SWIVT – Technical Reference

Xi advies bv
noorderdiep 55
9521 bb nieuw–buinen
tel 06 28200233
dekker@xi–advies.nl
www.xi–advies.nl

Authors
ir Pieter J. Dekker
ir Foskea A.T. Kleissen
ing Erwin Maliepaard
dr ir Ivo Wenneker (Deltares)

Version
2.2 December 6th, 2024
3038.22

Contents

List of Tables

List of Figures

1 General overview

1.1 Motivation for a SWAN Validation & Testing application

SWAN (**S**imulating **WA**ves **N**earshore) plays a key role in many coastal climate studies and in the computation of the Hydraulic Boundary Conditions to assess the required level of protection of the Dutch primary coastal structures. Therefore, quality assessment of SWAN in the form of validation is important. Validation of a numerical model such as SWAN requires comparison of a large number of simulation results against objective data sets. These sets consist of high–quality wave data (integral wave parameters, 1D and 2D spectra) obtained by analytical means or by means of observations in the laboratory or in the field. Comparing simulation results against objective data is done quantitatively (tables, statistics, etc.) and qualitatively (figures, etc.). Validation of a complex and broadly applicable model such as SWAN is a time–consuming task. This is caused by the large amount of model runs and related post–processing tasks that need to be executed. Fortunately, many steps in a validation process can be automated largely. This reduces the amount of human workload significantly.

The need for an efficient and flexible validation tool has led to the development of SWIVT (**SW**an **I**nstrument for **V**alidation and **T**esting).

1.2 Objectives of SWIVT

SWAN is a third–generation wave model that computes random, short–crested wind–generated waves in coastal regions and inland waters. The purpose of SWIVT is to validate SWAN in stationary and nonstationary mode in a flexible, efficient and effective way. This is achieved by offering the possibility to compare SWAN simulation results against observed data as well as against other simulations results (for example obtained with another version of SWAN or with different SWAN model settings). SWIVT includes validation cases, which are all taken from well–documented laboratory and field data and from analytical solutions. SWIVT is flexible in the sense that it offers the user the possibility to include new validation cases, to change physical model settings and to adjust the presentation of the results. SWIVT is freeware, under LGPL conditions. SWIVT is developed for experienced SWAN users.

1.3 SWIVT cases

A SWIVT case consists of a set of SWIVT files and a set of SWAN files, which belong to one SWAN simulation. There are two types of SWIVT files, both in xml format:

- One containing information for the simulation of the physical processes from which the user can select (xml file).
- One containing information on how to present the results in graphs and/or tables (spt file).

The first set of SWIVT cases available from the SWIVT server contains the ONR Testbed cases. Please note that the SWIVT cases incorporate a small number of improvements. Strict nomenclature rules have been adopted for the case and associated filenames, they are described in the User Manual, and summarised in its Glossary Section.

1.4 Documentation

SWIVT documentation consists of five documents, all written in English:

- SWIVT Installation Guide

- SWIVT User Manual
- SWIVT Technical Reference Documentation (this document)
- SWIVT Programmers Manual
- SWIVT Management and Maintenance Manual

This document describes the Technical background of SWIVT, and includes a detailed example of a case. It should be read in conjunction with the User Manual [**Dekker et al 2024a**]. Furthermore the formulae used in the statistical scores calculations are listed in Appendix A.1 and the process of making certain output parameters dimensionless for use in the wave growth cases is described in Appendix A.2.

1.5 Possible future enhancements

A number of desired enhancements were identified at a user's meeting on March 27th, 2008, those that haven't yet been implemented are listed here:

- Meta data
  - url for hindcast report
    A url could be introduced which refers to the location of the hindcast report from which the case is derived.
- Presentation templates
  Introduce a cascading search for the presentation template to ensure always the latest version of a case-specific template comes with the case from the server, but also allowing the user to generate his/her own case-specific template without imposing the danger of overwriting this. The order should be:
    - look in the **templates** directory, containing the users own case-specified templates
    - look in the **model_io** directory for the case-specific template from the server
    - look in the **swivt** root directory for the default template.

1.6 Free software, LGPL

1.7 Acknowledgements

The development of SWIVT was part of the SBW (Strength and Loads on Water Defences) study commissioned by the RijksWaterStaat, department of the Ministry of Public Works in the Netherlands. In 2024 it is part of the project SITO-PS 2024 MAD08 – Hydraulica programmatuur.

2 Directory structure

In the SWIVT documentation, it is the assumed that the SWIVT root directory is a subdirectory of the **projects** directory. The directory structure may then be as follows (the contents of the sessions directory depends on the chosen sessions and cases and directories with new SWAN versions may also be available):

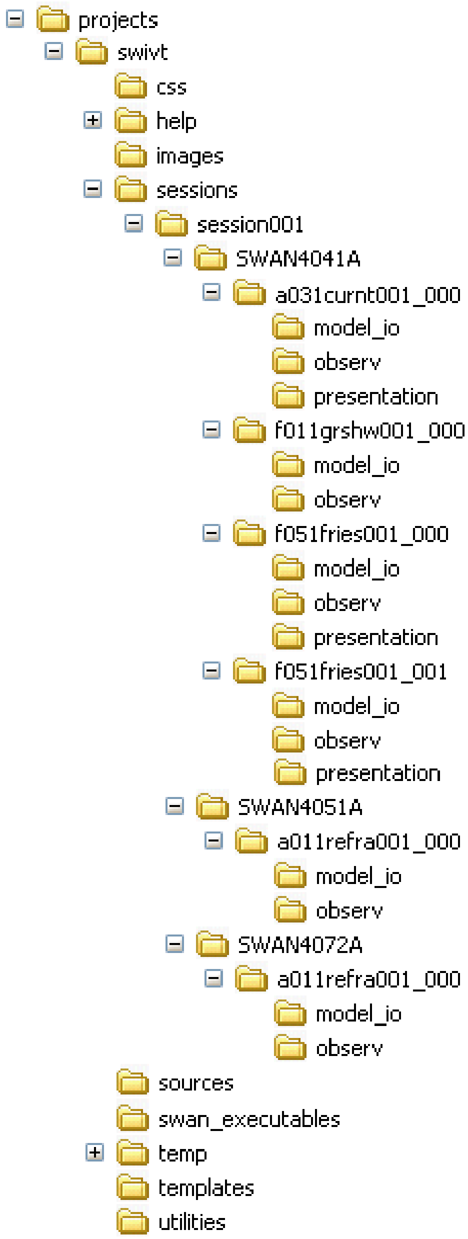

Figure 2.1 Directory structure SWIVT root

Table 2.1 SWIVT directories

DIRECTORY	DESCRIPTION
swivt	root directory, contains the default templates, the run log files and the icon

DIRECTORY	DESCRIPTION
css	contains the SWIVT stylesheet
help	contains the SWIVT help files, both in pdf and in html
images	contains the images used in the SWIVT GUI
sessions	contains the cases with the input and output data and plots/tables
sources	contains the matlab source files (if they are provided)
swan_executables	contains the different SWAN executables
temp	is empty
templates	contains the case specific presentation templates
utilities	contains the matlab source files for the utilities that are used by SWIVT

3 Input and output files: nomenclature

3.1 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.

SWIVT can only execute automated actions if certain conventions are adhered to. These conventions relate to the filenames (Section 3.2), certain parts of the SWAN command file (Section 3.2.2) and the format of the observed data files (Section 3.2.6). For reasons of completeness, the format of SWAN output is briefly mentioned in Section 3.4. The relation between observation locations, SWAN output locations and observed data files is discussed in Section 3.6.

Adhering to these conventions also ensures that SWIVT 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. Both methods are used for SWAN related files.

3.2 Filenames

Five file categories can be distinguished in SWIVT:

- **SWIVT input files**
  Files containing parameter values, physical process switches and presentation layout definitions.
- **SWIVT output files**
  SWIVT output files can optionally be generated by the SWIVT presentation module. The output format can be chosen from a list, in addition to which tables are also generated in csv format.
- **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 or more 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. SWIVT does not create SWAN input files, though SWIVT offers the option to adjust physical and numerical parameter settings, see Section 3.3.
- **SWAN output files**
  SWAN output files are created by SWAN during execution. Output files contain data to be used by SWIVT for comparison against observed data or against other SWAN output data and for presentation purposes.
- **Observed data files**
  Observed data files contain observed data. SWAN is validated against this data. Observed data files are provided by the user or are supplied with the cases that are retrieved from the server.

Files in the first, third and fourth category are stored in the **model_io** subdirectory; the observed data files are stored in the **observ** subdirectory. Files in the second category are stored in the **presentation** subdirectory and the swivt_ci_pres_set subdirectory, which are generated when required (see also Sections 4.2.3 and 4.2.5).

Filenames must be written in lower case, except for the automatically generated files *INPUT* (generated by SWIVT) and *PRINT* (generated by SWAN).

3.2.1 Code name

In SWIVT, a validation case is identified by a unique code which must be contained in all filenames related to that case. The first twelve characters of this code are built–up identical for nested and single domain cases. For nested cases additional identification is provided. The filename itself can have any length (within the limits of the operating system[1]) but must consist of alphanumeric characters (0…9, a…z), or an underscore (no minus sign!). Supplementary 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:

swivt_f061westr001_01_loc01.sp1

where

Table 3.1 Description for parts of the file code

PART	DESCRIPTION
`swivt`	additional part, explained in Section 3.2.5
`f061westr001`	the unique case code, described in Section 3.2.1.1
`01`	the nestcode, described in Section 3.2.1.2
`loc01`	additional part, explained Section 3.6
`sp1`	the extension

3.2.1.1 Single domain cases

The code name of a single domain case is built–up as:

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

where

Table 3.2 Code description single domain cases

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 = subtype identifying code (3 integers)

for example: f051fries003 or l012triad001.

3.2.1.2 Nested domain cases

The code name of a nested–domain case is built–up as:

[1] For Microsoft Windows XP the limit for the length of the filename including the full path is 260 characters

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

where the first four blocks are identical to the four blocks in the single–domain case, for clarity repeated here, and the fifth block is new:

Table 3.3 Code description nested domain cases

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 = subtype identifying code (3 integers)
5th block	ii = nestcode identifying nest run (2 integers)

Please note that it is imperative that the first run is identified by nestcode 01, the second run by nestcode 02, and so on. For example: f061westr001_**01** is the code name of the first run, and generates boundary conditions for f061westr001_**02**. Similarly f011grshw001_**05** uses boundary conditions generated by f011grshw001_**04**, and in turn generates boundary conditions for f011grshw001_**06**.

3.2.1.3 Five more identifiers: subcode, loccode, pagecode, tablecode and swanversioncode

A **subcode** is used to identify a case that is derived from a servercase, which is the case that has been downloaded from the server. The servercase is identified by subcode 000. SWIVT automatically assigns the subcodes in order.

The **loccode** is used to link the output file to a specific output location if required. Loccode is a two–digit number and derived from the order of the coordinate specification in the ***code**.loc* file.

The **pagecode** is used to identify the page in the page layout list of pages. SWIVT automatically assigns the pagecodes.

The **tablecode** is used to identify the table output file associated with an output page. SWIVT automatically assigns the tablecodes in order.

The **swanversioncode** is used to identify the SWAN version that was used to generate the export file. This value is taken from the associated case parameter.

These codes are illustrated by means of (real server case) examples:

- For the storm in the Friesche Zeegat at 09–10–1992, several validation runs can be generated. All these fall under the **casename f051fries**. In this storm, three instances (at 5:00, 8:00 and 11:00) are used for SWAN validation. For example, the **code** for the case at 11:00 is **f051fries003**. Hence, the **subtype** is **003**.
  Downloading this case from the SWIVT server provides the user with the **servercase**: **f051fries003_000**, where the last three integers (**000**) are the **subcode**. Changing eg the parameter settings leads to a new case: the subcase **f051fries003_001**. Similarly, other subcases can be generated; these will be assigned subcodes **002**, **003**, etc. For the cases with code f051fries003, the coordinates of six observation locations are supplied in file *f051fries003.loc*. As an example, the 1D spectrum measured in **location 4** (ie the fourth line in the file) is stored in a file with the name *meas_f051fries003_loc04.sp1*, where **04** is the **loccode**.

A presentation output file might be: *SWIVT_f051fries003_000_page01.ps*, on which three tables may be specified, these are then also available as: *SWIVT_f051fries003_000_page01_1.csv*, *SWIVT_f051fries003_000_page01_2.csv* and *SWIVT_f051fries003_000_page01_3.csv*. Hereby is **page01** the **pagecode** and the final **1**, **2** and **3** the respective **tablecode**.
An export output file may be *f051fries003_000_SWAN4072A.mat*, whereby **SWAN4072A** is the **swanversioncode**.

- For the storm in the Westerschelde at 20–12–1991, four validation cases are selected. Each validation case requires nested runs. All the validation cases fall under the **casename f061westr**. For the situation at 18:00 hrs, the **codes** of the two nests are **f061westr003_01** and **f061westr003_02**.
For the case with **code f061westr003_02**, the coordinates of eight observation locations are given in **file** *f061westr003_02.loc*. For instance, the measured 1D spectrum in location 8 is stored in a file with name *meas_f061westr003_02_loc08.sp1*, where **08** is the **loccode**.
A presentation output file might be: *SWIVT_f061westr003_02_000_page01.ps*, on which three tables may be specified, these are then also available as: *SWIVT_ f061westr003_02_000_page01_1.csv*, *SWIVT_ f061westr003_02_000_page01_2.csv* and *SWIVT_ f061westr003_02_000_page01_3.csv*. Hereby is **page01** the **pagecode** and the final **1**, **2** and **3** the respective **tablecode**.
An export output file may be *f061westr003_01_000_SWAN4051A.mat*, whereby **SWAN4051A** is the **swanversioncode**.

3.2.2 SWIVT input files

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

Table 3.4 SWIVT input files

FILENAME	CONTENT	COMMENT
code.*xml*	SWIVT file containing the parameter value settings and physical processes switches used for the run. Also containing the predefined default sets.	located in the **model_io** directory.
UserParamValueswithProcess.xml *UserParamValueswithoutProcess.xml*	Optional file containing user–defined parameter values and (optional) switches to switch physical processes on or off.	Is provided in the main SWIVT directory. Can be stored anywhere.
default.spt	File containing default values for presentation settings	see User Manual [**Dekker et al 2024a**]. This file is located in the main SWIVT directory.
code.*spt*	Case specific file containing preferred values for presentation settings.	optional; if available, located in the **templates** directory.

3.2.3 SWIVT output files

In the following, ***code etc*** refer to the codenames as described above.

Table 3.5 SWIVT output files

FILENAME	CONTENT	COMMENT
code_subcode_pagecode.*ext* *(ex t= ps, eps, jpg, tif, png or fig)*	SWIVT output file from the presentation module in the	located in the **presentation** directory, which will be

FILENAME	CONTENT	COMMENT
	format as specified in the page layout window.	generated when required.
code_subcode_pagecode_tablecode*.csv*	SWIVT output file, generated if a table output file is requested.	located in the **presentation** directory, which will be generated when required.
code_subcode_swanversion.mat	SWIVT output file generated by the export function in the File Menu	

3.2.4 SWAN input files

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

Table 3.6 SWAN input files

FILENAME	CONTENT	COMMENT
INPUT	SWAN input.	Is a copy of ***code****.swn*, with the placement holders replaced with the selected parameter settings. This copy is generated by the SWIVT 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 SWIVT GUI.
code*.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 SWIVT, since their names are not used in the SWIVT 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* contains the SWAN output coordinates of all observation locations for which validation is performed. Please note that SWAN–to–SWAN comparison is carried out in the locations given in file *code.loc* as well, although, strictly speaking, these locations are not observation locations, but rather 'validation locations'. This *code.loc* file is compulsory in SWIVT.
- The file *code.loc* may also contain location names, these follow the coordinates and are preceded by a '$', eg: `206767. 622696.   $SON`. These names will be used for presentation purposes when they are available.

3.2.5 SWAN output files

Table 3.7 SWAN output files

FILENAME	CONTENT	ACTION
PRINT	Log file	SWIVT checks for warnings and convergence
norm_end	Normal end of run	SWIVT checks for existence of this file
swaninit	SWAN initialisation file	Is generated by the SWAN executable. No further action required.
*swivt_**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
*swivt_**code**_loc.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
*swivt_**code**_loc.sp1*	SWAN 1D–spectral output in all observation locations given in file ***code**.loc*	Plot versus observed 1D spectra (stored in files *meas_**code**_loc*.sp1*)
*swivt_**code**_loc.sp2*	SWAN 2D–spectral output in all observation locations given in file ***code**.loc*	Plot versus observed 2D spectra (stored in files *meas_**code**_loc*.sp2*)
*swivt_**code**_loc.tab*	SWAN output of wave parameters in table format in all observation locations given in file ***code**.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 3.3.
- SWAN output files containing wave data for validation all start with '*swivt_*'.
- SWAN can only generate block data (in Matlab binary format) for 2D cases. For 1D cases, the table format is used (the *swivt **code**.tab* files).
- The curve data (*swivt_**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 need to be output.
- If the user wishes to output data in other locations than the observation locations, this data should be stored in files with names different from the ones given above.

3.2.6 Observed data files

Table 3.8 Observed 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 *swivt_**code**_loc.sp1*)
*meas_**code**_loc*.sp2*	Observed 2D spectra for the location identified by '*'	Plot versus computed 2D spectra (stored in file *swivt_**code**_loc.sp2*)

FILENAME	CONTENT	ACTION
*meas_**code**_loc.tab*	Observed integral wave parameters	Plot and statistical analysis versus computed integral wave parameters (stored in file *swivt_**code**_loc.tab*)

Remarks:
- The symbol '*' in the table represents two integer numbers, so '01', '02', '03', etc (loccode). Note that '00' is not used. These numbers refer to the position (line number) of the output location in ***code**.loc* file. In Section 3.6, this issue is elaborated on further.
- Note that labels in the filenames do not contain a reference to the name of the observation location (eg, ELD or SCW) or to the time of measurement. The reason for this is twofold:
  1) not all observation 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.
- For certain cases the **observ** directory also contains files for which the name is preceded by ONR_ (eg *ONR_meas_**code**_loc.tab*) this exception is described in Section 6.2.1.

3.3 Format of the SWAN command file

Each case has one or more SWAN command files 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 SWIVT functionalities. These conventions relate to the following issues:

- Settings of physical parameters
- Settings of numerical parameters
- Definition of SWAN output

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

3.3.1 Settings of physical parameters

The SWIVT 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:

```
$********************************************* PHYSICA
**********************************************
$
GEN3 <#model#>
<#WCAPOFF#>OFF WCAP
<#WCAP1OFF#>OFF WCAP
<#WCAPON#>WCAP KOM cds2=<#cds2#> stpm=<#stpm#> powst=<#powst#> delta=<#delta#> powk=<#powk#>
<#WCAP1ON#>WCAP WESTH cds2=<#cds2#> br=<#br#> p0=<#p0#> powst=<#powst#> powk=<#powk#>  &
<#WCAP1ON#>         nldisp=<#nldisp#> cds3=<#cds3#> powfsh=<#powfsh#>
<#QUADON#>QUAD iquad=<#iquad#> lambda=<#lambda#> Cnl4=<#Cnl4#>
<#QUADON#>LIMITER ursell=<#ursell#> qb=<#qb#>
<#QUADOFF#>OFF QUAD
<#FRICON#>FRIC JONSWAP cfjon=<#cfjon#>
<#FRICOFF#>$
<#BREAOFF#>OFF BREA
<#BREA1OFF#>OFF BREA
<#BREAON#>BREA CON alpha=<#alpha#> gamma=<#gamma#>
<#BREA1ON#>BREA WESTH alpha=<#alpha#> pown=<#pown#> bref=<#bref#> shfac=<#shfac#>
<#TRIADON#>TRIAD trfac=<#trfac#> cutfr=<#cutfr#>
<#TRIADOFF#>$
<#NUMREFRLON#> NUM REFRL frlim=<#frlim#> power=<#power#>
<#NUMREFRLOFF#>$
```

```
$
```

SWIVT will then 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.
  Please 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. A SWAN command file must then 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 lines marked in grey are only used for SWAN versions 40.72ABCDE and 40.81.

The selection of parameter settings available in SWIVT is described in detail in the Edit Case Section of the SWIVT User Manual [**Dekker et al 2024a**].

3.3.2 Settings of numerical parameters

Similar to the physical parameters, the numerical parameters can be edited with SWIVT. The following code needs to be available in the SWAN command file:

```
$ --- Begin of convergence criterion
$
<#NUMON#>NUM STOPC <#dabs#> <#drel#> <#curvat#> <#npnts#> STAT mxitst=<#mxitst#>
<#NUMOFF#>$
$
$ --- End of convergence criterion
```

SWIVT will then replace the placement holder strings <#...#> with the selected settings. If the NUM block is switched off, the SWAN default parameters will be used.

More parameters can be added if required, as can be seen in *l041curbl001.swn*, where under–relaxation (alfa=0.01) was added to the ONR case (see below). However, these extra parameters cannot be edited with SWIVT.

```
$ --- Begin of convergence criterion
$
<#NUMON#>NUM STOPC <#dabs#> <#drel#> <#curvat#> <#npnts#> STAT mxitst=<#mxitst#>  alfa=0.01
<#NUMOFF#>$
$
$ --- End of convergence criterion
```

3.3.3 Definition of SWAN output

The following output is generated for SWIVT:

- 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 observation locations.

The output parameters from SWAN, which are available through SWIVT[2], are listed and described in **Table 3.9**.

Table 3.9 Available output parameters from SWAN through SWIVT

SWIVT PARAMETER	SWAN PARAMETER	UNITS	DESCRIPTION
Xp	XP		X coordinate of observation location (in problem coordinate system)
Yp	YP		Y coordinate of observation location (in problem coordinate system)
Depth	DEP	[m]	water depth
Botlev	BOTLEV[3]	[m]	bottom level with respect to reference level
Watlev	WATLEV[3]	[m]	water level with respect to reference level
Hsig	HS	[m]	Significant wave height Hm0
RTpeak	RTP	[s]	peak period of the variance density spectrum (relative frequency spectrum)
Tp	TPEAK	[s]	peak period
Tm_10	TMM10	[s]	mean wave period $T_{m-1,0}$
Tm01	TM01	[s]	mean wave period $T_{m0,1}$
Tm02	TM02	[s]	mean wave period $T_{m0,2}$
FSpr	FSPR	[–]	normalised width of the frequency spectrum
Dir	DIR	[degrees]	mean wave direction
Dspr	DSPR	[degrees]	directional spreading of the waves
Wlen	WLENGTH	[m]	mean wavelength
TPsmoo	TPS[3]	[s]	'smoothed' peak period
dHs	DHSIGN	[m]	difference in significant wave height as computed in the last two iterations
dTm	DRTM01	[s]	difference in the mean wave period (RTM01) as computed in the last two iterations
X_Winvd; Y_Windv	WIND	[m/s]	wind velocity (vector)
X_Vel; Y_Vel.	VEL	[m/s]	current velocity (vector)
Dissip	DISSIP	[m^2/s]	energy dissipation due to bottom friction, wave breaking and whitecapping
Qb	QB	[–]	fraction of breaking waves due to depth–induced breaking
WForce_x, WForce_y	FORCE	[N/m^2]	wave induced force per unit surface area (vector)

[2] Please note that more output parameters are available from SWAN, the given selection was chosen for SWIVT v1.0 and may be expanded in future.

[3] WATLEV, BOTLEV and TPS are only available from SWAN version 40.51 and onwards

SWIVT PARAMETER	SWAN PARAMETER	UNITS	DESCRIPTION
Ubot	UBOT	[m/s]	the rms–value of the maxima of the orbital velocity near the bottom
Steepn	STEEP	[–]	mean wave steepness
Setup	SETUP	[m]	setup due to waves

In addition to the parameters above, SWIVT calculates `Hsig/Depth` [–], which can also be used for presentation purposes.

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

```
$***************** OUTPUT REQUEST ************************
$
$ --- Begin of SWIVT output
$
$     Output block data in Matlab format
$
BLOCK 'COMPGRID' NOHEADER 'swivt_code.mat' LAYOUT 3 &
   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
$
$     Define locations for data output
$
POINTS 'POINT1' FILE   'code.loc'
CURVE  'CURVE1' [xp1] [yp1] [int] [xp] [yp]
$
$     Write output data (tables, 1D and 2D spectra) for
$     these locations
$
$     +++ Point +++
$
TABLE  'POINT1' HEAD   'swivt_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   'POINT1' SPEC1D 'swivt_f051fries001_loc.sp1'
SPEC   'POINT1' SPEC2D 'swivt_f051fries001_loc.sp2'
$
$     +++ Curve +++
$
TABLE  'CURVE1' HEAD   'swivt_code_loc.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 of SWIVT output
$
```

Remarks:

- The TABLE statement is compulsory the location coordinates in the presentation part of SWIVT are derived from the resulting output file.
- The BLOCK statement is only included for 2D cases.
- The statements containing CURVE1 are only provided when curve output is required.
- The bold code statement refers to the code of the case (see Section 3.2).
- 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 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.

3.4 Format of SWAN output

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

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

These file formats are described in the SWAN user manual (**SWAN team 2024**), which can be downloaded from the **SWAN website**.

3.5 Format of observed data

Three standard types of observed data are considered in SWIVT. 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.

3.5.1 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 observation locations must be stored in the table.
- The first four lines are comment statements for the user, and are not used by SWIVT. These lines may be used for meta information (name of location, time of measurement, etc.).
- The variable names and units[4] 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[5].
- As the Hsig/Depth [–] parameter is usually not provided in this file, this parameter is calculated by SWIVT and internally added to the list. In the situation that the Depth parameter is not available from the observation file, the Depth calculated by SWAN will be used to derive this parameter. If Hsig is missing, the parameter Hsig/Depth will also be missing.

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

```
%
%
% Observed data - integral wave parameters
%
%     Xp          Yp          Depth       Hsig        Tm_10       Tm01
%     [m]         [m]         [m]         [m]         [sec]       [sec]
%
   206767.     622696.      31.4861     8.12573     11.7525     10.4109
```

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

[5] Please 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.

```
   199959.     612484.      37.2988     7.32000     10.6899      9.2793
   196992.     612714.      17.5831     5.70961     10.6899      8.8744
   196215.     608476.      24.7256     7.69968     12.9208   -999.0000
   200738.     607693.      27.4365  -999.00000   -999.0000      9.8763
...
```

3.5.2 1D spectra

1D spectra derived from observations are stored in files *meas_**code**_loc*.sp1* (the meaning of * will become clear in Section 3.6). 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)*:

```
%
%
% Observed 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 SWIVT. These lines may be used for meta information (name of location, time of measurement, etc.).
- The coordinates of the observation 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. m^2/Hz (variance density) or J/(m^2 Hz) (energy density)) as the given observed 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.

3.5.3 2D spectra

2D spectra derived from observations are stored in file *meas_**code**_loc*.sp2* (the meaning of * will become clear in Section 3.6). The SWAN 2D–spectral file format is used[6]. An example is given below, where some lines have been skipped in order to save space:

```
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
```

[6] 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.

```
    0.0330
        |
....    |
    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 2024**).
- The first four lines are not used by SWIVT. These lines may be used for meta information (name of location, time of measurement, etc.).

3.5.4 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.

Information on the fourth row is used as legend in the plots, data on the fifth row to identify the correct column and data on the sixth row to determine the units.

3.5.4.1 "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 3.6). An example is given below:

```
%
%
% Observed data - dimensionless; versus fetch Data of Pierson and Moskowitz
% Pierson and Moskowitz
%          X*                    E*                  fpeak*
%         [-]                   [-]                   [-]
%
        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
```

3.5.4.2 "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 3.6). An example is given below, where some lines have been skipped in order to save space:

```
%
%
% Observational data - dimensionless; versus depth Converted data of Bretschneider PERLOW  PERHIG
                                                                                 :  6.26 31.31
% Bretschneider
%       Depth*            E*         fpeak*
%          [-]           [-]            [-]
%
        0.0025    0.1708E-06    0.1521E+01
        0.0027    0.1937E-06    0.1474E+01
        0.0029    0.2198E-06    0.1428E+01
        0.0032    0.2493E-06    0.1384E+01
        0.0034    0.2829E-06    0.1341E+01
        0.0037    0.3209E-06    0.1300E+01
etc
```

Please note that the text on the right of the fourth line in this example is part of the third line. The header is built from seven lines starting with %.

3.5.4.3 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 3.6). An example is given below, where some lines have been skipped in order to save space (***meas_code_loc*.ple)***:

```
%
%
% Observed data - dimensionless Energy in polygonal format
% Wilson
%        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
```

3.6 Relationship between the observation locations, SWAN output locations and observed data files

Suppose that the number of observation 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[7].
- Idem for observed 1D spectra and observed 2D spectra.
- The number of observed integral wave parameters can differ per observation location.
- The frequency resolution and frequency domain for the observed 1D and 2D spectra can differ per observation location. The directional resolution (for 2D spectra only) can differ per observation location.

The consequence of the last bullet is that observed spectra should be stored in separate files, i.e. one per observation 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.

[7] In a batch process, the user may want to include an observed 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.

The items mentioned above are illustrated by means of an fictive example. Suppose there are seven observation locations for a validation case. These are stored in the file ***code**.loc*:

```
206767. 622696.  $SON
199959. 612484.  $NWG
196992. 612714.  $OWG
196215. 608476.  $PIG
200738. 607693.  $ENG
205996. 604802.  $RHO
32751   331980
```

The location names that are to be presented on a surface plot are also defined here, as illustrated above. They need to be preceded by the $–sign. If they are omitted their location will be indicated by Station 1, Station 2 etc.

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

Suppose in addition that the availability of observed data for a quantity is as given in **Table 3.10**.

Table 3.10 Availability of observed data per location

XP	YP	DEPTH	HSIG	TM_10	TM01	1D SPECTRUM	2D SPECTRUM
206767.	622696.	Yes	Yes	Yes	Yes	Yes	Yes
199959.	612484.	Yes	Yes	Yes	Yes	Yes	No
196992.	612714.	Yes	Yes	Yes	Yes	No	No
196215.	608476.	Yes	Yes	Yes	No	Yes	Yes
200738.	607693.	Yes	No	No	Yes	Yes	No
205996.	604802.	Yes	Yes	No	Yes	No	No
32751.	331980.	No	No	No	No	No	No

Note that the seventh observation location contains no observed data at all.

Table 3.10 leads to the file *meas_**code**_loc.tab*, its contents is described in Section 3.5.1. The number of data lines as well as the order must correspond to the locations in file ***code**.loc*.

According to **Table 3.10**, 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 (*refers to the loccode).

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.

4 Detailed Case

4.1 Description

In this section a fictitious detailed example case is given. Take a laboratory case, which has been assigned the code name: `l061examp`. `l061` being the next number in line for the laboratory cases and `examp` reminding us that it is indeed an example. Assume the case is to be run with SWAN version 4072A.

4.2 Directory structure and files

The name of the local directory is arbitrarily (**l061examp001** is chosen here), but two subdirectories are compulsory: **model_io** and **observ**. These are described in the next sections. Thus:

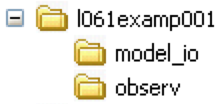

Figure 4.1 Directory structure local case

These two subdirectories are copied into a new case directory when the local case is loaded into SWIVT. If this case were added to the situation in Figure 2.1 it would become **l061examp001_001** in **session001/SWAN4072A**, even if the local directory name is different.
Please note that the subcode starts with the value 001. The subcode 000 is reserved for server cases.

The presentation template, if available, can be stored in the **templates** directory. If this template is missing, the default template will be used. Please note that it is also possible to start with the default template, adjust the settings in SWIVT and save the settings in a template file that can be stored in the **templates** directory.

4.2.1 model_io

4.2.1.1 Input

The **model_io** directory contains the SWIVT and SWAN input files, as well as the SWIVT and SWAN output files. A detailed description of the SWAN input files is given in [**SWAN team 2024**], and not repeated here. Although most filenames are optional, as long as they correspond to the ones given in the **.swn* file, it is advised to use the codename and subtype combination here too. In case of our `l061examp` example, the following input files are stored in this directory (the list is limited to the file types available for the server cases, but may be expanded):

Table 4.1 Input files in the model_io directory

FILE	DESCRIPTION
l061examp001.bnd	SWAN input file, defining boundary and initial conditions
l061examp001.bot	SWAN input file, defining bottom levels
l061examp001.cur	SWAN input file, defining current velocity
l061examp001.lev	SWAN input file, defining water levels
l061examp001.loc	SWAN input file, defining locations for output
l061examp001.swn	SWAN input file, controlling the SWAN run
l061examp001.xml	SWIVT input file, defining the default values for parameter that can be tweaked by SWIVT
l061examp001.wnd	SWAN input file, defining wind conditions

The sections from the *l061examp001.swn* file relevant to the above nomenclature are as follows (any values are arbitrarily chosen):

```
INPGRID BOTTOM  [xpinp] [ypinp] [alpinp] [mxinp] [myinp] [dxinp] [dyinp]
READINP BOTTOM  -0.01 'l061examp001.bot' 2 6 FREE
$
INPGRID CURRENT [xpinp] [ypinp] [alpinp] [mxinp] [myinp] [dxinp] [dyinp]
READINP CURRENT 1. 'l061examp001.cur' 1 18 FREE
$
INPGRID WLEVEL  [xpinp] [ypinp] [alpinp] [mxinp] [myinp] [dxinp] [dyinp]
READINP WLEVEL  1. 'l061examp001.lev' 1 17 FREE

INPGRID WX [xpinp] [ypinp] [alpinp] [mxinp] [myinp] [dxinp] [dyinp]
INPGRID WY [xpinp] [ypinp] [alpinp] [mxinp] [myinp] [dxinp] [dyinp]
READINP WIND 1. 'l061examp001.wnd' 3 0 FREE

BOU SIDE N CCW CON FILE 'l061examp001.bnd' 1

POINTS 'POINT1' FILE   'l061examp001.loc'
```

An example of a complete *.swn file (f051fries001.swn) can be found in Appendix A.3.

4.2.1.1.1 Nesting

In case of nesting, the directory may contain the following files:

Table 4.2 Input files in the model_io directory, in a case with nesting

FILE	DESCRIPTION
l061examp001_01.bot	SWAN inputfile, defining bottom levels for the first part of the run (nest 1)
l061examp001_02.bot	SWAN inputfile, defining bottom levels for the second part of the run (nest 2)
l061examp001_03.bot	SWAN inputfile, defining bottom levels for the third part of the run (nest 3)
l061examp001.cur	SWAN inputfile, defining current velocity
l061examp001.lev	SWAN inputfile, defining water levels
l061examp001_01.loc	SWAN inputfile, defining locations for output
l061examp001_02.loc	SWAN inputfile, defining locations for output
l061examp001_03.loc	SWAN inputfile, defining locations for output
l061examp001_01.swn	SWAN inputfile, controling the SWAN run
l061examp001_02.swn	SWAN inputfile, controling the SWAN run
l061examp001_03.swn	SWAN inputfile, controling the SWAN run
l061examp001.xml	SWIVT inputfile, defining the default values for parameter that can be tweaked by SWIVT

The sections from the *l061examp001_01.swn* file relevant to the above nomenclature are as follows (any values are arbitrarily chosen):

```
INPGRID BOTTOM  [xpinp] [ypinp] [alpinp] [mxinp] [myinp] [dxinp] [dyinp]
READINP BOTTOM  -0.01 'l061examp001_01.bot' 2 6 FREE
$
INPGRID CURRENT [xpinp] [ypinp] [alpinp] [mxinp] [myinp] [dxinp] [dyinp]
READINP CURRENT 1. 'l061examp001.cur' 1 18 FREE
$
INPGRID WLEVEL  [xpinp] [ypinp] [alpinp] [mxinp] [myinp] [dxinp] [dyinp]
READINP WLEVEL  1. 'l061examp001.lev' 1 17 FREE

INPGRID WX [xpinp] [ypinp] [alpinp] [mxinp] [myinp] [dxinp] [dyinp]
INPGRID WY [xpinp] [ypinp] [alpinp] [mxinp] [myinp] [dxinp] [dyinp]
READINP WIND 1. 'l061examp001.wnd' 3 0 FREE

POINTS 'POINT1' FILE   'l061examp001_01.loc'
```

The sections from the *l061examp001_02.swn* file relevant to the above nomenclature are as follows (any values are arbitrarily chosen):

```
INPGRID BOTTOM  [xpinp] [ypinp] [alpinp] [mxinp] [myinp] [dxinp] [dyinp]
READINP BOTTOM  -0.01 'l061examp001_02.bot' 2 6 FREE
$
INPGRID CURRENT [xpinp] [ypinp] [alpinp] [mxinp] [myinp] [dxinp] [dyinp]
READINP CURRENT 1. 'l061examp001.cur' 1 18 FREE
$
INPGRID WLEVEL  [xpinp] [ypinp] [alpinp] [mxinp] [myinp] [dxinp] [dyinp]
READINP WLEVEL  1. 'l061examp001.lev' 1 17 FREE

INPGRID WX [xpinp] [ypinp] [alpinp] [mxinp] [myinp] [dxinp] [dyinp]
INPGRID WY [xpinp] [ypinp] [alpinp] [mxinp] [myinp] [dxinp] [dyinp]
READINP WIND 1. 'l061examp001.wnd' 3 0 FREE

BOU NEST 'l061examp001_02.nst'
$ BOU SIDE W CCW CONS FILE 'l061examp001_02.abs'

POINTS 'POINT1' FILE   'l061examp001_02.loc'
```

And so on for more nests. In the above example, the bottom level was stored in a separate file for each nest; an analogous procedure needs to be followed if one of the other files is split over more nests.

Please note that the boundary nest file *l061examp001_02.nst* (and also *l061exampl001_02.abs*, for which the command is commented out, but is shown as a second example) is generated by SWAN, and therefore not an input file that needs to be defined by the user. The command, which ensures the file is generated, is given in Section 4.2.1.2.1.

4.2.1.2 Output

An example of the output section for *l061examp001.swn* is given below. Depending on the case, certain output cannot be produced by SWAN (eg the **.mat* file for a 1D case), it is up to the user to make the right selection. This selection does have an effect on the availability of the output graphs and tables. The output filenames are compulsory as these are used by the plotting routine (the codename is obviously case dependent).

```
$
$ --- Begin of SWIVT output
$
$     Output block data in Matlab format
$
BLOCK 'COMPGRID' NOHEADER 'swivt_l061examp001.mat' LAYOUT 3 &
   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
$
$     Define locations for data output
$
POINTS 'POINT1' FILE   'l061examp001.loc'
CURVE  'CURVE1' 90. 0. 100 900. 0.
$
$     Write output data (tables, 1D and 2D spectra) for
$     these locations
$
$     +++ Point +++
$
TABLE  'POINT1' HEAD   'swivt_l061examp001_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   'POINT1' SPEC1D 'swivt_l061examp001_loc.sp1'
SPEC   'POINT1' SPEC2D 'swivt_l061examp001_loc.sp2'
$
$     +++ Curve +++
$
TABLE  'CURVE1' HEAD   'swivt_l061examp001_loc.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 of SWIVT output
```

Table 4.3 Output files in model_io directory

FILE	DESCRIPTION
INPUT	SWAN input file, copy of *l061examp.swn*, where the placeholders have been replaced by the required values
input01.swn	SWAN input file, copy of *l061examp.swn*, where the placeholders have been replaced by the required values (identical to *INPUT* if only one **.swn* file is available)
norm_end	SWAN output file, denoting a normal end of the calculations
PRINT	SWAN output file, containing an echo of the command file and error messages
swivt_l061examp001.mat	SWAN output file, containing information required for the surface plots
swivt_l061examp001_loc.cuv	SWAN output file, containing the results produced along a curve
swivt_l061examp001_loc.sp1	SWAN output file, containing the spectral 1D results
swivt_l061examp001_loc.sp2	SWAN output file, containing the spectral 2D results
swivt_l061examp001_loc.tab	SWAN output file, containing the tabulated results for the specified locations
swaninit	SWAN output file, containing the run information, including version number

4.2.1.2.1 Nesting

In the example below the output section for *l061examp001_02.swn* is given. Again depending on the case, certain output cannot be produced by SWAN (eg the **.mat* file for a 1D case), it is up to the user to make the right selection. Two ways of generating boundary conditions for the next nest are given, using NEST and using SPEC (commented out). For the use of these two output files, please refer to Section 4.2.1.1.1. This selection does have an effect on the availability of the output graphs and tables. The output filenames are compulsory as these are used by the plotting routine.

```
$
NGRID 'NEST01' 9900. 372000. 0. 40000. 22000. 160 110
NEST  'NEST01'  'l061examp001_03.nst'
$
$POINTS 'BNDPT' 800. 0.
$SPEC   'BNDPT' SPEC2D 'l061examp001_03.abs'
$
$ --- Begin of SWIVT output
$
$     Output block data in Matlab format
$
BLOCK 'COMPGRID' NOHEADER 'swivt_l061examp001_02.mat' LAYOUT 3 &
   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
$
$     Define locations for data output
$
POINTS 'POINT1' FILE    'l061examp001_02.loc'
CURVE  'CURVE1' 90. 0. 100 900. 0.
$
$     Write output data (tables, 1D and 2D spectra) for
$     these locations
$
$     +++ Point +++
$
TABLE  'POINT1' HEAD    'swivt_l061examp001_02_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   'POINT1' SPEC1D 'swivt_l061examp001_02_loc.sp1'
SPEC   'POINT1' SPEC2D 'swivt_l061examp001_02_loc.sp2'
$
$     +++ Curve +++
$
TABLE  'CURVE1' HEAD    'swivt_l061examp001_02_loc.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 of SWIVT output
```

Table 4.4 Output files in model_io directory in a case with Nesting

FILE	DESCRIPTION
INPUT	SWAN input file, copy of *l061examp.swn*, where the placeholders have been replaced by the required values
input01.swn	SWAN input file, copy of *l061examp.swn*, where the placeholders have been replaced by the required values (identical to *INPUT* if only one **.swn* file is available)
l061examp002.abs	boundary file generated by SWAN during nest 1 calculations for input in nest 2 (only if specified in **.swn*)
l061examp003.abs	boundary file generated by SWAN during nest 2 calculations for input in nest 3 (only if specified in **.swn*)
l061examp002.nst	boundary file generated by SWAN during nest 1 calculations for input in nest 2 (only if specified in **.swn*)
l061examp003.nst	boundary file generated by SWAN during nest 2 calculations for input in nest 3 (only if specified in **.swn*)
norm_end	SWAN output file, denoting a normal end of the calculations
PRINT	SWAN output file, containing an echo of the command file and error messages
swivt_l061examp001.mat	SWAN output file, containing information required for the surface plots
swivt_l061examp001_01_loc.cuv	SWAN output file, containing the results produced along a curve, nest 1
swivt_l061examp001_02_loc.cuv	SWAN output file, containing the results produced along a curve, nest 2
swivt_l061examp001_03_loc.cuv	SWAN output file, containing the results produced along a curve, nest 3
swivt_l061examp001_01_loc.sp1	SWAN output file, containing the spectral 1D results, nest 1
swivt_l061examp001_02_loc.sp1	SWAN output file, containing the spectral 1D results, nest 2
swivt_l061examp001_03_loc.sp1	SWAN output file, containing the spectral 1D results, nest 3
swivt_l061examp001_01_loc.sp2	SWAN output file, containing the spectral 2D results, nest 1
swivt_l061examp001_02_loc.sp2	SWAN output file, containing the spectral 2D results, nest 2
swivt_l061examp001_03_loc.sp2	SWAN output file, containing the spectral 2D results, nest 3
swivt_l061examp001_01_loc.tab	SWAN output file, containing the tabulated results for the specified locations, nest 1
swivt_l061examp001_02_loc.tab	SWAN output file, containing the tabulated results for the specified locations, nest 2
swivt_l061examp001_03_loc.tab	SWAN output file, containing the tabulated results for the specified locations, nest 3
swaninit	SWAN output file, containing the run information, including version number

4.2.2 observ

The **observ** directory contains the files with observed data. These need to satisfy a number of criteria:

- the locations in the **.tab* file need to be in the same order as in the **.loc* file in the **model_io** directory
- the column titles in the **.tab* file need to be identical spelt to the ones in the *swivt_*.tab* file in the **model_io** directory, only those columns for which data is available need to be specified

- a separate **.sp** file needs to be defined for each location in the **.loc* file, unless no data is available for that location, in which case the file can be omitted. However, the location number of the files need to correspond to the order of the locations in the **.loc* file in the **model_io** directory.

Please note that for the current server cases no observed **.sp2* data is available.

In our example, the filenames are as follows (assuming 3 locations):

```
meas_l061examp001_loc01.sp1
meas_l061examp001_loc02.sp1
meas_l061examp001_loc03.sp1
meas_l061examp001_loc01.sp2
meas_l061examp001_loc02.sp2
meas_l061examp001_loc03.sp2
meas_l061examp001_loc.tab
```

4.2.2.1 Nesting

In case of nesting, it is possible to have a different number of locations in each nest (in our example, 3, 6 and 2 respectively), and the filenames will be as follows:

```
meas_l061examp001_01_loc01.sp1
meas_l061examp001_01_loc02.sp1
meas_l061examp001_01_loc03.sp1
meas_l061examp001_01_loc01.sp2
meas_l061examp001_01_loc02.sp2
meas_l061examp001_01_loc03.sp2
meas_l061examp001_01_loc.tab
meas_l061examp001_02_loc01.sp1
meas_l061examp001_02_loc02.sp1
meas_l061examp001_02_loc03.sp1
meas_l061examp001_02_loc04.sp1
meas_l061examp001_02_loc05.sp1
meas_l061examp001_02_loc06.sp1
meas_l061examp001_02_loc01.sp2
meas_l061examp001_02_loc02.sp2
meas_l061examp001_02_loc03.sp2
meas_l061examp001_02_loc04.sp2
meas_l061examp001_02_loc05.sp2
meas_l061examp001_02_loc06.sp2
meas_l061examp001_02_loc.tab
meas_l061examp001_03_loc01.sp1
meas_l061examp001_03_loc02.sp1
meas_l061examp001_03_loc01.sp2
meas_l061examp001_03_loc02.sp2
meas_l061examp001_03_loc.tab
```

4.2.3 Presentation

This subdirectory is only created when the option `Immediate print to file` is chosen in the Page layout window. The following files may have been produced:

```
SWIVT_l061examp001_001_page01.pdf
SWIVT_l061examp001_001_page02.pdf
SWIVT_l061examp001_001_page03.pdf
SWIVT_l061examp001_001_page03_1.csv
SWIVT_l061examp001_001_page03_2.csv
SWIVT_l061examp001_001_page03_3.csv
SWIVT_l061examp001_001_page04.pdf
SWIVT_l061examp001_001_page05.pdf
SWIVT_l061examp001_001_page06.pdf
SWIVT_l061examp001_001_page07.pdf
SWIVT_l061examp001_001_page07_1.csv
SWIVT_l061examp001_001_page07_2.csv
SWIVT_l061examp001_001_page08.pdf
SWIVT_l061examp001_001_page09.pdf
SWIVT_l061examp001_001_page10.pdf
SWIVT_l061examp001_001_page11.pdf
```

4.2.3.1 Nesting

In case of nesting the following files may have been produced:

```
SWIVT_l061examp001_01_001_page01.pdf
SWIVT_l061examp001_01_001_page02.pdf
SWIVT_l061examp001_01_001_page03.pdf
SWIVT_l061examp001_01_001_page03_1.csv
SWIVT_l061examp001_01_001_page03_2.csv
SWIVT_l061examp001_01_001_page03_3.csv
SWIVT_l061examp001_01_001_page04.pdf
SWIVT_l061examp001_01_001_page05.pdf
SWIVT_l061examp001_02_001_page01.pdf
SWIVT_l061examp001_02_001_page02.pdf
SWIVT_l061examp001_02_001_page02_1.csv
SWIVT_l061examp001_02_001_page02_2.csv
SWIVT_l061examp001_02_001_page03.pdf
SWIVT_l061examp001_02_001_page04.pdf
SWIVT_l061examp001_02_001_page04_1.csv
SWIVT_l061examp001_02_001_page05.pdf
SWIVT_l061examp001_02_001_page06.pdf
SWIVT_l061examp001_02_001_page07.pdf
SWIVT_l061examp001_03_001_page01.pdf
SWIVT_l061examp001_03_001_page02.pdf
SWIVT_l061examp001_03_001_page03.pdf
SWIVT_l061examp001_03_001_page04.pdf
SWIVT_l061examp001_03_001_page05.pdf
SWIVT_l061examp001_03_001_page05_1.csv
SWIVT_l061examp001_03_001_page05_2.csv
SWIVT_l061examp001_03_001_page05_3.csv
SWIVT_l061examp001_03_001_page06.pdf
SWIVT_l061examp001_03_001_page07.pdf
SWIVT_l061examp001_03_001_page07_1.csv
SWIVT_l061examp001_03_001_page07_2.csv
SWIVT_l061examp001_03_001_page08.pdf
SWIVT_l061examp001_03_001_page09.pdf
SWIVT_l061examp001_03_001_page10.pdf
SWIVT_l061examp001_03_001_page11.pdf
```

4.2.4 User defined directory (optional)

This subdirectory is used when the export option is chosen from the File menu. The following file may have been produced:

```
l061examp001_001_SWAN4072A.mat
```

4.2.4.1 Nesting

In case of nesting the following files may have been produced:

```
l061examp001_01_001_SWAN4072A.mat
l061examp001_02_001_SWAN4072A.mat
l061examp001_03_001_SWAN4072A.mat
```

4.2.5 The swivt_ci_pres_set directory

This subdirectory contains the presentation settings file *ud_swivt_ci_pres_set.mat*. It is only created when the option `Immediate print to file` is chosen in the Page layout window. The content of this directory is only used in conjunction with the Calibration Instrument developed at Deltares and is further described in its associated documentation. This directory is not relevant when the Calibration Instrument is not used.

5 Running with different SWAN versions

SWIVT has been developed to run with SWAN version 4041A onwards. One of the differences between SWAN version 4041A and 4051 onwards is the possibility to output BOTLEV, WATLEV and TPS. By default, these are output parameters in SWIVT. In order to run SWAN version 4041A these are removed from the output parameter lists in the SWAN input file: *INPUT*, which is derived from the **code**.*swn* file. This change will be notified on the Matlab Screen (m–code) or in the DOS window (Standalone code).

Therefore, the output list for running with SWAN version 4041A will be as follows:

```
$***************** OUTPUT REQUEST ************************
$
$ --- Begin of SWIVT output adapted for SWAN version 4041; only the BLOCK and TABLE statements
$     are repeated here

...

BLOCK 'COMPGRID' NOHEADER 'swivt_f051fries001.mat' LAYOUT 3 &
   XP YP DEP HS RTP TMM10 TM01 TM02 FSPR DIR DSPR   &
   WLENGTH DHSIGN DRTM01 WIND VEL DISSIP QB FORCE UBOT STEEP SETUP

...

TABLE  'POINT1' HEAD   'swivt_f051fries001_loc.tab' &
        XP YP DEP HS RTP TMM10 TM01 TM02 FSPR DIR DSPR   &
        WLENGTH DHSIGN DRTM01 WIND VEL DISSIP QB FORCE UBOT STEEP SETUP

...
$
```

6 Obtaining wave parameters by integration over a suitable frequency range

6.1 Introduction

Comparison of wave parameters as computed by SWAN with wave parameters deduced from observations is only valid if both have been obtained by identical processing of the 1D spectra. In particular, for both integration should be performed over an identical frequency range `[fmin,fmax]`. Suitable values for the lower bound `fmin` and the upper bound `fmax` often depend on the measurement equipment (wave rider, buoy, etc). Due to inherent inaccuracies in the measurement equipment, the variance density spectrum outside this range is expected to be less accurate, and should therefore be discarded in the computation of the wave parameters.

From SWAN version 40.72 onwards, SWAN can output wave parameters by integration over a user–defined integration range `[fmin,fmax]`. This is facilitated by inserting the following command in the SWAN command file:

```
QUANT HS TMM10 TM01 TM02 FMIN fmin FMAX fmax
```

where for `fmin` and `fmax` suitable values are given. The wave parameters `Hsig` (H_{m0}) and the period measures `Tm_10` ($T_{m\_1,0}$), `Tm01` (T_{m01}) and `Tm02` (T_{m02}) are computed by integration over the frequency range `[fmin,fmax]`, ie:

$$m_n = \int_{fmin}^{fmax} f^n S(f)\,df; \quad H_{m0} = 4\sqrt{m_0}; \quad T_{m-1,0} = \frac{m_{-1}}{m_0}; \quad T_{m01} = \frac{m_0}{m_1}; \quad T_{m02} = \sqrt{\frac{m_0}{m_2}}$$

As mentioned before, the wave parameters derived from a measured variance density spectrum (in files *meas_**code**_loc.tab*) should also have been obtained by integration over the same frequency range `[fmin,fmax]`.

6.2 Implications for wave parameters in SWIVT

This section describes how the measured wave parameters (ie, the wave parameters in the files *meas_**code**_loc.tab*) are obtained in SWIVT. As this is different for the validation cases that originate from the ONR Test Bed and other validation cases, these methods are given in separate subsections.

6.2.1 ONR Test Bed cases

The ONR Test Bed cases can be identified by the first four characters in their codename:

- Analytical cases: a011, a021, a031
- Laboratory cases: l011, l021, l031, l041, l051
- Field cases: f011, f021, f031, f041, f051, f061, f071, f081

In the original ONR Test Bed, both 1D wave spectra and wave parameters (table files) are given for most observation locations and in most cases. It turns out that not in all cases the wave spectra and wave parameters are consistent, ie suitable processing of the wave spectrum results in another value for the wave parameters than their corresponding values in the table files. Unfortunately, the ONR Testbed cases are rather old and up to now it has been impossible to find out how the data was processed at the time. Therefore, the cause of these deviations will probably remain unknown.

As a consequence the 1D spectra of the following ONR Test Bed cases have been reprocessed for SWIVT:

- f031harin
- f041lakgr
- f051fries
- f061westr
- f071delil
- f081norde
- l021triad
- l031setup
- l041curbl
- l051hiswa

For the remaining ONR Test Bed cases, reprocessing of the data is not relevant (eg no spectra are available or monochromatic waves are considered).

The reprocessing was carried out along the following steps:

- The 1D wave spectra as used in the ONR Test Bed were copied without modification in SWIVT.
- The original wave parameters as used in the ONR Test Bed were saved in files with the name *ONR_meas_**code**_loc.tab*. These files are stored along with the other *meas** files in the **observ** directory which is downloaded from the server once a case is added to a SWIVT session.
- The new wave parameters, obtained by processing of the 1D spectra, are stored in files with the names *meas_**code**_loc.tab*. In this file, the information on used frequency range [`fmin,fmax`] as well as whether a diagnostic tail is applied, is given as meta data in the header of the file. This frequency range corresponds (from SWAN version 40.72A onwards) to the frequency range used in SWAN to compute the wave parameters.

Therefore, for the SWIVT cases with property `SWAN40.72A`[8] (and onwards) the measured wave parameters in the file *meas_**code**_loc.tab* and the SWAN computed wave parameters in the file *swivt_**code**_loc.tab* are obtained by integration over the same frequency range.

The newly derived wave parameters are now default in SWIVT (since SWIVT uses the file *meas_**code**_loc.tab* as wave parameter file). In order to validate SWAN against the wave parameters originally used in the ONR Test Bed, the file *ONR_meas_**code**_loc.tab* needs to be renamed as *meas_**code**_loc.tab* (preferably keeping a copy of the original file).

Important remark – SWAN version 40.41A

- For validation cases with the property `SWAN40.41A`[8] the original ONR Testbed wave parameters are stored *meas_**code**_loc.tab* file. This implies that the wave parameters have not been reprocessed.

Important remark – SWAN version 40.51A

- For validation cases with the property `SWAN40.51A`[8] the newly derived wave parameters are obtained by integration of the measured wave spectrum over the entire frequency range [`fminspec,fmaxspec`] of the available measured spectrum and including – as is the case in SWAN by default – a diagnostic tail of the fourth power (f^{-4}).

[8] This property indicates which SWAN version will be used during the SWIVT run

Important remarks – SWAN version 40.72A and onwards

- For validation cases with the property SWAN40.72A[8] and onwards, the newly derived wave parameters are obtained by integration over a suitably chosen frequency range [`fmin,fmax`]. The values `fmin` and `fmax` can be found as meta data in the header of the file ***meas_code_loc.tab*** and in the SWAN command file ***code.swn*** (in the string `QUANT HS TMM10 TM01 TM02 FMIN fmin FMAX fmax`).
- The selected values for `fmin` and `fmax` are determined based on the frequency range [`flow,fhigh`] as applied in the SWAN computation and on the frequency range [`fminspec,fmaxspec`] in the available measured spectra files. Except where specified otherwise, the following rules were used:

```
fmin = max ( flow, fminspec );
fmax = min( fhigh, fmaxspec )
```

  This has resulted in the integration ranges as specified in the following table. No diagnostic tail has been applied.
- The only exception to the rules described in the previous bullet are the three cases l031setup. In order to remove the influence of a spurious low frequency part of the measured spectra, `fmin` was set at 0.25 Hz, whereas the rule above would yield `fmin` = 0.15 Hz.

Table 6.1 Applied frequency range to obtain wave parameters

CODE	FMIN [Hz]	FMAX [Hz]
f031harin[001–004]	0.0521	1.0
f041lakgr001	0.166	2.0
f041lakgr[002–003]	0.125	1.0
f051fries[001–006]	0.05209	1.0
f061westr[001–004]	0.05	0.8
f071delil[001–013]	0.0469	0.6
f081norde[001–004]	0.04545	1.0
l021triad001	0.0837	2.5
l031setup[001–003]	0.25	2.0
l041curbl001	1.0	3.85
l051hiswa001	0.317	3.125

6.2.2 Other validation cases

For the other validation cases, ie the validation cases that do not originate from the ONR Test Bed, the following conditions apply:

- There are no validation cases in SWIVT with the property SWAN40.51A[8] other than ONR Test Bed cases.
- For validation cases with the property SWAN40.72A[8] and onwards, the wave parameters are obtained by integration over a suitably chosen range [`fmin,fmax`]. The values `fmin` and `fmax` can be found as meta data in the header of the files ***meas_code_loc.tab*** and in the SWAN command file ***code.swn*** (in the string `QUANT HS TMM10 TM01 TM02 FMIN fmin FMAX fmax`).

7 Glossary

<table>
<tr><th>TERM</th><th>DESCRIPTION</th></tr>
<tr><td>case</td><td>a SWIVT case consists of a set of SWAN model input files, a SWIVT input file and optional files with observed data</td></tr>
<tr><td>casename</td><td>block 1,2 and 3 of the code (see code)</td></tr>
<tr><td>code</td><td>The code name of a case is built–up as:

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

where
Table 7.1 Code description cases
<table>
<tr><th>ITEM</th><th>DESCRIPTION</th></tr>
<tr><td>1st block</td><td>a = analytical case, l = laboratory case, f = field case</td></tr>
<tr><td>2nd block</td><td>iii = case identifying code (3 integers)</td></tr>
<tr><td>3rd block</td><td>ccccc = case identifying code (5 characters)</td></tr>
<tr><td>4th block</td><td>iii = subtype identifying code (3 integers)</td></tr>
<tr><td>5th block</td><td>ii = nestcode identifying nest run (2 integers) (optional)</td></tr>
</table>
The first three blocks constitute the casename.</td></tr>
<tr><td>loccode</td><td>code with the text "loc" followed by two integers. These integers refer to the position (line number) of the location in the code.loc file. This code is separated from its predecessors by an underscore.</td></tr>
<tr><td>nestcode</td><td>see code (5th block)</td></tr>
<tr><td>pagecode</td><td>automatically generated code with the text "page" followed by two integers. This is a number denoting the page number in the page layout list of pages. This code is separated from its predecessors by an underscore.</td></tr>
<tr><td>servercase</td><td>a case derived from the SWIVT server, identified by subcode 000 (see also subcase)</td></tr>
<tr><td>session</td><td>a SWIVT run, ie all activities that take place between starting and closing SWIVT.</td></tr>
<tr><td>subtype</td><td>see code (4th block)</td></tr>
<tr><td>subcase</td><td>a case derived from a servercase, identified by subcode other than 000.</td></tr>
<tr><td>subcode</td><td>automatically generated code of three integers, 000 identifying the base case as defined on the SWIVT server. This code is placed after the nestcode and separated from this by an underscore. The subcode is not part of code.
Please note that subcode 000 is reserved for server cases. Local cases always start with subcode 001.</td></tr>
<tr><td>swanversioncode</td><td>code starting with SWAN, followed by the version number, eg 4072A. The value is taken from the property swan_version which comes with the case from the server.</td></tr>
<tr><td>tablecode</td><td>automatically generated code of one integer, used to identify the table output file associated with an output page. This code is separated from its predecessors by an underscore.</td></tr>
</table>

8 References

Dekker et al 2007a
Dekker, P.J., F.A.T. Kleissen and E. Maliepaard. *SHiVa – Design and Prototype*. Xi advies bv. Version 1.0, 3038.01, October 24th, 2007.

Dekker et al 2007b
Dekker, P.J., F.A.T. Kleissen and E. Maliepaard. *SHiVa – Prototype User Manual*. Xi advies bv. Version 1.0, 3038.01, October 24th, 2007.

Dekker et al 2024a
Dekker, P.J., F.A.T. Kleissen, E. Maliepaard and I. Wenneker. *SWIVT – User Manual*. Xi advies bv. Version 2.2, 3038.22, 2024.

Dekker et al 2024b
Dekker, P.J., F.A.T. Kleissen and E. Maliepaard. *SWIVT – Programmer Manual*. Xi advies bv. Version 2.2, 3038.22, 2024.

Dekker et al 2024c
Dekker, P.J., F.A.T. Kleissen and E. Maliepaard. *SWIVT – Management and Maintenance Manual*. Xi advies bv. Version 2.2, 3038.22, 2024.

Dekker et al 2024d
Dekker, P.J., F.A.T. Kleissen and E. Maliepaard. *SWIVT – Installation Guide*. Xi advies bv. Version 2.2, 3038.22, 2024.

Gerritsen et al 2006
Gerritsen, H., A.J. van der Westhuysen, G. Ph. van Vledder, J. Groeneweg, G. Ruessink and 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.

Gerritsen et al 2007
Gerritsen, H., Hummel, S. and Verlaan, M.. *Calibration tool for SWAN – Detailed specification Calibration Instrument. Activity 8.3 of SBW project Waddenzee*. WL | Delft Hydraulics, Report H4918.65.

Ris et al 2002
Ris, R.C., T.T. Janssen, C. Kuiper, I. Wallast, N. Doorn, A.R. van Dongeren, L.H Holthuijsen, and N. Booij. *The ONR Test bed for Coastal and Oceanic Wave Models, Manuals*. WL|Delft Hydraulics Marine, Coastal and Industrial Infrastructure commissioned by the US Office of Naval Research, Version 2.0 December 2002

SWIVT website
https://swivt.deltares.nl/

SWAN team 2024
SWAN team, *SWAN User Manual*, Delft University of Technology. Version 41.51, SWAN Cycle III, 2024.

SWAN website
https://swanmodel.sourceforge.io

Terms of Reference
ToR (Terms of reference) – SHiVa, Bij|lage 1, Offerte aanvraag: 'Ontwerpen en prototype SWIVT (SWAN Hindcast and Validation tool)'. WL|Delft Hydraulics, MCI–24379/H4918/IW, 20 april 2007 (in Dutch).

Wenneker 2007a
Wenneker, I, *Validation Instrument for SWAN, Operational Requirements and Structure*, WL|Delft Hydraulics, commissioned by the National Institute for Coastal and Marine Management/RIKZ (Dutch Ministry of Transport, Public Works and Water Management), H4803.80, March 2007.

Wenneker 2007b
Wenneker, I, Offerte *aanvraag: 'Ontwerpen en prototype SHiVa (SWAN Hindcast and Validation tool)'*, WL|Delft Hydraulics, MCI–24379/H4918/IW, 20 april 2007 (in Dutch).

A Appendices

A.1 Definition of statistical parameters

This appendix describes the definitions for the statistical quantities that are used in the statistical post–processing. Of a given quantity (for example, the wave height or the wave period), two series are compared. Observations and model results are denoted as series x_i and y_i respectively. Both series should have the same number of results (N). Subscript i , with i being an integer between 1 and N , refers to the location. There are N observation locations with useful data. A weight w_i is assigned to location i . This weight indicates the importance ('the weight') that the user assigns to the data in location i . Though it may be standard that no preference is given to any observation location ($w_i = 1$ for all i), there are situations in which the user may want to deviate from this. Some examples are:

- the user may have less confidence in the observed data in a given location (typically, the user chooses a small value for w_i in this location);
- the user wants to give preference to certain observation locations, for example those near the coast (typically, the weights w_i associated with locations near the coast are assigned a larger value than those associated with locations further offshore);
- the user wants to avoid that a geographic non–uniform distribution of the observation locations leads to 'geographic bias' in the statistical post–processing.

In SWIVT, the convention is adopted that $0 \le w_i \le 1$. When $w_i = 0$, this implies that location i is effectively omitted from the statistical post–processing. Note that it is not allowed to have $w_i = 0$ for all i . Also, a useful definition of the *STDEV* requires that $\sum_{i=1}^{N} w_i > 1$.

In case of the statistical post–processing for two or more validation runs aggregated scores can be calculated. In SWIVT results will only be meaningful if all locations for all runs are given the weight factor 1.

A.1.1 Statistical parameters for non–directional quantities

BIAS is defined as

$$BIAS = \frac{\sum_{i=1}^{N} w_i \left(y_i - x_i \right)}{\sum_{i=1}^{N} w_i} = \frac{\sum_{i=1}^{N} w_i y_i}{\sum_{i=1}^{N} w_i} - \frac{\sum_{i=1}^{N} w_i x_i}{\sum_{i=1}^{N} w_i} \equiv \overline{y} - \overline{x} \qquad \textbf{Eq. A. 1}$$

BIAS is positive if the mean model result ($\overline{y}$) is higher than the mean observed value ($\overline{x}$).

MAE , the mean absolute error, is defined as

$$MAE = \frac{\sum_{i=1}^{N} \left| w_i \left(y_i - x_i \right) \right|}{\sum_{i=1}^{N} w_i} \qquad \textbf{Eq. A. 2}$$

STDEV, the standard deviation, is defined as

$$STDEV = \sqrt{\frac{\sum_{i=1}^{N} \left(w_i \left(y_i - x_i\right) - BIAS\right)^2}{\sum_{i=1}^{N} w_i^2 - 1}}$$ **Eq. A. 3**

Note that this quantity is only defined if $\sum_{i=1}^{N} w_i > 1$

RMS, the root mean square error, is defined by

$$RMS = \sqrt{\frac{\sum_{i=1}^{N} w_i \left(y_i - x_i\right)^2}{\sum_{i=1}^{N} w_i}}$$ **Eq. A. 4**

SCI, the scatter index, is defined as the *RMS* normalised by the by the mean of the observations, $\overline{x}$

$$SCI = \frac{RMS}{\overline{x}}$$ **Eq. A. 5**

Please note that data that is not present in the observations cannot be taken into account in the statistical measures. This is achieved by giving them a weight equal to zero.

A.1.2 Statistical parameters for directional quantities (wave direction)

BIAS is defined as

$$BIAS = \frac{\sum_{i=1}^{N} w_i \Delta\theta_i}{\sum_{i=1}^{N} w_i}$$ **Eq. A. 6**

BIAS is always positive or equal to zero

MAE, the mean absolute error, is defined as

$$MAE = \frac{\sum_{i=1}^{N} w_i \Delta\theta_i}{\sum_{i=1}^{N} w_i}$$ **Eq. A. 7**

MAE is, when wave direction is considered, equal to *BIAS*

STDEV, the standard deviation, is defined as

$$STDEV = \sqrt{\frac{\sum_{i=1}^{N} w_i \left(\Delta\theta_i - BIAS\right)^2}{\sum_{i=1}^{N} w_i - 1}} \qquad \text{Eq. A. 8}$$

Note that this quantity is only defined if $\sum_{i=1}^{N} w_i > 1$

RMS, the root mean square error, is defined by

$$RMS = \sqrt{\frac{\sum_{i=1}^{N} w_i \left(\Delta\theta_i\right)^2}{\sum_{i=1}^{N} w_i}} \qquad \text{Eq. A. 9}$$

SCI, the scatter index, is defined as the *RMS* normalised by the mean of the observations, $\bar{x}$. Since this quantity depends on the choice of the orientation of the axes, the SCI is not uniquely defined and should therefore be discarded.

Please note that data that is not present in the observations cannot be taken into account in the statistical measures. This is achieved by giving them a weight equal to zero.

A.1.3 Statistical parameters for an aggregation of SWAN validation runs

BIAS, the aggregated bias, defined as

$$BIAS = \frac{\sum_{j=1}^{N} w_j BIAS_j}{\sum_{j=1}^{N} w_j} \qquad \text{Eq. A. 10}$$

where $BIAS_j$ is the bias for validation run j and w_j the sum of the weight factors for validation run j.

MAE, the aggregated mean absolute error, is defined as

$$MAE = \frac{\sum_{j=1}^{N} w_j MAE_j}{\sum_{j=1}^{N} w_j} \qquad \text{Eq. A. 11}$$

where MAE_j is the mean absolute error for validation run j and w_j the sum of the weight factors for validation run j.

STDEV, the aggregated standard deviation, is defined as

$$STDEV = \sqrt{\frac{\sum_{j=1}^{N}((w_j - 1)STDEV_j^2)}{\sum_{j=1}^{N} w_j - 1}} \qquad \textbf{Eq. A. 12}$$

where $STDEV_j$ is the standard deviation for validation run j and w_j the sum of the weight factors for validation run j. Note that this quantity is only defined if $\sum_{j=1}^{N} w_j > 1$.

RMS, the aggregated root mean square error, is defined by

$$RMS = \sqrt{\frac{\sum_{j=1}^{N} w_j RMS_j^2}{\sum_{j=1}^{N} w_j}} \qquad \textbf{Eq. A. 13}$$

where RMS_j is the root mean square error for validation run j and w_j the sum of the weight factors for validation run j.

SCI, the aggregated scatter index, is defined as follows:

$$SCI = \frac{RMS}{\frac{\sum_{j=1}^{N} w_j \frac{RMS_j}{\max(SCI_j, \varepsilon)}}{\sum_{j=1}^{N} w_j}} \qquad \textbf{Eq. A. 14}$$

where:

SCI_j	the scatter index for validation run j
RMS_j	the root mean square error for validation run j
w_j	the sum of the weight factors for validation run j
ε	a small number (typically 2.2204 E−16)

Please note that data that is not present in the observations cannot be taken into account in the statistical measures. This is achieved by giving them a weight equal to zero.

A.1.4 Some useful relations

In this section, some useful relations between the various statistical parameters are given.

A.1.4.1 Relation between *STDEV*, *RMS* and *BIAS*

The following identity is useful in some occasions:

$$STDEV^2 = \frac{M}{M-1}\left[RMS^2 - BIAS^2\right], \qquad M = \sum_{i=1}^{N} w_i$$

Eq. A. 15

The proof for this identity is given here. Denoting $\delta_i = y_i - x_i$ and inserting the definitions for *STDEV* and *RMS* in this identity leads to:

$$STDEV^2 = \frac{\sum_{i=1}^{N} w_i \left(\delta_i - BIAS\right)^2}{M-1} = \frac{M}{M-1}\left[\frac{\sum_{i=1}^{N} w_i \, \delta_i^2}{M} - BIAS^2\right]$$

Eq. A. 16

Multiplying this by $(M-1)$ and writing out yields:

$$\sum_{i=1}^{N} w_i \left(\delta_i - BIAS\right)^2 = \sum_{i=1}^{N} w_i \, \delta_i^2 - M \cdot BIAS^2$$

Eq. A. 17

Expanding the left–hand side and application of the definitions for *BIAS* and M proves the aforementioned identity:

$$\sum_{i=1}^{N} w_i \left(\delta_i - BIAS\right)^2 = \sum_{i=1}^{N} w_i \, \delta_i^2 - 2\sum_{i=1}^{N} w_i \, \delta_i \cdot BIAS + \sum_{i=1}^{N} w_i \cdot BIAS^2 =$$
$$= \sum_{i=1}^{N} w_i \, \delta_i^2 - 2\, BIAS \cdot \sum_{i=1}^{N} w_i \, \delta_i + BIAS^2 \cdot \sum_{i=1}^{N} w_i = \sum_{i=1}^{N} w_i \, \delta_i^2 - M \cdot BIAS^2$$

Eq. A. 18

A.1.4.2 Equal weights: $w_i = w > 0$

If the user decides to assign the same weight to all locations ($w_i = w > 0$ for all i ; typically $w = 1$), then the values for the statistical quantities *BIAS*, *MAE*, *RMS* and *SCI* do not depend on the chosen value w. This does not hold for the statistical quantity *STDEV*; this quantity depends on the chosen value for w.

These statements are easy to prove. First, we need to realize that the identity $\sum_{i=1}^{N} w_i = Nw$ holds. Next, we insert this in the definitions for the statistical quantities. For brevity, we have denoted $\delta_i = y_i - x_i$ or, if wave direction is considered, $\delta_i = \Delta\theta_i$. This leads to the well–known expressions.

$$BIAS = \frac{\sum_{i=1}^{N} w_i \delta_i}{\sum_{i=1}^{N} w_i} = \frac{w\sum_{i=1}^{N} \delta_i}{Nw} = \frac{\sum_{i=1}^{N} \delta_i}{N} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - x_i\right) = \overline{y} - \overline{x}$$ **Eq. A. 19**

$$MAE = \frac{\sum_{i=1}^{N} w_i \left|\delta_i\right|}{\sum_{i=1}^{N} w_i} = \frac{w\sum_{i=1}^{N} \left|\delta_i\right|}{Nw} = \frac{\sum_{i=1}^{N} \left|\delta_i\right|}{N}$$ **Eq. A. 20**

$$STDEV = \sqrt{\frac{\sum_{i=1}^{N} w_i \left(\delta_i - BIAS\right)^2}{\sum_{i=1}^{N} w_i - 1}} = \sqrt{\frac{w \sum_{i=1}^{N}\left(\delta_i - BIAS\right)^2}{Nw - 1}}$$ **Eq. A. 21**

$$RMS = \sqrt{\frac{\sum_{i=1}^{N} w_i \delta_i^2}{\sum_{i=1}^{N} w_i}} = \sqrt{\frac{w \sum_{i=1}^{N} \delta_i^2}{Nw}} = \sqrt{\frac{\sum_{i=1}^{N} \delta_i^2}{N}}$$ **Eq. A. 22**

In statistical quantities *BIAS*, *MAE*, *RMS* and *SCI*, the weight w drops out. This is not the case for *STDEV*. This proves the statements made above.

A.2 Wave growth cases, non–standard treatment

The ONR testbed wave growth cases are non–standard cases in the way the results are presented. The associated observed data has the following properties:

- In order to cover the whole computed range, three different sets of observed data are required
- One of these sets is represented by one or two polygons.
- The data is dimensionless.
- The results are presented on a log–log scale.

In order to plot the SWAN results, SWIVT converts this data to dimensionless data according to the rules described below.

A.2.1 Shallow water

For shallow water the depth and the results are scaled with U_{10} , the wind velocity at 10 m above sea level, an input value for the cases, and the gravitational acceleration g. The non–dimensional depth d^* is calculated as:

$$d^* = \frac{gd}{(U_{10})^2} \quad \textbf{Eq. A. 23}$$

The non–dimensional total energy E^* is calculated as:

$$E^* = \frac{g^2 m_0}{(U_{10})^4} \quad \textbf{Eq. A. 24}$$

The non–dimensional peak frequency is calculated as:

$$f_p^* = \frac{f_p U_{10}}{g} \quad \textbf{Eq. A. 25}$$

where:

d	depth [m]
d^*	dimensionless depth
E^*	dimensionless total energy
f_p^*	dimensionless peak frequency
f_p	calculated peak frequency [Hz]
g	gravitational acceleration = 9.81 [m/s^2]
H_{m0}	calculated significant wave height in spectral sense (in SWAN notation *Hsig*) [m]
m_0	zero order moment over the spectral density, calculated as $\frac{(H_{m0})^2}{16}$ [m^2]
U_{10}	wind velocity at 10 m above sealevel [m/s]

A.2.2 Deep water

For deep water the fetch and results are scaled with the friction velocity U_{fr} and the gravitational acceleration g. the friction velocity is defined as

$$U_{fr} = \sqrt{c_D} * U_{10} \qquad \textbf{Eq. A. 26}$$

where, according to Wu[9]:

$$\begin{aligned} c_D &= 1.2875 * 10^{-3} & U_{10} < 7.5 \\ c_D &= \frac{0.8 + 0.065 U_{10}}{1000} & U_{10} \geq 7.5 \end{aligned} \qquad \textbf{Eq. A. 27}$$

The non–dimensional fetch X^* is calculated as:

$$X^* = \frac{gX}{(U_{fr})^2} \qquad \textbf{Eq. A. 28}$$

The non–dimensional total energy E^* is calculated as:

$$E^* = \frac{g^2 m_0}{(U_{fr})^4} \qquad \textbf{Eq. A. 29}$$

The non–dimensional peak frequency f_p^* is calculated as:

$$f_p^* = \frac{f_p U_{fr}}{g} \qquad \textbf{Eq. A. 30}$$

where:

E^*	dimensionless total energy
f_p^*	dimensionless peak frequency
f_p	calculated peak frequency [Hz]
g	gravitational acceleration = 9.81 [m/s^2]
H_{m0}	calculated significant wave height in spectral sense (in SWAN notation *Hsig*) [m]
m_0	zero order moment over the spectral density, calculated as $\frac{(H_{m0})^2}{16}$ [m^2]
U_{fr}	friction velocity [m/s] as calculated above.
X	fetch [m]
X^*	dimensionless fetch

[9] Wu, J. *Wind–stress coefficients over sea surface from breeze to hurricane*, J. Geophys. Res., 87, C12, pp 9704–9706, 1982

A.3 Example **.swn* file (SWAN 40.91 onwards)

```
$*********************** HEADING *************************
$
PROJ 'f051fries001' 'f051'
$
$ Friesche Zeegat
$ Wave propagation with flood current (9210090600)
$
$ --|------------------------------------------------------------|--
$   | This SWAN input file is part of the bench mark tests for     |
$   | SWAN. More information about this test can be found in       |
$   | an accompanied document.                                     |
$ --|------------------------------------------------------------|--
$
$******************** MODEL INPUT ***********************
$
SET DEPMIN=0.05
$
CGRID  185305. 599939. 10. 31000.  18800. 124 94 CIRCLE 36 0.05209 1. 24
$
INPGRID BOTTOM  182000. 596364. 0. 346 290 100. 100.
READINP BOTTOM  -0.01 'f051fries001.bot' 2 6 FREE
$
INPGRID CURRENT 186000. 596000 10. 310 230 100. 100.
READINP CURRENT 1. 'f051fries001.cur' 1 18 FREE
$
INPGRID WLEVEL  186000. 596000 10. 310 230 100. 100.
READINP WLEVEL  1. 'f051fries001.lev'  1 17 FREE
$
WIND 11.5 310.
$
BOU SIDE N CCW CON FILE 'f051fries001.bnd' 1
$
$ --- Begin of settings physical process parameters
$
GEN3 <#model#>
<#WCAPOFF#>OFF WCAP
<#WCAPON#>WCAP KOM cds2=<#cds2#> stpm=<#stpm#> powst=<#powst#> delta=<#delta#> powk=<#powk#>
<#QUADON#>QUAD iquad=<#iquad#> lambda=<#lambda#> Cnl4=<#Cnl4#>
<#QUADON#>LIMITER ursell=<#ursell#> qb=<#qb#>
<#QUADOFF#>OFF QUAD
<#FRICON#>FRIC JONSWAP cfjon=<#cfjon#>
<#FRICOFF#>$
<#BREAOFF#>OFF BREA
<#BREAON#>BREA CON alpha=<#alpha#> gamma=<#gamma#>
<#TRIADON#>TRIAD trfac=<#trfac#> cutfr=<#cutfr#>
<#TRIADOFF#>$
<#NUMREFRLON#> NUM REFRL frlim=<#frlim#> power=<#power#>
<#NUMREFRLOFF#>$
$
$ --- End of settings physical process parameters
$
$ --- Begin of convergence criterion
$
<#NUMON#> NUM STOPC <#dabs#> <#drel#> <#curvat#> <#npnts#> STAT mxitst=<#mxitst#>
<#NUMOFF#>$
$
$ --- End of convergence criterion
$
$***************** OUTPUT REQUEST ***********************
$
$
$
$ --- Integrate over frequency range [FMIN,FMAX] to obtain
$     wave parameters
QUANT HS TMM10 TM01 TM02 FMIN 0.05209 FMAX 1
$
$ --- Begin of SWIVT output
$
$     Output block data in Matlab format
$
BLOCK 'COMPGRID' NOHEADER 'swivt_f051fries001.mat' LAYOUT 3 &
   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
$
$     Define locations for data output
```

```
$
POINTS 'POINT1' FILE   'f051fries001.loc'
$
$     Write output data (tables, 1D and 2D spectra) for
$     these locations
$
$     +++ Point +++
$
TABLE  'POINT1' HEAD   'swivt_f051fries001_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   'POINT1' SPEC1D 'swivt_f051fries001_loc.sp1'
SPEC   'POINT1' SPEC2D 'swivt_f051fries001_loc.sp2'
$
$ --- End of SWIVT output
$
COMPUTE
STOP
```

A.4 Parameter settings – *xml* file

Cases can differ from each other by their respective properties, which will cause them to occupy different rows in the SWIVT server table. As the user can also alter the parameter settings for a certain case, a `sub_code`, which is introduced automatically in the SWIVT GUI whenever parameter settings are changed, is used to distinguish the existing and the new case. The `sub_code` is also used for the handling of sets of default settings.

In the following options are available for editing the physical parameters:

1. User–defined settings. The commissioning party or calibration instrument results may for example, define these.
2. SWAN default settings 4041A.
3. SWAN default settings 4051A.
4. SWAN default settings 4072A.
5. SWAN default settings 4072ABCDE (only for cases intended to run with SWAN4072ABCDE onwards)
6. SWAN default settings 4081 (only for cases intended to run with SWAN4081 onwards)
7. SWAN default settings 4091 (only for cases intended to run with SWAN4091 onwards)
8. SWAN default settings 4101 (only for cases intended to run with SWAN4101 onwards)
9. SWAN default settings 4110 (only for cases intended to run with SWAN4110 onwards)
10. ONR testbed settings.
11. HR2006 settings.

These settings are stored in a file in xml format with the case data set provided by the SWIVT server. The associated xsd schema file is given in Appendix A.6.
An example (*f051fries001.xml*) (text on the third line should be placed at the end of the second, without any additional spaces):

```xml
<?xml version="1.0" encoding="utf-8"?>
<ROOT xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation
                                                              ="http://swivt.deltares.nl/swivt_v1_6.xsd">
   <swivt>
      <case>
         <code>f051fries001</code>
         <settings>
            <physics>
               <user>
                  <GEN3>
                     <model>KOM</model>
                  </GEN3>
                  <WCAP>
                     <selected>on</selected>
                     <cds2>2.36e-005</cds2>
                     <stpm>0.00302</stpm>
                     <powst>2</powst>
                     <delta>0</delta>
                     <powk>0</powk>
                  </WCAP>
                  <QUAD>
                     <selected>on</selected>
                     <iquad>2</iquad>
                     <lambda>0.25</lambda>
                     <Cnl4>30000000</Cnl4>
                     <ursell>10</ursell>
                     <qb>1</qb>
                  </QUAD>
                  <FRIC>
                     <selected>on</selected>
```

```xml
                <cfjon>0.067</cfjon>
            </FRIC>
            <BREA>
                <selected>off</selected>
                <alpha>1</alpha>
                <gamma>0.73</gamma>
            </BREA>
            <TRIAD>
                <selected>on</selected>
                <trfac>0.05</trfac>
                <cutfr>2.5</cutfr>
            </TRIAD>
            <NUMREFRL>
                <selected>off</selected>
                <frlim>0.2</frlim>
                <power>2</power>
            </NUMREFRL>
        </user>
        <SWAN4072ABCDE>
            <GEN3>
                <model>KOM</model>
            </GEN3>
            <WCAP>
                <selected>on</selected>
                <cds2>2.36e-005</cds2>
                <stpm>0.00302</stpm>
                <powst>2</powst>
                <delta>0</delta>
                <powk>0</powk>
            </WCAP>
            <QUAD>
                <selected>on</selected>
                <iquad>2</iquad>
                <lambda>0.25</lambda>
                <Cnl4>30000000</Cnl4>
                <ursell>10</ursell>
                <qb>1</qb>
            </QUAD>
            <FRIC>
                <selected>on</selected>
                <cfjon>0.067</cfjon>
            </FRIC>
            <BREA>
                <selected>off</selected>
                <alpha>1</alpha>
                <gamma>0.73</gamma>
            </BREA>
            <TRIAD>
                <selected>on</selected>
                <trfac>0.05</trfac>
                <cutfr>2.5</cutfr>
            </TRIAD>
            <NUMREFRL>
                <selected>off</selected>
                <frlim>0.2</frlim>
                <power>2</power>
            </NUMREFRL>
        </SWAN4072ABCDE>
        <SWAN4081>
            <GEN3>
                <model>KOM</model>
            </GEN3>
            <WCAP>
                <selected>on</selected>
                <cds2>2.36e-005</cds2>
                <stpm>0.00302</stpm>
```

```
            <powst>2</powst>
            <delta>0</delta>
            <powk>0</powk>
        </WCAP>
        <QUAD>
            <selected>on</selected>
            <iquad>2</iquad>
            <lambda>0.25</lambda>
            <Cnl4>30000000</Cnl4>
            <ursell>10</ursell>
            <qb>1</qb>
        </QUAD>
        <FRIC>
            <selected>on</selected>
            <cfjon>0.067</cfjon>
        </FRIC>
        <BREA>
            <selected>off</selected>
            <alpha>1</alpha>
            <gamma>0.73</gamma>
        </BREA>
        <TRIAD>
            <selected>on</selected>
            <trfac>0.05</trfac>
            <cutfr>2.5</cutfr>
        </TRIAD>
        <NUMREFRL>
            <selected>off</selected>
            <frlim>0.0</frlim>
            <power>0</power>
        </NUMREFRL>
    </SWAN4081>
    <SWAN4091>
        <GEN3>
            <model>KOM</model>
        </GEN3>
        <WCAP>
            <selected>on</selected>
            <cds2>2.36e-005</cds2>
            <stpm>0.00302</stpm>
            <powst>2</powst>
            <delta>0</delta>
            <powk>1</powk>
        </WCAP>
        <QUAD>
            <selected>on</selected>
            <iquad>2</iquad>
            <lambda>0.25</lambda>
            <Cnl4>30000000</Cnl4>
            <ursell>10</ursell>
            <qb>1</qb>
        </QUAD>
        <FRIC>
            <selected>on</selected>
            <cfjon>0.067</cfjon>
        </FRIC>
        <BREA>
            <selected>on</selected>
            <alpha>1</alpha>
            <gamma>0.73</gamma>
        </BREA>
        <TRIAD>
            <selected>on</selected>
            <trfac>0.05</trfac>
            <cutfr>2.5</cutfr>
        </TRIAD>
```

```
                        <NUMREFRL>
                            <selected>off</selected>
                            <frlim>0.0</frlim>
                            <power>0</power>
                        </NUMREFRL>
                    </SWAN4091>
                    <SWAN4101>
                        <GEN3>
                            <model>KOM</model>
                        </GEN3>
                        <WCAP>
                            <selected>on</selected>
                            <cds2>2.36e-005</cds2>
                            <stpm>0.00302</stpm>
                            <powst>2</powst>
                            <delta>0</delta>
                            <powk>1</powk>
                        </WCAP>
                        <QUAD>
                            <selected>on</selected>
                            <iquad>2</iquad>
                            <lambda>0.25</lambda>
                            <Cnl4>30000000</Cnl4>
                            <ursell>10</ursell>
                            <qb>1</qb>
                        </QUAD>
                        <FRIC>
                            <selected>on</selected>
                            <cfjon>0.067</cfjon>
                        </FRIC>
                        <BREA>
                            <selected>on</selected>
                            <alpha>1</alpha>
                            <gamma>0.73</gamma>
                        </BREA>
                        <TRIAD>
                            <selected>on</selected>
                            <trfac>0.05</trfac>
                            <cutfr>2.5</cutfr>
                        </TRIAD>
                        <NUMREFRL>
                            <selected>off</selected>
                            <frlim>0.0</frlim>
                            <power>0</power>
                        </NUMREFRL>
                    </SWAN4101>
                </physics>
                <numerics>
                    <user>
                        <NUM>
                            <selected>off</selected>
                            <dabs>0</dabs>
                            <drel>0.01</drel>
                            <curvat>0.005</curvat>
                            <npnts>98</npnts>
                            <mxitst>50</mxitst>
                        </NUM>
                    </user>
                </numerics>
            </settings>
        </case>
    </swivt>
</ROOT>
```

For completeness the lines that were used for the SWAN executables v 4072ABCDE and 4081 are provided here:

```
<?xml version="1.0" encoding="utf-8"?>
<ROOT xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation
                                                        ="http://swivt.deltares.nl/swivt_v1_6.xsd">
   <swivt>
      <case>
         <code>f051fries001</code>
         <settings>
            <physics>
               <user>
…
                  <WCAP1>
                     <selected>off</selected>
                     <cds2>5e-005</cds2>
                     <br>0.00175</br>
                     <p0>4</p0>
                     <powst>0</powst>
                     <powk>0</powk>
                     <nldisp>0</nldisp>
                     <cds3>1.5</cds3>
                     <powfsh>1</powfsh>
                  </WCAP1>
…
                  <BREA1>
                     <selected>on</selected>
                     <alpha>0.98</alpha>
                     <pown>2.5</pown>
                     <bref>-1.3963</bref>
                     <shfac>500</shfac>
                  </BREA1>
…
               </user>
               <SWAN4072ABCDE>
…
                  <WCAP1>
                     <selected>off</selected>
                     <cds2>5e-005</cds2>
                     <br>0.00175</br>
                     <p0>4</p0>
                     <powst>0</powst>
                     <powk>0</powk>
                     <nldisp>0</nldisp>
                     <cds3>1.5</cds3>
                     <powfsh>1</powfsh>
                  </WCAP1>
…
                  <BREA1>
                     <selected>on</selected>
                     <alpha>0.98</alpha>
                     <pown>2.5</pown>
                     <bref>-1.3963</bref>
                     <shfac>500</shfac>
                  </BREA1>
…
               </SWAN4072ABCDE>
               <SWAN4081>
…
                  <WCAP1>
                     <selected>off</selected>
                     <cds2>5e-005</cds2>
                     <br>0.00175</br>
                     <p0>4</p0>
                     <powst>0</powst>
```

```
                    <powk>0</powk>
                    <nldisp>0</nldisp>
                    <cds3>1.5</cds3>
                    <powfsh>1</powfsh>
                </WCAP1>
…
                <BREA1>
                    <selected>on</selected>
                    <alpha>0.98</alpha>
                    <pown>2.5</pown>
                    <bref>-1.3963</bref>
                    <shfac>500</shfac>
                </BREA1>
…
            </SWAN4081>
…
        </physics>
…
      </settings>
    </case>
  </swivt>
</ROOT>
```


In future new, approved settings for a case made by a user can simply be added to this file. The settings will appear automatically in the drop down box of the Case edit window in the SWIVT GUI.

A.5 Presentation template – *spt* file

Part of an example template (*f051fries001.spt*) is given below (note that text on the third line is part of the second, without any spaces between both texts). For a complete overview of the xml tags available, please refer to the xsd file given in Appendix A.6.

```
<?xml version="1.0" encoding="utf-8"?>
<ROOT xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation
                                                  ="http://swivt.deltares.nl/swivt_v1_6.xsd">
   <swivt>      <presentation>
         <name>SWIVT - F051FRIES001</name>
         <page>
            <item>
               <page_style>111</page_style>
               <page_name>Overview of calculated vs observed parameters</page_name>
               <presentation_element>
                  <type>3</type>
                  <layout>
                     <title/>
                     <x_axis>
                        <selected>off</selected>
                        <label/>
                        <limits/>
                        <scale/>
                     </x_axis>
                     <y_axis>
                        <selected>off</selected>
                        <label/>
                        <limits/>
                        <scale/>
                     </y_axis>
                     <z_axis>
                        <selected>off</selected>
                        <label/>
                        <limits/>
                        <scale/>
                     </z_axis>
                     <colormap/>
                     <colorbar>off</colorbar>
                  </layout>
                  <parameter>
                     <item>Dir</item>
                     ...
                  </parameter>
                  <location>
                     <item>
                        <id>1</id>
                        <name>Station 1</name>
                        <selected>on</selected>
                        <weight>1</weight>
                        <style>
                           <linestyle>none</linestyle>
                           <linewidth>1</linewidth>
                           <color>0 0 1</color>
                           <marker>none</marker>
                           <markersize>6</markersize>
                           <markeredgecolor>0 0 1</markeredgecolor>
                           <markerfacecolor>0 0 1</markerfacecolor>
                        </style>
                     </item>
                     <item>
                        <id>2</id>
                        <name>Station 2</name>
                        ...
                     </item>
                  </location>
                   <observed>on</observed>
               </presentation_element>
            </item>
            <item>
            ...
            </item>
         </page>
      </presentation>
   </swivt>
</ROOT>
```

A.6 SWIVT xsd schema

The SWIVT xsd schema can be downloaded from the **SWIVT website** (Source code).

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="root">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="swivt">
                    <xs:complexType>
                        <xs:choice>
                            <xs:sequence>
                                <xs:element name="presentation" type="presentation"/>
                            </xs:sequence>
                            <xs:sequence>
                                <xs:element name="case" type="case"/>
                            </xs:sequence>
                        </xs:choice>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:complexType name="presentation">
        <xs:sequence>
            <xs:element name="name" type="xs:string"/>
            <xs:element name="page" type="page"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="case">
        <xs:sequence>
            <xs:element name="code" type="xs:string"/>
            <xs:element name="settings" type="settings"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="settings">
        <xs:sequence>
            <xs:element name="physics" type="physics"/>
            <xs:element name="numerics" type="numerics"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="physics">
        <xs:choice>
            <xs:sequence>
                <xs:element name="user" type="setting4041A"/>
                <xs:element name="SWAN4041A" type="setting4041A"/>
                <xs:element name="SWAN4051A" type="setting4041A"/>
                <xs:element name="SWAN4072A" type="setting4041A"/>
                <xs:element name="ONR" type="setting4041A"/>
                <xs:element name="HR2006" type="setting4041A"/>
            </xs:sequence>
            <xs:sequence>
                <xs:element name="user" type="setting4072ABCDE"/>
                <xs:element name="SWAN4072ABCDE" type="setting4072ABCDE"/>
                <xs:element name="SWAN4081" type="setting4072ABCDE"/>
            </xs:sequence>
            <xs:sequence>
                <xs:element name="user" type="setting4091"/>
                <xs:element name="SWAN4091" type="setting4091"/>
                <xs:element name="SWAN4101" type="setting4091"/>
            </xs:sequence>
```

```
        </xs:choice>
    </xs:complexType>
    <xs:complexType name="numerics">
        <xs:sequence>
            <xs:element name="user">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="NUM" type="NUM"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="NUM">
        <xs:sequence>
            <xs:element name="selected" type="textbool"/>
            <xs:element name="dabs" type="xs:string"/>
            <xs:element name="drel" type="xs:string"/>
            <xs:element name="curvat" type="xs:string"/>
            <xs:element name="npnts" type="xs:string"/>
            <xs:element name="mxitst" type="xs:integer"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="setting4041A">
        <xs:sequence>
            <xs:element name="GEN3" type="GEN3"/>
            <xs:element name="WCAP" type="WCAP"/>
            <xs:element name="QUAD" type="QUAD"/>
            <xs:element name="FRIC" type="FRIC"/>
            <xs:element name="BREA" type="BREA"/>
            <xs:element name="TRIAD" type="TRIAD"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="setting4072ABCDE">
        <xs:sequence>
            <xs:element name="GEN3" type="GEN3"/>
            <xs:element name="WCAP" type="WCAP"/>
            <xs:element name="WCAP1" type="WCAP1"/>
            <xs:element name="QUAD" type="QUAD"/>
            <xs:element name="FRIC" type="FRIC"/>
            <xs:element name="BREA" type="BREA"/>
            <xs:element name="BREA1" type="BREA1"/>
            <xs:element name="TRIAD" type="TRIAD"/>
            <xs:element name="NUMREFRL" type="NUMREFRL"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="setting4091">
        <xs:sequence>
            <xs:element name="GEN3" type="GEN3"/>
            <xs:element name="WCAP" type="WCAP"/>
            <xs:element name="QUAD" type="QUAD"/>
            <xs:element name="FRIC" type="FRIC"/>
            <xs:element name="BREA" type="BREA"/>
            <xs:element name="TRIAD" type="TRIAD"/>
            <xs:element name="NUMREFRL" type="NUMREFRL"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="GEN3">
        <xs:sequence>
            <xs:element name="model" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="WCAP">
        <xs:sequence>
            <xs:element name="selected" type="textbool"/>
            <xs:element name="cds2" type="xs:string"/>
```

```
            <xs:element name="stpm" type="xs:string"/>
            <xs:element name="powst" type="xs:integer"/>
            <xs:element name="delta" type="xs:integer"/>
            <xs:element name="powk" type="xs:integer"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="WCAP1">
        <xs:sequence>
            <xs:element name="selected" type="textbool"/>
            <xs:element name="cds2" type="xs:string"/>
            <xs:element name="br" type="xs:string"/>
            <xs:element name="p0" type="xs:string"/>
            <xs:element name="powst" type="xs:string"/>
            <xs:element name="powk" type="xs:string"/>
            <xs:element name="nldisp" type="xs:integer"/>
            <xs:element name="cds3" type="xs:string"/>
            <xs:element name="powfsh" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="QUAD">
        <xs:sequence>
            <xs:element name="selected" type="textbool"/>
            <xs:element name="iquad" type="xs:integer"/>
            <xs:element name="lambda" type="xs:string"/>
            <xs:element name="Cnl4" type="xs:integer"/>
            <xs:element name="ursell" type="xs:integer"/>
            <xs:element name="qb" type="xs:integer"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="FRIC">
        <xs:sequence>
            <xs:element name="selected" type="textbool"/>
            <xs:element name="cfjon" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="BREA">
        <xs:sequence>
            <xs:element name="selected" type="textbool"/>
            <xs:element name="alpha" type="xs:integer"/>
            <xs:element name="gamma" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="BREA1">
        <xs:sequence>
            <xs:element name="selected" type="textbool"/>
            <xs:element name="alpha" type="xs:string"/>
            <xs:element name="pown" type="xs:string"/>
            <xs:element name="bref" type="xs:string"/>
            <xs:element name="shfac" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="TRIAD">
        <xs:sequence>
            <xs:element name="selected" type="textbool"/>
            <xs:element name="trfac" type="xs:string"/>
            <xs:element name="cutfr" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="NUMREFRL">
        <xs:sequence>
            <xs:element name="selected" type="textbool"/>
            <xs:element name="frlim" type="xs:string"/>
            <xs:element name="power" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="page">
```

```
        <xs:sequence>
            <xs:element name="item" type="page-item" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="page-item">
        <xs:sequence>
            <xs:element name="page_style">
                <xs:simpleType>
                    <xs:restriction base="xs:integer">
                        <xs:minInclusive value="1"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element name="page_name" type="xs:string"/>
            <xs:choice>
                <xs:sequence>
                    <xs:element name="presentation_element" type="presentation-item"/>
                </xs:sequence>
                <xs:sequence>
                    <xs:element name="presentation_element" type="presentation-items"/>
                </xs:sequence>
            </xs:choice>
            <!-- xs:element name="presentation_element" type="presentation-items"/ -->
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="presentation-items">
        <xs:sequence>
            <xs:element name="item" type="presentation-item" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="presentation-item">
        <xs:sequence>
            <xs:element name="type">
                <xs:simpleType>
                    <xs:restriction base="xs:integer">
                        <xs:minInclusive value="1"/>
                        <xs:maxInclusive value="19"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element name="layout" type="layout"/>
            <xs:choice>
                <xs:sequence>
                    <xs:element name="parameter" type="parameter"/>
                </xs:sequence>
                <xs:sequence>
                    <xs:element name="parameter" type="xs:string"/>
                </xs:sequence>
            </xs:choice>
            <xs:element name="location" type="location"/>
            <xs:element name="observed" type="textbool"/>
            <!-- xs:element name="parameter" type="parameter"/ -->
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="layout">
        <xs:sequence>
            <xs:element name="title" type="xs:string"/>
            <xs:element name="x_axis" type="axis"/>
            <xs:element name="y_axis" type="axis"/>
            <xs:element name="z_axis" type="axis"/>
            <xs:element name="colormap" type="xs:string"/>
            <xs:element name="colorbar" type="textbool"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="parameter">
        <xs:sequence>
```

```
            <xs:element name="item" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="location">
        <xs:sequence>
            <xs:element name="item" type="location-item" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="axis">
        <xs:sequence>
            <xs:element name="selected" type="textbool"/>
            <xs:element name="label"/>
            <xs:element name="limits"/>
            <xs:element name="scale"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="location-item">
        <xs:sequence>
            <xs:element name="id" type="xs:integer"/>
            <xs:element name="name" type="xs:string"/>
            <xs:element name="selected" type="textbool"/>
            <xs:element name="weight" type="xs:integer"/>
            <xs:element name="style" type="style"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="style">
        <xs:sequence>
            <xs:element name="linestyle" type="xs:string"/>
            <xs:element name="linewidth" type="xs:integer"/>
            <xs:element name="color" type="xs:string"/>
            <xs:element name="marker" type="xs:string"/>
            <xs:element name="markersize" type="xs:integer"/>
            <xs:element name="markeredgecolor" type="xs:string"/>
            <xs:element name="markerfacecolor" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
    <xs:simpleType name="textbool">
        <xs:restriction base="xs:string">
            <xs:enumeration value="off"/>
            <xs:enumeration value="on"/>
        </xs:restriction>
    </xs:simpleType>
</xs:schema>
```

Please note that as a result of the addition of editable numerical parameters, version 1.0 and version 1.1 (and so on) of the xsd schema are not interchangeable.
Furthermore:

- In version 1.2 the observed tag was added.
- In version 1.3 the extra keywords for SWAN4072ABCDE were added.
- In version 1.4 SWAN4081 was added; but with the same keywords as SWAN4072ABCDE.
- In version 1.5 SWAN4091 was added; keywords WCAP1 and BREA1 were removed compared to the settings of the previous one.
- In version 1.6 SWAN4101 was added; but with the same keywords as SWAN4091.
- In version 1.7 SWAN4110 was added; but with the same keywords as SWAN4101.
- In version 1.8 ONR, HR2006, SWAN4041A, SWAN4051A , SWAN4072A and SWAN4081 were removed. SWAN 4120A was added; but with the same keywords as SWAN4110.
- In version 2.0 SWAN4131AB was added, with the same keywords as SWAN4101, and the new keywords mxitns, deltc, deltblk, delttbl, and deltspc.