical parameters
- Definition of SWAN output

For the other parts in the SWAN command file, no conventions hold.

4.3.1 Settings of physical parameters

The SHiVa user can set values for a subset of physical parameters steering SWAN. The selection of this subset is based on the requirements for the Calibration–Instrument (see **Gerritsen et al 2006**) and (**Gerritsen et al 2007**) and the settings employed for HR2006. The adopted convention is that these parameters must be stated in the SWAN command file as follows:

```
$ --- Begin of settings physical process parameters
$
GEN3
<#WCAPON#>WCAP KOM cds2=<#cds2#> stpm=<#stpm#> powst=<#powst#> delta=<#delta#> powk=<#powk#>
<#WCAPOFF#>OFF WCAP
<#QUADON#>QUAD iquad=<#iquad#> lambda=<#lambda#> Cnl4=<#Cnl4#>
<#QUADOFF#>OFF QUAD
<#QUADON#>LIMITER ursell=<#ursell#> qb=<#qb#>
<#FRICON#>FRIC JONSWAP cfjon=<#cfjon#>
<#FRICOFF#>$
<#BREAON#>BREA CON alpha=<#alpha#> gamma=<#gamma#>
<#BREAOFF#>OFF BREA
<#TRIADON#>TRIAD trfac=<#trfac#> cutfr=<#cutfr#>
<#TRIADOFF#>$
$
$ --- End of settings physical process parameters
```

A search–and–replace tool or the SWAN wrapper is then used to replace the placement holder strings `<#...#>` by the selected settings.

Remarks:

- In various cases, certain physical processes are excluded in the SWAN computation, and therefore not all physical parameters given above can be set. For example, when wind input is absent (e.g,. case l021triad in the ONR Testbed), quadruplet interaction must be de–activated using the command '`OFF QUAD`'. Another example is case a21shoal, in which quadruplet interaction, wave breaking and white capping are de–activated, using the commands '`OFF QUAD`', '`OFF BREA`' and '`OFF WCAP`'.
  This is ensured by the use of the placement holder strings `<#...ON#>` and `<#...OFF#>`. If the physical process is included in the computation, then the `<#...ON#>` string is removed and the `<#...OFF#>` string is replaced by a '`$`'–sign. The latter signifies a comment statement, so that the remainder of the line is not used by SWAN. If the physical process is excluded from the computation, then the `<#...OFF#>` string is removed and the `<#...ON#>` string is replaced by a '`$`'–sign.
  Note that if the keyword '`TRIAD`' is absent, triads are automatically excluded from SWAN; this means that there is no `OFF TRIAD` comment. This also holds for bottom friction (keyword FRIC).
- Suppose the user wants to study the sensitivity of variations in one physical parameter. Then a SWAN command file must be provided in which all physical parameters are already set with their preferred values, with the exception of the one under study. For this one, a placement holder string as illustrated above must be inserted. The situation in which a user wants to study variations in more physical parameters is done similarly.

The selection of parameter settings available in SHiVa is described in detail in Section 3.2.2.

4.3.2 Definition of SWAN output

The following output is generated for SHiVa:

- Curve data or block data. This data consists of integral wave parameters in all computational grid points. This data is used for spatial distribution plots: function plots of type integral wave parameter along a predefined curve, or area plots of type integral wave parameter versus the (x,y)–coordinates.
- Tabular, 1D–spectral and 2D–spectral data. This data consists of tabulated integral wave parameters (the same list as given in the previous bullet), 1D spectra and 2D spectra in all observational locations.

The output parameters from SWAN which are available through SHiVa[4] are listed and described in Table 4.5.

Table 4.5 Available output parameters from SWAN through SHiVa

PARAMETER	UNITS	DESCRIPTION
XP		X coordinate of observational location (in problem coordinate system)
YP		Y coordinate of observational location (in problem coordinate system)
DEP	[m]	water depth
BOTLEV[5]	[m NAP]	bottom level with respect to reference level
WATLEV[5]	[m NAP]	water level with respect to reference level
HS	[m]	Significant wave height Hm0
RTP	[s]	peak period of the variance density spectrum (relative frequency spectrum)
TMM10	[s]	mean wave period $T_{m-1,0}$
TM01	[s]	mean wave period $T_{m0,1}$
TM02	[s]	mean wave period $T_{m0,2}$
FSPR	[–]	normalised width of the frequency spectrum
DIR	[degrees]	mean wave direction

[4] Please note that more output parameters are available from SWAN, the given selection was chosen for SHiVa v0.2 and may be enlarged in future.

[5] WATLEV en BOTLEV are only available from SWAN version 40.51 and onwards

PARAMETER	UNITS	DESCRIPTION
DSPR	[degrees]	directional spreading of the waves
WLENGTH	[m]	mean wavelength
TPS	[s]	'smoothed' peak period
DHSIGN	[m]	difference in significant wave height as computed in the last two iterations
DRTM01	[s]	difference in the mean wave period (RTM01) as computed in the last two iterations
WIND	[m/s]	wind velocity
VEL	[m/s]	current velocity
DISSIP	[m^2/s]	energy dissipation due to bottom friction, wave breaking and whitecapping
QB	[–]	fraction of breaking waves due to depth–induced breaking
FORCE	[N/m^2]	wave induced force per unit surface area (vector)
UBOT	[m/s]	the rms–value of the maxima of the orbital velocity near the bottom
STEEP	[–]	mean wave steepness
SETUP	[m]	setup due to waves

This output is generated using the following statements in the SWAN command file:

```
$ --- Begin of SHiVa output
if ( 2D case ) then

$     Output block data in Matlab format
BLOCK 'COMPGRID' NOHEADER 'shiva_code.mat' LAYOUT 3 &
       XP YP WATLEV BOTLEV HS TPS TMM10 TM01 TM02 DIR DSPR FRCOEF VEL WIND
$

end if

if ( curve data needs to be written )

$     Write output locations on a curve (equidistantly distributed)
CURVE  'LINE1' [xp1] [yp1] [int] [xp] [yp]
TABLE  'LINE1' HEAD 'shiva_code.cuv' &
        XP YP DEP BOTLEV WATLEV HS RTP TMM10 TM01 TM02 FSPR DIR DSPR   &
        WLENGTH TPS DHSIGN DRTM01 WIND VEL DISSIP QB FORCE UBOT STEEP SETUP
$

end if

$     Define locations where data must be output
POINTS 'SHIVA' FILE   'code_loc.loc'
$
$     Write output data (tables, 1D and 2D spectra) in
$     these locations
TABLE  'SHIVA' HEAD   'shiva_code_loc.tab' &
        XP YP DEP BOTLEV WATLEV HS RTP TMM10 TM01 TM02 FSPR DIR DSPR   &
        WLENGTH TPS DHSIGN DRTM01 WIND VEL DISSIP QB FORCE UBOT STEEP SETUP
SPEC   'SHIVA' SPEC1D 'shiva_code_loc.sp1'
SPEC   'SHIVA' SPEC2D 'shiva_code_loc.sp2'
$
$ --- End of SHiVa output
```

Remarks:

- The bold underlined if–then–statement is not present in the SWAN command file, but is pseudo–code to indicate whether certain output statements need to be included or not.
- The bold code statement refers to the code of the case (see Section 4.2).
- The values for [xp1], [yp1], [int], [xp] and [yp] depend on the case. [xp1] and [yp1] are the problem coordinates of the begin point of the curve, taken to be the most left grid location (in many cases, this will be xp1=0, yp1=0). [xp] and [yp] are the problem coordinates of the end point of the curve, taken to be the most right grid location (in many cases, xp will be equal to the length of the 1D domain, and

yp=0). SWAN generates output at [int]–1 equidistant locations between the begin and end point (including the latter). The value for [int] should, in case a regular grid is used, be taken equal to the number of grid cells in the x–direction plus 2; which will result output to be generated in all SWAN computational grid points output is generated. On irregular grids, the value for [int] should be on: length of domain divided by the smallest grid cell size.

4.4 Format of SWAN output

Below, a list of the SWAN output for SHiVa, including the file format, is given:

➢ Curve data (file *shiva_**code**.cuv*).	SWAN table format.
➢ Block data (file *shiva_**code**.mat*).	Matlab binary format.
➢ Table data (file *shiva_**code**_loc.tab*).	SWAN table format.
➢ 1D spectra (file *shiva_**code**_loc.sp1*).	SWAN 1D–spectral format.
➢ 2D spectra (file *shiva_**code**_loc.sp2*).	SWAN 2D–spectral format.

These file formats are described in the SWAN user manual (**SWAN team 2007**), which can be downloaded from the **SWAN website**.

4.5 Format of observational data

Three standard types of observational data are considered in SHiVa. The file formats described in this section are either identical or very similar to the SWAN output format. This facilitates the use of existing functions for reading these files. In addition three formats have been defined to accommodate observations used in the ONR testbed cases. These six file formats are discussed below.

4.5.1 Integral wave parameters

Integral wave parameters derived from observations are stored in file *meas_**code**_loc.tab*. The SWAN table format is used for this file, keeping the following in mind:

- It depends on the available observed wave parameters which columns are present. Hence, this varies from case to case. At least the (x,y)–coordinates of the observational locations must be stored in the table.
- The first four lines are comment statements for the user, and are not used by SHiVa. These lines may be used for meta information (name of location, time of measurement, etc.).
- The variable names and units[6] must be the same as in the SWAN output.
- Missing data in this table is stored using the exception value –999. In SWAN results, exception values are not necessarily –999, but they are always negative (e.g. Hsig = –9). This makes it possible to distinguish exception values from actual values[7].

An example illustrating the file format is given below (*meas_**code**_loc.tab*):

```
%
%
% Observational data – integral wave parameters
%
%     Xp          Yp          Depth       Hsig        Tm_10       Tm01
%     [m]         [m]         [m]         [m]         [sec]       [sec]
%
    106514.     587986.     31.4861     8.12573     11.7525     10.4109
      9963.     447601.     37.2988     7.32000     10.6899      9.2793
     -7797.     380645.     17.5831     5.70961     10.6899      8.8744
```

[6] This includes the applied convention for the wave direction: nautical or Cartesian.

[7] Note that it may occur that an x– or y–coordinate, due to the location of the origin, has the value –999.0. In this and only this case, this is not an exception value. However, this is not very likely to occur.

```
    206527.     623483.     24.7256     7.69968     12.9208   -999.0000
     65344.     507662.     27.4365  -999.00000   -999.0000      9.8763
     21564.     429871.     25.1432     1.42156   -999.0000     10.0012
     32751.     331980.   -999.0000  -999.00000   -999.0000   -999.0000
```

4.5.2 1D spectra

1D spectra derived from observations are stored in files *meas_**code**_loc*.sp1* (the meaning of * will become clear in Section 4.6). In principle, the SWAN 1D–spectral file format suffices. However, this file format is rather complicated in the sense that manual conversion of observed spectra to this file format is error prone. The following file format, which is simpler and less error prone, is used instead. It is given in the form of an example (*meas_**code**_loc*.sp1)*:

```
%
%
% Observational data – 1D spectra
%
%     Xp          Yp
%     [m]         [m]
%
    106514.0     587986.0
%
%     FREQ        VADENS           DIR              DSPR
%     [Hz]        [m2/Hz]          [degr]           [degr]
%
    0.0300       0.7418E-27      -9.9900E+03      -9.9900E+03
    0.0330       0.2156E-13      -9.9900E+03      -9.9900E+03
    0.0363       0.2470E-09      -9.9900E+03      -9.9900E+03
       |            |                |                |
       |            |                |                |
    0.9096       0.5772E-02      -9.9900E+03      -9.9900E+03
    1.0000       0.3265E-02      -9.9900E+03      -9.9900E+03
```

Remarks:

- The first four lines are comment statements for the user, and are not used by SHiVa. These lines may be used for meta information (name of location, time of measurement, etc.).
- The coordinates of the observational location are stored on the eighth line.
- The first, second, third and fourth column (from line 13 on) contain respectively the frequency, spectral density, the mean wave direction and directional spreading.
- It is the responsibility of the user to ensure that SWAN outputs the spectral density in the same units (i.e. m2/Hz (variance density) or J/(m2 Hz) (energy density)) as the given observational data.
- The same holds for the frequency (Hz or rad/s) and the mean wave direction (Cartesian or nautical).
- Missing data in this table is stored using the exception value –999. In the given example, this is the case for the wave direction and directional spreading.

4.5.3 2D spectra

2D spectra derived from observations are stored in file *meas_**code**_loc*.sp2* (the meaning of * will become clear in Section 4.6). The SWAN 2D–spectral file format is used[8]. An example is given below, where some lines have been skipped in order to save space (*meas_**code**_loc*.sp2)*:

```
SWAN   1                                Swan standard spectral file, version
$   Data produced by SWAN version 40.51
$   Project: SBW             ;  run number: 001
LOCATIONS                               locations in x-y-space
     1                                  number of locations
   106514.0000   587986.0000
AFREQ                                   absolute frequencies in Hz
    38                                  number of frequencies
    0.0300
    0.0330
```

[8] In contrast with observed 1D spectra, for which we have selected a file format different from the SWAN file format, for 2D spectra the SWAN file format is opted for. This file format is, in our view, the best way to store a rectangular matrix of values.

```
        |
....    |
    0.9096
    1.0000
NDIR                                    spectral nautical directions in degr
    36                                  number of directions
  265.0000
  255.0000
        |
        |
  -65.0000
  -75.0000
  -85.0000
QUANT
     1                                  number of quantities in table
VaDens                                  variance densities in m2/Hz/degr
m2/Hz/degr                              unit
   -0.9900E+02                          exception value
FACTOR
    0.48662204E-03
    0    0    0    0    0    0    0    0    0    0    0    0    --- line continues
    0    0    0    0    0    0    0    0    0    0    0    0    --- line continues
   41   61   65   28    7    1    0    0    0    0    0    0    --- line continues
        |                                   |                             |
        |                                   |                             |
    9   10   33   53   70  127   34    0    0    0    0    0    --- line continues
   60   78  238  351  300  242   26    0    0    0    0    0    --- line continues
        |                                   |                             |
        |                                   |                             |
    0    0    0    0    0    0    0    0    0    0    0    0    --- line continues
    0    0    0    0    0    0    0    0    0    0    0    0    --- line continues
```

Remarks:

- The file format is described in the SWAN User Manual (**SWAN team 2007**).
- The first four lines are not used by SHiVa. These lines may be used for meta information (name of location, time of measurement, etc.).

4.5.4 Wave growth curve data

To verify depth–limited wave growth and fetch–limited deep water wave growth in SWAN various expressions for wave growth curves are available. The growth curves are presented in dimensionless parameters. Three types of analytical growth curve "data" are considered below. At present, the code names starting with f011 and f021 are wave growth cases.

4.5.4.1 "Versus dimensionless fetch data"

Data from observations which use dimensionless fetch as x–axis are stored in file *meas_**code**_loc*.fch* (the meaning of * will become clear in Section 4.6). An example is given below, where some lines have been skipped in order to save space (*meas_**code**_loc*.fch)*:

```
%
%
% Observational data - dimensionless; versus fetch
%
%          X*                Etot*                   fp*
%         [-]                  [-]                   [-]
%
         1.0E8           1100.0000              0.0056
         1.8E8           1100.0000              0.0056
         3.2E8           1100.0000              0.0056
         5.5E8           1100.0000              0.0056
         9.0E8           1100.0000              0.0056
         1.0E12          1100.0000              0.0056
```

4.5.4.2 "Versus dimensionless depth" data

Data from observations which use dimensionless depth as x–axis are stored in file *meas_**code**_loc*.dpt* (the meaning of * will become clear in Section 4.6). An example is given below, where some lines have been skipped in order to save space (*meas_**code**_loc*.dpt)*:

```
%
%
% Observational data - dimensionless; versus depth
%
%         d*      Etot*        fp*
%        [-]        [-]        [-]
%
      0.0600     0.0000     0.5784
      0.0625     0.0000     0.5694
      0.0652     0.0000     0.5605
      0.0680     0.0000     0.5517
      0.0709     0.0000     0.5431
      0.0740     0.0000     0.5346
etc
```

4.5.4.3 Polygon data

Data from observations which are given as polygons are stored in file *meas_**code**_loc*.ple* or *meas_**code**_loc*.plf* , depending on whether the data relates to energy (*.ple*) or frequency (*.plf*) (the meaning of * will become clear in Section 4.6). An example is given below, where some lines have been skipped in order to save space (*meas_**code**_loc*.ple)*:

```
%
%
% Observational data - dimensionless Energy in polygonal format
%
%         X*      Etot*         X*      Etot*          X*      Etot*              X*      Etot*
%        [-]        [-]        [-]        [-]         [-]        [-]             [-]        [-]
%
     10.2000     0.0012  8200.0000     0.4250  18300.0000     1.9200  207000000.0000  1210.0000
     10.3000     0.0011  7960.0000     0.4910  19300.0000     1.5900  207000000.0000  1320.0000
     10.9000     0.0011  7230.0000     0.6190  21200.0000     1.4200  198000000.0000  1540.0000
     11.6000     0.0011  6620.0000     0.7910  24000.0000     1.4000  180000000.0000  1810.0000
     13.4000     0.0012  6420.0000     0.9020  28000.0000     1.4000  161000000.0000  2150.0000
etc
```

4.6 Relationship between the observational locations, SWAN output locations and observational data files

Suppose that the number of observational locations is equal to N_{obs} , the following must be kept in mind:

- It is not required that in all N_{obs} locations observed integral wave parameters are available[9].
- Idem for observed 1D spectra and observed 2D spectra.
- The number of observed integral wave parameters can differ per observational location.
- The frequency resolution and frequency domain for the observed 1D and 2D spectra can differ per observational location. The directional resolution (for 2D spectra only) can differ per observational location.

The consequence of the last bullet is that observed spectra should be stored in separate files, i.e. one per observational location; the file format does not allow for storing varying frequency or directional resolutions in one file. This, on its turn, has consequences for the names of the files containing observed spectra.

The items mentioned above are illustrated by means of an example.

Suppose there are seven observational locations for a validation case. These are stored in the file ***code**_loc.loc*:

```
    106514.     587986.
```

[9] In a batch process, the user may want to include an observational location where, for certain instances, no integral wave parameters are observed. Namely, it can occur that in some instances the measurement equipment in that location is out of order. Of course, validation on integral wave parameters cannot include absent data.

```
     9963.     447601.
    -7797.     380645.
   206527.     623483.
    65344.     507662.
    21564.     429871.
    32751.     331980.
```

SWAN creates table, 1D– and 2D–spectral output data in these observational locations, to be stored in files *shiva_**code**_loc.tab*, *shiva_**code**_loc.sp1* and *shiva_**code**_loc.sp2* respectively.

Suppose further that the availability of observational data for a quantity is as given in Table 4.6.

Table 4.6 Availability of observed data per location

XP	YP	DEPTH	HSIG	TM_10	TM01	1D SPECTRUM	2D SPECTRUM
106514.	587986.	Yes	Yes	Yes	Yes	Yes	Yes
9963.	447601.	Yes	Yes	Yes	Yes	Yes	No
–7797.	380645.	Yes	Yes	Yes	Yes	No	No
206527.	623483.	Yes	Yes	Yes	No	Yes	Yes
65344.	507662.	Yes	No	No	Yes	Yes	No
21564.	429871.	Yes	Yes	No	Yes	No	No
32751.	331980.	No	No	No	No	No	No

Note that the seventh observational location contains no observational data at all.

Table 4.6 leads to the file *meas_**code**_loc.tab*, its contents[10] is described in Section 4.5.1. The number of data lines as well as the order must correspond to the locations in file ***code**_loc.loc*.

According to Table 4.6, there are four observed 1D spectra: in locations 1, 2, 4 and 5. These are stored in files *meas_**code**_loc01.sp1*, *meas_**code**_loc02.sp1*, *meas_**code**_loc04.sp1* and *meas_**code**_loc05.sp1*. There are no other *meas_**code**_loc*.sp1* files than these four.

According to the table, there are two observed 2D spectra: in locations 1 and 4. These are stored in files *meas_**code**_loc01.sp2* and *meas_**code**_loc04.sp2*. There are no other *meas_**code**_loc*.sp2* files than these two.

[10] The coordinate values and the exception values are in correspondence with Table 4.6. The remaining data values are arbitrarily chosen.

B Overview of test results with version build 122 d.d. 08 01 30

In this Appendix, an overview is given of the status of the C-I test results that have been completed with version build 120 d.d. 30 01 08.

Meaning of the table entries:

Beji-Battjes bar	: Beji-Battjes bar (SWAN case l21triad in the ONR Testbed)
Friesche Zeegat	: Friesche Zeegat (SWAN case f51fries-01 in the ONR Testbed)
Windows	: Test is executed under a Windows XP operating system
Linux	: Test is executed under a Linux operating system
DUD	: Test is executed using the DUD calibration algorithm
S	: Test is executed using the Simplex calibration algorithm
P	: Test is executed using the Powell calibration algorithm
IW	: Test is executed by Ivo Wenneker; results satisfactory
MV	: Test is executed by Martin Verlaan; results satisfactory
---	: Test has not been performed yet.

The initial experiments with the test case of the Beji-Battjes bar were primarily aimed at testing the software on both Windows and LINUX platforms, and less so for the results per se. The advantage of this one-dimensional laboratory test case is its extreme efficiency in terms of SWAN computing time.

The Beji-Battjes twin experiment allows detailed comparison of the actual optimised results and the effectivity and computational efficiency of the three optimisation methods. One must realise, however, that the best (most efficient) method is often case dependent.

Table B.1 Overview and status of first tests (aims: software should work technically correct; results should be mathematically and physically acceptable) - calibration experiments **without** constraints for the calibration parameters

Beji-Battjes bar						**Friesche Zeegat**					
Windows platform			**Linux platform**			**Windows platform**			**Linux platform**		
S	**P**	**DUD**	**S**	**P**	**DUD**	**S**	**P**	**DUD**	**S**	**P**	**DUD**
IW	**IW**	**IW**	**MV**	**MV**	**MV**	**IW**	**IW**	**IW**	**---**	**---**	**---**
Beji-Battjes bar Twin experiment											
Windows platform only											
S	**P**	**DUD**									
IW	**IW**	**IW**									

Table B.2 Overview and status of the tests (aims: software should work technically correct; results should be mathematically and physically acceptable) - calibration experiments **including** constraints for the calibration parameters

Beji-Battjes bar						**Friesche Zeegat**					
Windows platform			**Linux platform**			**Windows platform**			**Linux platform**		
S	**P**	**DUD**	**S**	**P**	**DUD**	**S**	**P**	**DUD**	**S**	**P**	**DUD**
---	---	---	**MV**	**MV**	**MV**	---	---	---	---	---	---
Beji-Battjes bar Twin experiment											
Windows platform only											
S	**P**	**DUD**									
---	---	---									

Conclusions on initial Beji-Battjes bar experiments

Parameter settings

The Beji-Battjes bar testcase (l21triad) is a simple one-dimensional case for which laboratory measurements of the wave height H_{m0} and the wave periods $T_{m-1,0}$ and T_{m02} in seven positions along the flume are available. In this test case, the model parameter settings as discussed in the example in Chapters 2 and 3 are used, unless stated otherwise. For convenience, they are repeated here. Note that the selected settings are for illustration purposes only; they do not intend to be optimal calibration settings.

We select the following four parameters for calibration: $C_{ds,bot}$ (bottom friction coefficient), γ_{BJ} (breaker parameter), and α_{EB} and $f_{\max,EB}$ (both related to triads). The parameter settings in the SWAN command file are then specified as:

```
$ --- Begin of settings physical process parameters
$
GEN3
WCAP KOM cds2=2.36E-5 stpm=3.02E-3 powst=2.0 delta=0.0 powk=0.0
OFF QUAD
FRIC JONSWAP cfjon=PAR_CFJON
BREA CON alpha=1.0 gamma=PAR_GAMMA
TRIAD trfac=PAR_TRFAC cutfr=PAR_CUTFR
$
$ --- End of settings physical process parameters
```

See also the content of file ParameterState.xdf, in Figure 3.11. The default values of these parameters, as well as the standard deviation for the uncertainty in these parameters are taken the same as in Table 2.1. See also the content of files ParameterState.xda (Figure 3.10) and uncertainties_v1.2.xml (Figure 3.9). Calibration is performed by minimising the GoF in Section 2.1 in which three quantities are used:. Observed values for these three quantities are known in Stations 02 – 08, see Appendix C. We have taken the uncertainty in all the observed quantities equal to 0.1 times the observed value; see file obsUncertainties.xml (cf. Figure 3.6).

With these settings, the three calibration algorithms have been executed. As expected, on a Windows and Linux platform, identical results are obtained.

Without weak constraints – DUD

First we discuss the results without weak constraints (Table B.1). The DUD algorithm stops after 10 iterations (SWAN runs); this value was specified in the file dudAlgorithm.xml, but was not active due to mistake in settings. As the number of iterations was not explicitly specified, the default numbers of iterations is assumed (for DUD: 10).The obtained optimal model parameters are: $C_{ds,bot}$ = 0.0578; γ_{BJ} = 0.779; α_{EB} = 0.0619; $f_{\max,EB}$ = 2.53, and the GoF is 10.69. The initial values were 0.067, 0.73, 0.05 and 2.5 respectively.

Without weak constraints – Powell

The default number of iterations for the Powell method is 200. The bracketing maximum is 20. When applying the Powell algorithm, no convergence is achieved. The values for parameter γ_{BJ} drifted to the order of 10^{14} during the iterations. After 94 iterations, we killed the process. The SWAN results for this test case are very insensitive to the value for γ_{BJ}, which explains why the Powell algorithm comes up with this unrealistically large value. This case clearly demonstrates the need for weak constraints.

Without weak constraints – Simplex

The Simplex algorithm converges after 45 iterations (default value: 100). The obtained best model parameters are: $C_{ds,bot}$ = 0.0826; γ_{BJ} = 0.355; α_{EB} = 0.0609; $f_{\max,EB}$ = 2.93, and the GoF is: 9.29.

In summary, the results for the results on Windows without constraints are as follows:

	Initial values	DUD	Powell	Simplex		
$C_{ds,bot}$	0.067	0.0578		0.0826		
γ_{BJ}	0.73	0.779	10^{14}	0.355		
α_{EB}	0.05	0.0619		0.0609		
$f_{\max,EB}$	2.5	2.53		2.93		
deault iter		10	200	100		
No iter		10	94	45		
final GoF		10.69	(user abort)	9.29		

Twin experiment without weak constraints – description

Next we discuss a twin experiment. A twin experiment is a test of the calibration software in which the optimal model parameters are known beforehand. Synthetic observational data is created by the numerical model (here: SWAN) with a user-defined set of model parameters. The calibration algorithm then should identify this set of parameters as the optimal set, since the corresponding GoF is zero. Of course, the calibration algorithm should be started with another set of parameter values to create a non-trivial search. The following user-defined ‘optimal’ model parameters are taken: $C_{ds,bot}$ = 0.05; γ_{BJ} = 0.7; α_{EB} = 0.06; $f_{\max,EB}$ = 3.0.

The same initial parameter values as above are used. Calibration is performed using the same three observed quantities as above (H_{m0}, $T_{m-1,0}$ and T_{m02}).

Twin experiment without weak constraints – Simplex

We discuss a twin experiment using the Simplex algorithm. After 47 iterations, the Simplex algorithm has converged with the following optimal model parameters: $C_{ds,bot}$ = 0.0729; γ_{BJ} = 0.708; α_{EB} = 0.0590; $f_{\max,EB}$ = 3.10. The GoF is 0.0521. From this result, we deduce that for three of the four parameters a good agreement is obtained. Only for the bottom friction coefficient, the agreement is less acceptable. This is understandable from a physical interpretation: for this test case the influence of bottom friction is rather small, which makes the results rather insensitive for even relatively large variations in the bottom friction. Related to this, we may state that the bottom friction coefficient is probably not an appropriate calibration parameter in this testcase. We have repeated the twin experiment, but now with fixing $C_{ds,bot}$ to 0.05. After 56 iterations, the Simplex algorithm has converged with the following optimal model parameters: γ_{BJ} = 0.554; α_{EB} = 0.0590; $f_{\max,EB}$ = 3.04. The GoF is 0.00634. From this result, we learn that the results are also rather insensitive to the wave breaking parameter γ_{BJ}. This is also according to expectations (as already observed above), since wave breaking hardly occurred in the experiment (maximum measured wave height is 0.027m over a minimum water depth of 0.1m). (The good correspondence for γ_{BJ} in the experiment with four calibration parameters is probably due to coincidence. Due to the insensitivity of the results for large changes in γ_{BJ}, the obtained value for γ_{BJ} is in fact rather arbitrary). We have repeated again the twin experiment, now fixing γ_{BJ} to 0.7. After 41 iterations, the Simplex algorithm has converged with the following optimal model parameters: α_{EB} = 0.05967; $f_{\max,EB}$ = 2.9978. The GoF is 7.00E–4. This demonstrates that the twin experiment with the Simplex algorithm without weak constraint converges successfully.

Twin experiment without weak constraints – Dud

We have repeated the twin experiment with the Dud algorithm. Given the conclusions above, we have restricted ourselves to calibration of the only two relevant parameters: α_{EB} and $f_{\max,EB}$. For the rest, the same settings as above are applied. After 11 iterations, the Dud algorithm has converged with the following optimal model parameters: α_{EB} = 0.05724; $f_{\max,EB}$ = 3.0585. The GoF is 0.0295. This demonstrates that the twin experiment with the Dud algorithm without weak constraint was performed successfully. The case will be rerun with different settings for the stopping criterion in the algorithm, to investigate the influence of the stopping criterion on the final result. It is expected that the results will then converge further. These additional simulations will be reported in the next release of the User Manual.

Twin experiment without weak constraints – Powell

We have repeated the twin experiment with the Powell algorithm. As for the Dud algorithm, we have restricted ourselves to calibration of the only two relevant parameters: α_{EB} and $f_{\max,EB}$. For the rest, the same settings as above are applied. After 101 iterations, the Powell algorithm has finished with the following optimal model parameters: α_{EB} = 0.09456; $f_{\max,EB}$ = 2.5379. The GoF is 0.830. This value for the minimal GoF is much larger than the values obtained with the other two algorithms. This is possibly related to the fact that the Powell algorithm is rather sensitive for not well-posed problems, with strong differences in the sensitivity of the GoF for variation of the individual parameters. This was already seen in the drift of γ_{BJ} in the first test with Powell. The conclusion is that Powell is best used in combination with specification of constraints. Another cause of the differences may be the optimisation in a different (local) minimum. This aspect will be investigated further and reported in the next release of the User Manual.

In summary, the results for the twin experiment on Windows without constraints are:

	Initial values	Optimal values	Simplex	Simplex	Simplex	DUD	Powell
No params			4	3	2	2	2
$C_{ds,bot}$	0.067	0.0500	0.0729	optimal	optimal	optimal	optimal
γ_{BJ}	0.73	0.7000	0.708	0.554	optimal	optimal	optimal
α_{EB}	0.05	0.0600	0.0590	0.0590	0.05967	0.05724	0.09456
$f_{\max,EB}$	2.5	3.0000	3.10	3.04	2.9978	3.0585	2.5379
deault iter			100	100	100	10	200
No iter			47	56	41	11	101
final GoF			0.0521	0.00634	0.0007	0.0295	0.830

For the present Twin experiment, use of the Simplex algorithm provides the best results. This preference for Simplex can not be generalised, though, as for other cases one of the other methods may give better results..

Experiments with weak constraints

Experiments with constraints on the maximum allowed variation of the parameters from their initial values were conducted on the LINUX platform. The results were compared with the results on LINUX for the unconstrained problem. They showed the influence of the constraining. It is concluded that the constraint mechanism works correctly. Detailed results will be presented in the next release of the User Manual.

Conclusions on Friesche Zeegat

In the Friesche Zeegat testcase (f51fries), the following parameter settings are specified: C_{nl4} (quadruplets), $C_{ds,bot}$ (bottom friction coefficient), and γ_{BJ} (breaker parameter).

```
$ --- Begin of settings physical process parameters
$
```

```
GEN3
WCAP KOM cds2=2.36E-5 stpm=3.02E-3 powst=2.0 delta=0.0 powk=0.0
QUAD iquad=2 lambda=0.25 Cnl4=PAR_CNL4
LIMITER ursell=10 qb=1
FRIC JONSWAP cfjon=PAR_CFJON
BREA CON alpha=1.0 gamma=PAR_GAMMA
TRIAD trfac=0.05 cutfr=2.5
$
$ --- End of settings physical process parameters
```

See also the content of file ParameterState.xdf, in Figure 3.11. The default values of these parameters, as well as the standard deviation for the uncertainty in these parameters are taken the same as in Table 2.1. See also the content of files ParameterState.xda (Figure 3.10) and uncertainties_v1.2.xml (Figure 3.9). Calibration is performed using the GoF (Section 2.1) in which only the wave height H_{m0} is compared with values derived from observations. We have taken the uncertainty in all the observed quantities equal to 0.1 times the observed value; see file obsUncertainties.xml (cf. Figure 3.6).

The objective of the first, preliminary runs is to confirm whether the calibration instrument is technically working for other cases than just the Beji-Battjes case.

Therefore, the three calibration algorithms were started and continued for a number of iterations, to confirm whether the optimization works in technical sense (no system crashes). After a few iterations (SWAN runs) the process was interrupted. Inspection of the results for this limited number of iterations showed that all expected output files were generated and that the SWAN parameters started to converge. The full simulations and the detailed optimization results of these lengthy optimization simulations will be reported in the next release of the User Manual.

C Description of the calibration testcases

At present, two calibration cases are studied: the Beji-Battjes case (l21triad) and the Friesche Zeegat case (f51fries – case 1). These two cases originate from the ONR Testbed. The descriptions of the two testcases, taken from the ONR Testbed manual, are included below.

C.1 L21: triads

Purpose

The purpose of this test is to verify the LTA method for the computation of the triad wave-wave interactions.

Situation

To test the model capability to reproduce triad interactions the laboratory flume experiment of Beji and Battjes (1993) is used. In their experiment, waves propagate from intermediate water depth over a submerged bar. The still-water depth varies between 0.4 m in the deep section and 0.1 m above the elevated bottom, see Fig. B1. The up- and down-wave bottom slopes of the submerged bar are 1:20 and 1:10 respectively. A one-dimensional spectrum, as observed by Beji and Battjes (1993) at station 01, is available as up-wave boundary condition. Ambient currents and wind are absent.

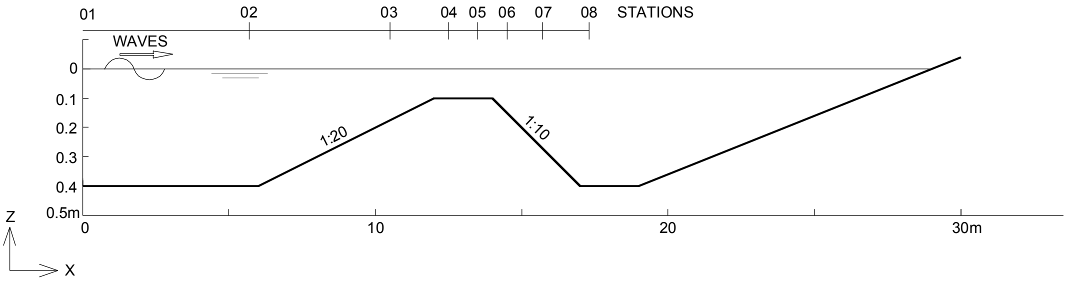


Fig. B1: Bathymetry of laboratory experiment of Beji and Battjes (1993)

Comparison

A comparison is made with observations of Beji and Battjes (1993) for energy density spectra and the significant wave height H_{m0} and mean wave period T_{m01}.

References

Beji, S. and J.A. Battjes, 1993: Experimental investigation of wave propagation over a bar, *Coastal Engineering,* 19, 151-162

Acknowledgements

Data courtesy of Delft University of Technology, Delft, The Netherlands.

C.2 F51: Friesche Zeegat (the Netherlands)

Purpose

The purpose of this test is to verify the wave model in a complex bathymetry in the field with tidal currents.

Situation

The Friesche Zeegat is located between the islands of Ameland and Schiermonnikoog in the North of the Netherlands. The bathymetry, in the computational domain of 15 km × 25 km, is rather complex, with the water depth varying from about 2 m over the shoals at high tide to about 15 m in the tidal channels (see Fig. B2). As the waves penetrate from the North Sea through the tidal gap, they refract out of the channels, across the shoals of the tidal flats. Behind the barrier islands, the waves completely reverse their direction due to refraction and waves are regenerated by the local wind. It appears that the offshore waves (at least for the case considered here) hardly penetrate into the interior region because of the strong filtering effect of the shallow flats in the centre of the tidal gap. The current velocities and water levels that are used in the computations have been obtained with the WAQUA circulation model (Les, 1996). The wind velocity and direction have been recorded at the observation station 'Huibertgat', located north of Schiermonnikoog (not shown in Fig. B2).

The geographical situation of the Friesche Zeegat was selected, because observations are available of conditions with tidal currents (maximum speed of about 1 m/s). These currents were computed with a fair degree of detail. Three cases are being considered. Time, wind speed and direction are shown in Table B1. Water levels and currents are available in digital format in the testcase directory.

Case nr.	Date/Time	Wind (U_{10}) [m/s]	Direction [°]
01	09-10-1992 / 05:00 hrs	11.5	310
02	09-10-1992 / 09:00 hrs	10	280
03	09-10-1992 / 11:00 hrs	11.5	290

Table B1: Physical parameters for case F51. Wind direction according to nautical convention. Time in UTC.

These specific cases have been selected, because (a) at these times, relatively high waves were observed (significant wave height about 3 m), generated by a storm in the northern North Sea, (b) during the period of these observations the wind speed was nearly constant, (c) the frequency spectrum was uni-modal and (d) both tidal currents and water levels were measured (to verify hydro-dynamic computation). Case F51-01 represents a case with a strong flood current while case F51-02 and F51-03 represent a high water (with practically no current) and an ebb current case respectively.

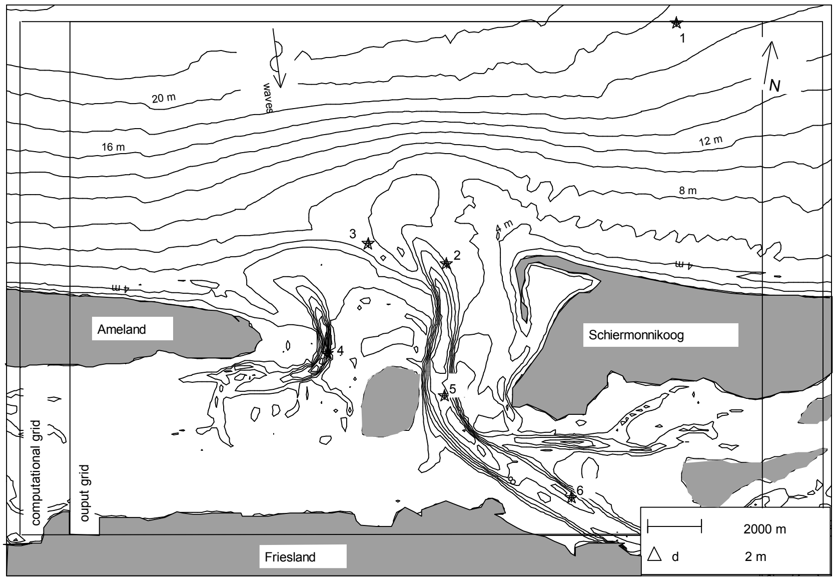


Fig. B2: Bathymetry of the Friesche Zeegat (the Netherlands) with the locations of the observation stations

The observation stations: (1) SON, (2) NWG, (3) OWG, (4) PIG, (5) ENG and (6) RHO.

Comparison

Comparisons are made for energy density spectra, significant wave height H_{m0} and mean wave period T_{m01}. (e.g. Dunsbergen, 1995b)

References

Dunsbergen, D.W., 1995b: Verification set Friesche Zeegat -October 1, 1992- November 17, 1992-, Rep. RIKZ-95.035, Ministry of Transport, Public works and Water Management, Den Haag, The Netherlands

Les, B.A.J., 1996: Flow computations in the Friesche Zeegat (Stromingsberekeningen in het Friesche Zeegat, in Dutch) M.Sc.-thesis, Delft University of Technology, Department of Civil Engineering, The Netherlands

Acknowledgements

Data courtesy: Dutch Ministry of Public Works and Coastal Management (RIKZ), Den Haag, the Netherlands.

D Proposed further functionality and testing (next phase of the present project)

Based on present status of the software (version build 122 d.d. 08 01 30), testing and documentation (User Manual), several activities are formulated to enhance the functionality and complete further testing during the next phase of the project in 2008:

- Inclusion of the factor 1/2 in the implemented expression for the Goodness-of-Fit in view of MLH theory and determination of accuracy bands around the parameters
- Include the quantity 'mean direction' in the Goodness-of-Fit expression, ensuring that directional periodicity is taken into account properly.
- Completion of the adjustments in the DUD algorithm regarding optimal use of the soft constraints in the calibration algorithms
- Include the computation of the uncertainties (Hessian / Jacobian), both the on-line as off-line option
- Determination of the uncertainty bands for the model parameters
- Introduction into the Progress Monitor of echo of key user information
- Include the options "user interrupt" and "restart"
- Complete the experiments described in Appendix B.
- Coupling of the SHiVa routines (standard SWAN statistical scores and visualisation, and access to database for SWAN cases).
- Provide output information on optimisation and convergence of calibration process in the form of tables and graphs.
- Inclusion of uncertainties in tables and graphs
- Minor adjustments to increase user-friendliness
- Provide information on the default start choices for the calibration algorithms, and how they influence the optimisation process
- Include the recommendations in Appendix E into software and User Manual insofar they have not already been included in the present version
- Update the existing version of the SWAN User Manual with information on configuring and use of the new functionalities, including convergence control
- Ensure and consolidate version control of software environment and User Manual

E Review report by external reviewer

To	Herman Gerritsen, project leader Development Calibration Instrument
From	Gerbrant van Vledder (Alkyon), external reviewer
Subject	Review Calibratie Instrument version build 122 d.d. 08 01 30
Date	30 January 2008

E.1 Introduction

On 30 January 2008 Gerbrant van Vledder carried out the review of "User Manual Calibration Instrument SWAN". In addition the calibration instrument was installed and tested on the laptop of GvV. The findings of this review are summarized in this memo.

E.2 User Manual

The User Manual was obtained as a word document. It has been read and comments were given in revision mode. The comment consisted of minor textual changes, suggestions for additional clarifications, remarks about internal inconsistencies and questions about the correctness of certain statements.

Specific remarks were made about the following items:

- A key problem in setting-up a calibration run is to select the parameters to be varied and to specify the uncertainty in each parameter. Usually, such information is obtained from separate sensitivity analyses. Suggestions were made how to obtain this information and why it is of interested to leave out a certain parameter because it speeds up the calibration process.
- The C-I uses many specific files with a unique syntax. The user does not need to learn the details of these files, but only needs to know how to modify these files. Text suggestions were given to better warn the user how to copy with these files.
- In the basic expression for the GoF the peak period Tp is used, whereas later in the document it becomes clear that the smoothed version of the peak period Tpsmoo will be used.
- The origin of the factor ½ in the GoF. For the calibration this factor is not needed, but it is related to the estimation of confidence intervals. This functionality will be added in a future version of the C-I;
- The reason for not varying the parameter alfa in the Battjes-Janssen surf breaking formulation;
- The description of placement holder strings <#..#>;
- The deactivation and activation of SWAN commands;
- How to make comment lines in a XML file;
- The concept of soft constraints and the spreading of a parameter were not properly introduced in the main text.
- The analysis of the twin tests should be extended to be sure that the model can reproduce settings of a reference case. This can easily be done using monitoring

tools to check that the parameters drift towards the expected values and that the GoF decreases.

In subsequent discussions with Herman Gerritsen en Stef Hummel all questions were answered satisfactorily. Recommendations will be included in the work plan for 2008.

E.3 Installation of the Calibration Instrument

The installation of the C-I consisted of three steps:

1. Check if Java is installed by typing java. The result of this test indicated that java is available on this machine.
2. Copying an existing data structure on to the laptop of GvV, and
3. Modification of the path OPENDA_BINDIR in the file ..\bin\run_openda_app.bat.

Hereafter, the calibration process was started successfully using the command from the bin directory:
run_openda_app.bat d:\a\a1848\ci\tests\l21triad\swanDudOpenDaConfig.xml
The second argument refers to the type of calibration method.

After installation, the contents of the various XML files was compared with the description given in the User Manual.

From this test it was concluded that the C-I was successfully installed without any problem.

The following differences were found with respect to the user manual:

- A check of the directory structure revealed that it differed slightly with the structure given in Figure 3.1. The directory \jre was not created and the name of the directory TESTS is referred to a MODEL in this figure.
- The file Application.jpif did not need to be modified for the C-I to work. It should be checked if this is critical.

E.4 Usage of the Calibration Instrument

The C-I needs to be started from the ..\bin directory with e.g. the following command:

run_openda_app.bat d:\a\a1848\ci\tests\l21triad\swanDudOpenDaConfig.xml

This resulted in a calibration run yielding a number of subdirectories with the name WORK, WORK01, WORK02, .. The content of each of these subdirectories contained the input and output results for each subroutine.

Next, some variations of calibration methods were tried and the results of these runs were inspected.
The experiments made included unconstrained and constrained tests; the effect of constraints was as expected. The constrained experiments needed more iterations, though.

E.5 Conclusions

The User Manual has been checked for errors and readability and suggestions were given to improve its usefulness.

The Calibration Instrument could easily be installed on a clean machine

The Calibration Instrument worked in accordance with the expectations.

E.6 Recommendations

On the basis of this review the following recommendations are given for the further development of the C-I.

1. Check the user manual with respect to the need to modify the file Application.jpif. See section 3.3.1.
2. The monitoring of the calibration process should be improved such that the user can follow the development of each parameter separately. All required information is available but is too much hidden in data files that are difficult to read.
3. A pause option should be included allowing the user to modify the settings of the calibration process. This includes the modification of parameters and their spreading when needed on the basis of the monitoring by the user.
4. The intermediate modification of these settings can be achieved in a number of ways. This information is now specified in a configuration file read once at the start of a calibration run. To allow intermediate changes this file could be (re)read every iteration. Another option is to add a restart option, such that changes to the settings can be made. Even a switch between calibration methods becomes possible in this way.
5. It should be verified if the C-I checks on the value of the spreading for a parameter. If this spreading is ZERO it will not vary but the program may still generate unnecessary (zero)-variations of this parameter.
6. In setting up a new case, the user needs to copy a complete data structure and modify the names for the specific case. As described in section 3.4.3 this can be done by replacing the string "l2ltriad" in to the new name. It would be better have a kind of master directory structure in which the case name is coded like [CASE]. Then, the user only needs to modify this generic name into the specific case, thus avoiding that the replacement of the old name is incompletely carried out. If one string is not modified properly, it will be very difficult to detect such an omission.
7. The user manual should be extended in section 3.4.3 \config with instructions where to modify the reference to a specific SWAN executable.
8. Add instructions in the User Manual how to install the Java toolbox.
9. The use of binary output files is not recommended because these cannot be checked visually, and because the gain in processing speed is insignificant.
10. An option should be included to remove all temporary work directories. In this phase of development it is still handy to have these results, but that may be different in a future version of the C-I.
11. It is recommended to perform experiments with different starting values of the parameters to investigate the robustness of each method.