Naval Research Laboratory
Stennis Space Center, MS 39529-5004

NRL/MR/7320--02-8286

User's Manual for the Simulating WAves Nearshore Model (SWAN)

RICHARD ALLARD
ERICK ROGERS

Ocean Dynamics and Prediction Branch
Oceanography Division

SUZANNE N. CARROLL

Planning Systems Incorporated
Stennis Space Center, MS

November 15, 2002

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
November 15, 2002

2. REPORT TYPE
Memorandum

3. DATES COVERED *(From - To)*

4. TITLE AND SUBTITLE

User's Manual for the Simulating WAves Nearshore Model (SWAN)

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
0602435N

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

6. AUTHOR(S)

Richard Allard, Erick Rogers, and Suzanne N. Carroll*

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
Oceanography Division
Stennis Space Center, MS 39529-5004

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/MR/7320--02-8286

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

10. SPONSOR / MONITOR'S ACRONYM(S)

11. SPONSOR / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

*Planning Systems Incorporated, Stennis Space Center, MS 39529-5004

14. ABSTRACT

The Simulating WAves Nearshore (SWAN) model is a numerical wave model used to obtain realistic estimates of wave parameters in coastal areas, lakes, and estuaries from given wind, bottom, and current conditions. The model is based on the wave action balance equation (or energy balance in the absence of currents) with sources and sinks. SWAN is a third-generation wave model with first-, second-, and third-generation options.

15. SUBJECT TERMS

Input, Output, Boundary conditions, Wind, Waves, Bottom friction, Grid

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Richard Allard
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	UL	171	**19b. TELEPHONE NUMBER** *(include area code)* (228) 688-4894

USER'S MANUAL

FOR THE

SIMULATING WAVES NEARSHORE MODEL (SWAN)

Cycle III Version 40.11

March 2002

Richard Allard
Erick Rogers

Ocean Dynamics and Prediction Branch
Oceanography Division
Naval Research Laboratory

Suzanne Carroll
Planning Systems Incorporated

Contract No. N00014-96-D-6031
PSI Technical Report SSC-001-02

Contract No. N00014-96-D-6031
PSI Technical Report SSC-001-02

Adapted From: SWAN Cycle III Version 40.11 USER MANUAL

Last Revision: October 19, 2000
Date of Printing: October 19, 2000

By: L. H. Holthuijsen
N. Booij
R.C. Ris
IJ. G. Haagsma
A. T. M. M. Kieftenburg
E. E. Kriezi
Contributors: R. Padilla-Hernandez
E.Rogers
J. Kaihatu
H. Petit

Delft University of Technology
Department of Civil Engineering
P.O. Box 5048
2600 GA Delft, the Netherlands
E-mail: swan.info@ct.tudelft.nl
Web site/home page: http://swan.ct.tudelft.nl

TABLE OF CONTENTS

List of Figures

List of Tables

1.0 Introduction

The Simulating WAves Nearshore (SWAN) model is a numerical wave model used to obtain realistic estimates of wave parameters in coastal areas, lakes and estuaries from given wind, bottom, and current conditions. The model is based on the wave action balance equation (or energy balance in the absence of currents) with sources and sinks. SWAN is a third-generation wave model with first-, second- and third-generation options.

2.0 Application

2.1 Description of SWAN Functionality

The following wave **propagation** processes are represented in SWAN:

- Rectilinear propagation through geographic space;
- Refraction due to spatial variations in bottom and current;
- Shoaling due to spatial variations in bottom and current;
- Blocking and reflections by opposing currents;
- Transmission through, blockage by, or reflection against sub-grid obstacles.

The effects of currents and sub-grid obstacles are not treated in this manual but are available in SWAN.

The following **wave generation** and **dissipation** processes are represented in SWAN:

- Generation by wind;
- Dissipation by whitecapping;
- Dissipation by depth-induced wave breaking;
- Dissipation by bottom friction;
- Wave-wave interactions (quadruplets and triads);
- Obstacles.

In addition, the wave-induced set-up of the mean sea surface can be computed in SWAN. Wave induced set-up and transmission/blockage/reflection are not treated in this manual but are available in SWAN.

Cycle III of SWAN is stationary, optionally non-stationary, and formulated in Cartesian (small scale) or spherical (small and large scale) coordinates. The stationary mode should be used only for waves with a relatively short residence time in the computational area under consideration. In other words, the travel time of the waves through the region should be small compared to the time scale of the geophysical conditions such as wave boundary conditions, wind, tides and storm surge. A quasi-stationary approach may be taken with stationary SWAN computations in a time-varying

Manuscript approved August 29, 2002.

sequence of stationary conditions. For one-dimensional (geographical) situations SWAN may be run in 1-D mode.

2.2 Changes and Compatibility Relative to Previous Version

The previous version of SWAN is 40.01. The present version of SWAN is 40.11.

2.2.1 Scientific Changes

SWAN can now compute waves on scales much larger than coastal scales. To accomplish this, SWAN uses more accurate numerical propagation schemes (see below), and it can compute on spherical coordinates (longitude, latitude), allowing calculations in laboratory situations, coastal regions, shelf seas and oceans, but not harbors. It must be emphasized that on oceanic scales SWAN is certainly less efficient than WAVEWATCH III and probably also less efficient than WAM (see **Section 3.0**).

In addition to transmitting waves through obstacles, SWAN can now reflect waves against obstacles such as coastlines or breakwaters.

In the formulation of depth-induced breaking, the option of using a variable wave height-to-depth ratio has been removed, as it did not improve the SWAN results (input of previous SWAN versions is still accepted by SWAN 40.11 but not supported).

2.2.2 Numerical Changes

More accurate propagation schemes (less diffusive) have replaced the Backward Space, Backward Time (BSBT) propagation scheme:

- The 2nd order upwind S&L scheme (for non-stationary computations with 3rd order diffusion);
- The Second ORder UPwind (SORDUP) scheme (for stationary computations with 3rd order diffusion).

These new schemes are default, but the old first order upwind BSBT scheme (for stationary and non-stationary computations with first order diffusion) is still available.

The approximation of the bathymetry in the refraction computations has improved. To give robust (but not necessarily accurate) results in case of poor resolution in bathymetry, currents or the wave field itself, the user can activate a limiter to avoid waves turning over more than 90° in one spatial grid step.

2.2.3 Configuration Changes

In addition to the existing option of nesting in WAM, SWAN can now also be nested in WAVEWATCH III (not fully tested yet).

The SWAN code is now written completely in Fortran90, except that occasionally the number of continuation lines exceeds 19. Therefore, SWAN can now operate on all platforms that support FORTRAN90 (including Linux). In addition, FORTRAN77 compilers cannot be used to install SWAN III. There are no other major configuration changes.

2.2.4 Compatibility

SWAN version 40.11 is fully downward compatible. The results may vary from results with previous versions because of the new propagation schemes, improved bathymetry approximations for refraction and bug fixes.

2.3 Relation to WAM, WAVEWATCH III and Others

The basic scientific philosophy of SWAN is identical to that of WAM. SWAN is a third-generation wave model, just as WAM (Cycle 3 and 4) is, and it uses the same formulations for the source terms (just like TOMAWAC) and some additional formulations, primarily for shallow water. However, the numerical techniques are very different, although SWAN shares the code for the quadruplet interactions with WAM that are originally from Hasselmann et al. (1985) and somewhat adapted by Tolman (personal communication, 1993). WAVEWATCH III also uses different numerical techniques and different formulations for the wind input and whitecapping.

This close similarity can be exploited in the sense that scientific findings with one model can be shared with the others. SWAN can be readily nested in WAM and WAVEWATCH III (the formulations of WAVEWATCH III have not (yet) been implemented in SWAN).

If SWAN is nested in WAM or WAVEWATCH III, then the boundary conditions for SWAN provided by WAM or WAVEWATCH III may not be model consistent even if the same physics are used. This may be due to numerical techniques or implementation for the geographic area (spatial and spectral resolutions, coefficients etc.). Generally the deep-water boundary of the SWAN nest must be located in WAM or WAVEWATCH III where shallow water effects do not dominate (to avoid too large discontinuities between the two models). Also, the spatial and spectral resolutions should not differ by more than a factor of two or three. If a finer resolution is required a second or third nesting may be needed.

2.4 Directory Structure

2.4.1 Model Run Files and Routines

The user has access to the following files through the SWAN web site (*http://swan.ct.tudelft.nl/home.htm*):

<u>In subdirectory **programs**:</u>

SWAN modules containing global variables:	swmod1.for
SWAN main program:	swanmain.for
SWAN preparation routines:	swanpre1.for swanpre2.for
SWAN computation routines:	swancom1.for swancom2.for swancom3.for swancom4.for swancom5.for swancomi.for
SWAN output routines:	swanout1.for swanout2.for swanout3.for
SWAN service routines:	swanser.for
Installation dependent subroutines:	ocpids.for
Command reading routines:	ocpcre.for
Dynamic pool routines:	ocpdpn.for
Plot routines:	ocplot.for
Miscellaneous routines:	ocpmix.for
Include files containing common block variables:	ocpcomm1.inc ocpcomm2.inc ocpcomm3.inc ocpcomm4.inc swcomm1.inc

	swcomm2.inc swcomm3.inc swcomm4.inc poolcomm.inc timecomm.inc
Run procedure on DOS system:	swan.bat
Run procedure on Unix system:	swan.unx
Command file editing support:	swan.edt
In subdirectory **test_cases**:	
For tests:	files with an extension .swn files with an extension .bot
In subdirectory **util**:	
For conversion of spectra:	convert1d.for cvspec1d.for cvspec2d.for
For conversion from DOS to Unix:	dos2unix dos2unix.readme

3.0 Limitations and Assumptions

3.1 Computer Limitations and Requirements

Most of the numerical data used by SWAN resides in the dynamic data pool. Its size is set at the time of SWAN installation. Sometimes this pool is too small. If so, the following message appears in the print file:

> "Terminating error: array is too small"

If this error message appears, the user must reduce the number of computational nodes or increase MXPOOL. In such cases, the command POOL is sometimes useful. This command is a diagnostic tool that gives the size of the data pool in the user's implementation of SWAN (diagnostic information is given in the print file).

To run the SWAN program for the test cases, 55 Mb of free internal memory is recommended. For more realistic cases 100 to 500 Mb may be needed, whereas for simpler stationary or 1-D cases significantly less memory is needed (less than 5 Mb for 1-D cases). The number of files addressable by the DOS system should be at least twenty, therefore the command: FILES=20 (or some higher number, if necessary) should be included in the file configuration system of the DOS operating system.

3.2 Model Limitations

Diffraction is not modeled in SWAN, so SWAN should not be used in areas where variations in wave height are large within a horizontal scale of a few wavelengths. Because of this, the wave field computed by SWAN will generally not be accurate in the immediate vicinity of obstacles and certainly not in harbors.

SWAN does not calculate wave-induced currents. If relevant, such currents should be provided as input to SWAN (e.g. from a hydrodynamic model, which can be driven by waves from SWAN in an iteration procedure). As an option SWAN computes wave-induced set-up. In a (geographic) 1-D case, the computations are based on exact equations. In 2-D cases, the computations are based on approximate equations. The effects of wave-induced currents are ignored and in 1-D cases they do not exist.

The Lumped Triad Approximation (LTA) for the triad wave-wave interactions seems to depend on the width of the directional distribution of the wave spectrum. The present tuning in SWAN (for default settings see command TRIAD, **Section 4.12.9.6**) seems to work reasonably well in many cases, but it has been obtained from observations in a narrow wave flume (long-crested waves).

The Discrete Interaction Approximation (DIA) for the quadruplet wave-wave interactions also seems to depend on the width of the directional distribution of the wave spectrum as well as the frequency resolutions. It works reasonably well in many cases, but is a poor approximation for long-crested waves (narrow directional distribution) and frequency resolutions different from 10% (see command CGRID, **Section 4.12.6.1**). WAM and WAVEWATCH III share this problem.

SWAN Version 40.11 can be used on any scale relevant for wind generated surface gravity waves (high quality, third order diffusion propagation and Cartesian or spherical coordinates). However, SWAN is specifically developed for coastal applications that would normally not require such flexibility in scale. The reasons for providing SWAN with such flexibility are:

- To allow SWAN to be used from laboratory conditions to shelf seas (but not harbors, see above);
- To nest SWAN in the WAM or WAVEWATCH III models, which are formulated in terms of spherical coordinates;
- These facilities are not meant to support the use of SWAN on oceanic scales, however, because the authors have no ambition to apply SWAN beyond coastal areas. SWAN has not been extensively tested on oceanic scales and is certainly less efficient than WAVEWATCH III or WAM on oceanic scales (SWAN does not parallelize or vectorize well).

3.3 Internal Scenarios, Limiters, Shortcomings and Coding Bugs

User inputs may cause SWAN to produce unreliable or unrealistic results, especially in cases where the bathymetry or the wave field is not well resolved. Be aware that the grid on which the computations are performed interpolates from the grids on which the input is provided; different resolutions for these grids (which are allowed) can therefore create unexpected interpolation patterns on the computational grid. In such cases SWAN, instead of terminating the computations, may invoke the following internal scenarios or limiters (see examples below). The reasons for this model policy are that (a) SWAN needs to be robust or (b) the problem may be only very local or (c) the problem needs to be computed through to be diagnosed.

The SWAN user may encounter other fundamental shortcomings and unintentional coding bugs of SWAN, which may or may not be typical for third-generation wave models.

Because of these shortcomings, the results may look realistic but they may (locally) not be accurate.

Coding bugs and limitations:

- The user can request that refraction over one spatial grid step is limited to about 90° (see command NUMERIC, **Section 4.12.10.2**). This may be relevant when the depth varies considerably over one spatial grid step (e.g. at the edge of oceans or near oceanic islands with only one or two grid steps to go from oceanic depths to a shallow coast). This implies inaccurate refraction computations in such grid steps. This may be acceptable when refraction has only local effects that can be ignored but, depending on the topography, the (inaccurately computed) effects may radiate far into the computational area.
- SWAN cannot handle wave propagation on super-critical current flow. If such flow is encountered during SWAN computations, the current is locally reduced to sub-critical flow.
- If the water depth is less than some user-provided limit, the depth is set at that limit (default is 0.05 m, see command SET, **Section 4.12.4.2**).
- SWAN may not reproduce the user-imposed wave boundary conditions as SWAN replaces the *imposed* waves that move out of the computational area at the boundaries with the *computed* waves that move out of the computational area at the boundaries.
- SWAN may have convergence problems. There are three iteration processes in SWAN:
  a) An iteration process for the spatial propagation of the waves.
     For spatial propagation the change of the wave field over one iteration is limited to some realistic value (usually several iterations for stationary conditions or one iteration or upgrade per time step for non-stationary conditions (see command NUMERIC, **Section 4.12.10.2**)). This is a common problem for all third-generation wave models that use the DIA of the quadruplet interactions. It does not seem to affect the result seriously in many cases but sometimes SWAN fails to converge properly due to an unresolved bug.
  b) An iteration process for spectral propagation (current-induced refraction and frequency shift) if ambient currents are present.

For spectral propagation (but only current-induced refraction and frequency shift), SWAN may not converge some unresolved bug in the numerical technique. (See the SWAN web site).

c) An iteration process for solving the set-up equation if the user requests wave-induced set-up.

   For the wave-induced set-up, SWAN may not converge some unresolved bug in the numerical technique. (See the SWAN web site).

Information on the actual convergence of a particular SWAN run is provided in the PRINT file (available on the SWAN web site).

Any change in these scenarios, limiters or shortcomings (in particular newly discovered coding bugs and their fixes) are published on the SWAN web site and implemented in new releases of SWAN (http://swan.ct.tudelft.nl).

4.0 Operating Guidelines

The purpose of this section is to give some general advice in choosing the basic input for SWAN computations. Suggestions are given that should help the user to choose among the many options available in SWAN, such as what physical processes to include or exclude, boundary conditions, and in which mode or dimension to run SWAN (first-, second- or third-generation mode and 1-D or 2-D).

An important question addressed is how to choose various grids in SWAN (input, computation, output), (resolution, orientation etc.), or nesting options (only for uniform rectilinear grid). The idea of nesting is to first compute the waves on a coarse grid for a larger region and then on a finer grid for a smaller region. The computation on the fine grid uses boundary conditions that are generated by the computation on the coarse grid. Nesting can be repeated on ever decreasing scales. Use the same type of coordinates for coarse and nested computations (Cartesian or spherical). Curvilinear grids can be used for nested computations but the boundaries should always be rectangular.

It must be pointed out that the application of SWAN on ocean scales is not recommended from an efficiency point of view. The WAM and WAVEWATCH III models have been designed specifically for ocean applications and are probably one order of magnitude more efficient than SWAN (SWAN can be nested into these models). SWAN can be run on large scales (much larger than coastal scales) but this option is mainly intended for the transition from ocean scales to coastal scales (transitions where non-stationary is an issue and spherical coordinates are convenient for nesting).

A general suggestion is to start simple. SWAN helps in this with default options.

4.1 Customizing SWAN

Although SWAN runs unaltered on almost any type of computer, there may be reasons to make changes in the source code before compilation:

(A) Depending on the complexity of the cases for which SWAN is used and the computer memory available, the size of the dynamic data pool can be set to another value. The procedure for this is explained in **Section 4.1.1**.

(B) Some computer or compiler-dependent Fortran statements can or need to be activated. **Section 4.1.2** deals with compiler dependent statements. The compiler dependent statements are provided for Fortran90 compilers that have been used at Delft University of Technology (Lahey for Windows and Linux, Absoft for MacOS, MIPSpro for SGI-Irix and CF90 for Cray-Unicos).

(C) SWAN also allows for customizing the input and the output to the user's wishes. This either can be done permanently by changing several statements in ocpids.for, as explained in **Section 4.1.3** or by changing the settings in the file swaninit, which is created the first time SWAN is executed on a computer system. The changes in swaninit only affect the runs executed in the directory that contains the modified swaninit file. A detailed description of the swaninit file is given in **Section 4.2**.

4.1.1 Modifications in Pool Size (poolcomm.inc)

The dynamic data pool stores all data used in the SWAN calculations that are dependent on the size of the case for which SWAN is used. The size of the dynamic data pool is set in one of the include files. In order to enlarge or decrease the size of the pool, change the number in the statement:

PARAMETER (MXPOOL=....).

This statement is located in the file poolcomm.inc at point 5 of the header (parameter variables). The size of the pool is approximately the product of the maximum number of grid points that can be used in the calculations. Therefore:

- In the case of stationary calculations:	mxc*myc*msc*mdc.
- In the case of non-stationary calculations:	2*mxc*myc*msc*mdc.
- In the case of [iquad] = 3, add:	mxc*myc*msc*mdc.

More detailed information regarding memory requirements is found in **Section 4.11.3**. **Note:** These numbers cost MXPOOL*4 bytes. The default setting for MXPOOL is 12,500,000, which is the recommended amount to run the cases in the benchmark test for SWAN. The setting for MXPOOL can be changed according to individual needs using the above estimates for the required memory. For field cases the pool-size is normally set between 20,000,000 and 100,000,000. SWAN has been used with pool-sizes up to 500,000,000 (2 GB of memory). If you make the pool-size too large (i.e.

you have limited memory in your computer system), errors can result during the compilation phase. If the pool is too small, the program will give an indication of how large it should be. One can also use the POOL command in SWAN to get information on the size of the pool during run time.

4.1.2 Compiler Dependent Statements (ocpids.for and ocpmix.for)

The subroutine OCPINI (ocpids.for) and FOR (ocpmix.for) should be adapted when using a Cray or SGI f90 compiler. These compilers cannot read/write lines longer than 256 characters by default. The option RECL = 1000 in the OPEN statement will also allow these compilers to read/write sufficiently long lines. The appropriate option is indicated as follows:

C/Cray:	Cray f90 compiler
C/SGI:	SGI f90 compiler

You should remove the C/* code to activate the commands for the specific compiler.

4.1.3 User Dependent Modifications (ocpids.for)

Subroutine OCPINI:

a. The integers INPUTF and PRINTF are the unit reference numbers for the input and print files. On most mainframe computers they are 5 and 6, respectively. If necessary, you can change these to the standard input and output unit numbers for your installation. Another unit reference number, SCREEN, is for output to screen. When using a batch-system, SCREEN should be set equal to PRINTF, as there is no separate output to screen. PRTEST is a unit number used for a separate test print file. In the version that you downloaded from SWAN's ftp site, PRTEST is equal to PRINTF so that the test output will appear on the same file as the standard print output.

b. The standard input file and standard print file are usually named 'INPUT' and 'PRINT', respectively (see Subroutine OCPINI). If the users system has problems with these explicit OPEN statements, please inform Delft University or NRL.

   NB Other OPEN statements appearing in SWAN usually need not be modified since the filenames are variable.

c. TABC is the ASCII value for the TAB character. Change the number 9 in the assignment statement TABC = CHAR(9) to the value in the user's ASCII table (which may be different from 9).

d. COMID is the comment identifier. It is usually '$', but on some installations this may be inappropriate because a line beginning with $ is interpreted as a command for the operating system (e.g. VAX systems). If necessary change to '!'.

e. The directory separation character varies depending on the platform that is used (CHAR(92) (='\') on a PC; / on a Unix system and : on a Macintosh). DIRCH1 indicates the separation character that is expected in the input file, and DIRCH2 indicates the character that is used on the platform that SWAN is run on. Proper use of this option allows the sharing of input files across platforms.

f. The character string, INST, contains the name of the institute that carries out the computations. You can assign to it the name of your institute instead of 'DELFT UNIVERSITY OF TECHNOLOGY', which is its present value.

g. The type of plot code is chosen by subroutine OCPINI as well as a number of other parameters. These are the same values that will appear in the file swaninit (see **Section 4.2**). The standard plot code option is HPGL. There are six significant plotting parameters that have the following meaning for the HPGL option:

1) Maximum size of the figure frame in horizontal direction
2) Maximum size of the figure frame in vertical direction
3) Conversion factor from cm to plotting units (plotting units per centimeter)
4) Horizontal margins in centimeters
5) Vertical margins in centimeters
6) Orientation of the paper (0 = landscape; 1 = portrait).

The other possibility is called OPPL (Ocean Pack Plot code), which is the same code as used in HISWA. HISWA users who prefer to keep the same type of plot code should change the word HPGL to OPPL in the setup file or in the source code.

NB The standard (default) paper format is A4. For other (standard) paper formats the first two parameters must be adjusted.

h. There are various options for the shape of the frame used in graphical output. If necessary, new options can be added in the subroutine OPFRAM, which is also in the file ocpids.for. The options include:

= 2	Full frame appears with caption (default),
= 1	Only the frame appears,
= 0	No frame is plotted.

The file ocpids.for contains a set of basic plotting subroutines with names starting with OP which can generate two types of plot code: HPGL (HP graphic language) or OPPL (Ocean Pack Plot code, i.e., the same code as produced by HISWA). If a user so desires he can add options in each subroutine to generate another plot code. The basic plotting routines do not send plot instructions directly to the plotter but generate ASCII data in a file. Depending on the plot code used, a separate program is needed to translate these data into actual plot instructions. Such a program is installation dependent and is therefore not commonly available with the SWAN package.

i. Date and time can be read and written according to the various options discussed below. Subroutines DTSTTI and DTTIST are used to convert a time string to and from numerical time indication. The following options are available:

= 1	ISO notation:	19870530.153000
= 2		30-May-87 15:30:00
= 3		05/30/87 15:30:00
= 4		15:30:00
= 5		87/05/30 15:30:00
= 6	(as in WAM):	8705301530

Suggestions for new options can be made to the SWAN authors.

Note: The ISO notation is recommended, as it has no millennium problem. In all other cases the range of valid dates is chosen to be between 1 January 1911 and 31 December 2010 (both inclusive).

4.2 Setup File *swaninit*

The standard version of the setup file swaninit (which is generated the first time that SWAN is run after implementation) reads:

```
      2                                          Version of initialization file.
   Delft University of Technology                Name of institute.
      3                                          Command file reference number.
 INPUT                                           Command file name.
      4                                          Print file reference number.
 PRINT                                           Print file name.
      4                                          Test file reference number.
                                                 Test file name.
      6                                          Screen reference number.
    9 9                                          Highest file reference number.
      $                                          Comment identifier.
 [TAB]                                           TAB character.
 \                                               Dir sep character in input file.
 \                                               Dir sep character replacing previous one.
  HPGL                                           Plotting option (OPPL, HPGL, etc).
      6                                          Number of plotting parameters (Default values shown.
                                                 Definition of parameters on line 1104 of ocpids.for).
     18. 0000   26. 0000   402. 0000   0. 0000   0. 0000   0. 0000
      2                                          Default plotting frame option.
      1                                          Default time coding option.
```

Use the TAB key on your keyboard to insert [TAB] in this file.

The version number of the initialization file is included in the file so that SWAN can verify whether the file it reads is a valid setup file. The references to the previous section (**4.1.3**), under OCPINI a to g are:

Name of institute:	see e.
The data for command file, print file, test file and screen:	see a and b.
Comment identifier:	see d.
Tab character:	see c.
Directory separation character:	see e.
Plotting option and plot parameters:	see g.
Plot frame:	see h.
Date/time option:	see i.

4.3 Compiling SWAN

After the necessary modifications are made as described in **Section 4.1**, the source code is ready for compilation. All source code except the input for test problems is written in fixed form Fortran90 so a Fortran90 compiler must be available in order to compile SWAN. SWAN has been developed with the Lahey F95 compiler version 5.00 and for SGI workstations, with the f90 compiler version 7.30. On Unix (Irix, Unicos, Linux, etc.) workstations it may be necessary to rename all *.for files to *.f files. The source code cannot be compiled with a Fortran77 compiler since it contains several Fortran90 features. The SWAN source code complies with the ANSI Fortran90 standard, except for a few cases where the limit of 19 continuation lines is violated. The SWAN developers are currently not aware of a compiler that cannot deal with this violation of the ANSI standard. Furthermore, the source code contains one remaining computed *goto*, which is an obsolete feature in Fortran90. This should not affect the compilation or the execution of SWAN. In the future, SWAN will be rewritten in free form Fortran90 and all CHARACTER*n statements will be replaced by a CHARACTER(LEN=n) statement to prepare for Fortran95.

When compiling SWAN, check that the compiler allocates the same amount of memory for all REALS, INTEGERS and LOGICALS. Normally four bytes are allocated for these variables. Sometimes on supercomputers (vector or parallel) the memory is eight bytes. When a compiler allocates eight bytes for a REAL and four bytes for an INTEGER, for example, SWAN will not run correctly. Most compilers have an option that initializes all variables automatically to zero. It is strongly recommended to use that option. There are cases that cannot run correctly if this compiler-option is not used. The module swmod1.for should be compiled first. Several subroutines use the module contained in this file and they need a compiled version of swmod1.for before they can be compiled.

4.4 Linking SWAN

The following files should be linked, in order to get a SWAN executable:

Object files of the source files:	swmod1.for swanmain.for, swanpre1.for, swanpre2.for, swancom1.for, swancom2.for, swancom3.for, swancom4.for, swancom5.for,

swancomi.for, swanout1.for, swanout2.for, swanout3.for, swanser.for, ocpids.for, ocpcre.for, ocpdpn.for, ocplot.for, ocpmix.for

The name of the executable file is

"a.out"	for the versions running under Unix.
"swanmain.exe"	for the PC version generated with the Lahey Fortran90 compiler (in case swanmain.for is first on the list). It is suggested that this file be renamed swan.exe.

4.5 SWAN Command File

The name of the command file should have the extension .swn.

- If the name is test01.swn, the DOS command to run SWAN with this file is swan.bat test01 runs in a DOS box under the WINDOWS operating system.

- In this command file, the user should give the file names. In the MS-DOS operating system these file names consist of two parts, the first of which is at most eight characters long, and the second of which is at most three characters long. The two parts are separated by a dot. The MS-DOS system does not distinguish between lowercase and uppercase characters in filenames.

- The file swan.bat should be generated at the time of installation of SWAN (see **Section 4.5.1**).

- The same procedure is followed on other computers with different operating systems. When running under the Unix operating system remember that filenames are case-sensitive unlike those on MS-DOS.

- The file swan.edt is available in Appendix C and from the internet-site to assist in editing an input file for SWAN.

4.5.1 Batch File and Unix Script

The run procedure for SWAN is dependent on the operating system. The procedure should do the following:

1. Copy the command file to INPUT (assuming INPUT is the standard file name for command input, see **Section 4.1.3b**).
2. Run SWAN.
3. Copy the file PRINT, assuming PRINT is the standard file name for output. See **Section 4.1.3b**).

Examples of such a procedure for DOS (file swan.bat) or for Unix systems (file swan.unx) are given below.

Suggested file swan.bat for DOS:

```
echo off
if not exist %1.swn goto error
copy %1.swn INPUT
if exist Errfile del Errfile
c:..\swan.exe
if exist %1.prt del %1.prt
ren PRINT %1.prt
if not exist Errfile goto finish
echo Errors have occurred
type Errfile
pause
goto finish
:error
echo.    *******************************************************
echo.
echo.              input file %1.swn does not exist
*
echo.    *******************************************************
echo.
*
echo
pause
:finish
echo on
exit
```

The path of the swan.exe file in this version of the batch file might have to be adjusted.

Suggested script file for Unix (swan.unx):

```
#!/bin/csh
set SWAN=/prog
set INP=/test
cp $INP/$1.swn INPUT >nul
$SWAN/swan
cp PRINT $INP/$1.prt >nul
cp source $INP/$1.src >nul
```

Some adjustments may be necessary. A similar procedure may be followed on other operating systems.

4.6 Units and Coordinate Systems

SWAN expects all quantities input by the user to be expressed in S.I. units: m, kg, s and composites of these with accepted compounds (such as Newton (N) and Watt (W)). Consequently the wave height and water depth are in m, wave period in s, etc. For wind and wave direction both the

Cartesian and a Nautical convention may be used (see below). Directions and spherical coordinates are in degrees (°) and not in radians.

For the output of wave energy the user can choose between variance (m^2) or energy (spatial density in Joule/m^2 i.e., energy per unit sea surface) and the equivalents in case of energy transport (m^3/s or Watt/m i.e., energy transport per unit length) and spectral energy density (m^2s/rad or $Js/m^2/rad$ i.e., energy per unit frequency and direction per unit sea surface area). The wave-induced stress components obtained as spatial derivatives of wave-induced radiation stress are always expressed in N/m^2 even if the wave energy is in terms of variance. Note that the energy density is also in Joule/m^2 in the case of spherical coordinates.

SWAN operates either in a Cartesian coordinate system or in a spherical coordinate system, i.e., in a flat plane or on a spherical earth. In the Cartesian system all geographic locations and orientations in SWAN, such as for the bottom grid or for output points, are defined in one common Cartesian coordinate system with origin (0, 0) by definition. The user may choose this geographic origin arbitrarily. This system is designated in this manual as the problem coordinate system. With a spherical coordinate system all geographic locations and orientations in SWAN are defined in geographic longitude and latitude. Both coordinate systems are designated in this manual as the problem coordinate system.

In the input and output of SWAN the direction of wind and waves are defined according to either (a) the Cartesian convention, which is the direction of the vector points, measured counterclockwise from the positive x-axis of this system (in degrees) or (b) a Nautical convention which is the direction the wind or waves come from, measured clockwise from geographic North. All other directions (such as orientation of grids) follow the Cartesian convention.

For regular grids (i.e. uniform and rectangular), **Figure 4.12-1** in **Section 4.12.6.1** shows how the locations of the various grids are determined with respect to the problem coordinates. All grid points of a curvilinear grid are relative to the problem coordinate system.

4.7 Choice of Grids, Time Windows and Boundary / Initial / First Guess Conditions

Several types of grids and time window(s) need to be defined: (a) spectral grid, (b) spatial (geographic) grids and time window(s) (for nonstationary computations).

The **spectral** grid that needs to be defined by the user is:

- A computational spectral grid on which SWAN performs the computations. SWAN has the option to make computations that are nested in SWAN, WAM, or WAVEWATCH III). In such cases, the nested spectral grid need not be equal to the spectral grid in the coarse model run.

The **spatial** grids that need to be defined by the user are (if required):

- One (or more) spatial input grid(s) for the bottom, current field, water level, friction coefficient and wind (each input grid may differ from the others),
- A computational spatial grid on which SWAN performs the computations and,
- One (or more) spatial output grid(s) on which the user requires output of SWAN.
- The winds and bottom friction do not require a grid if they are uniform over the area of interest.
- If a uniform and rectangular computational spatial grid is used in SWAN (to be decided by the user), then all input and output grids should be uniform and rectangular, but they may all be different from each other.
- If a curvilinear computational spatial grid is used in SWAN (to be decided by the user), then each input grid should be (a) uniform and rectangular (but not necessarily identical to the other grids) or (b) identical to the curvilinear computational grid or staggered with respect to the curvilinear computational grid.
- Output grids used for plotting of isolines and vector fields always need to be uniform and rectangular.
- SWAN has the option to make computations that are nested in SWAN, WAM or WAVEWATCH III. In such runs, SWAN interpolates the spatial boundary of the SWAN, WAM or WAVEWATCH III grid to the user provided grid of SWAN (that needs to nearly coincide along the grid lines of WAM, WAVEWATCH III, or the output nest grid boundaries of SWAN). Since grid lines in WAM and WAVEWATCH III are in spherical coordinates, it is recommended to use spherical coordinates in a nested SWAN when nesting in WAM or WAVEWATCH III.

Similarly, the wind fields may be available in different time windows than the current and water level fields and the computations may need to be carried out at other times than these input fields. For these reasons, SWAN operates with different time windows and different time steps each having a different start and end time and time step:

- The user gives one (or more) input time window(s) in which the bottom, current field, water level, friction coefficient and wind field (if present). Each input window may differ from the others,
- One computational time window in which SWAN performs the computations, and
- One (or more) output time window(s) in which the user requires output of SWAN.

SWAN has the option to make computations that are nested in SWAN, WAM or WAVEWATCH III. SWAN searches the boundary conditions in the relevant output file of the previous SWAN, WAM or WAVEWATCH III runs to take the boundary conditions at the start time of the nested run. It will not take the initial condition (i.e. over the entire computational grid) for the nested run from the previous SWAN, WAM or WAVEWATCH III run.

During the computations (on the computational grid and window) SWAN obtains bottom, current, water level, wind and bottom friction information by tri-linear interpolation from the input grid(s) and window(s). The output is in turn obtained in SWAN by bi-linear interpolation in space from the computational grid. There is no interpolation in time for output (this does not apply to input). The output time is shifted to the nearest computational time level. The user can reduce interpolation errors by taking the grids and windows as equal to one another as possible (preferably identical). It is recommended to choose output times such that they coincide with computational time levels.

4.7.1 Input Grid(s) and Time Window(s)

The bathymetry, current, water level, bottom friction and wind (if spatially variable) need to be provided to SWAN on so-called input grids. It is best to make an input grid so large that it completely covers the computational grid.

In the region outside the input grid, SWAN assumes that the bottom level, water level, and friction coefficient are identical to those at the nearest boundary of the input grid (lateral shift of that boundary). In the regions not covered by this lateral shift (i.e. in the outside quadrants of the corners of the input grid), a constant field equal to the value at the nearest corner point of the input grid is taken. For the current and wind velocity SWAN takes 0 m/s for points outside the input grid.

One should choose the spatial resolution for the input grids such that relevant spatial details in the bathymetry, currents, bottom friction and wind are properly resolved. Special care is required in cases with sharp and shallow ridges (sand bars, shoals) in the sea bottom and extremely steep bottom slopes. Inaccurate bathymetry can result in inaccurate refraction computations that can propagate into areas where refraction is not significant (the results may appear to be unstable). For instance, waves skirting an island that is poorly resolved may propagate beyond the island with entirely wrong directions. In such a case it may even be better to deactivate the refraction computations if refraction is irrelevant for the problem at hand. Such as if the refracted waves will run into the coast anyway. In such cases, the ridges are vitally important to obtaining good SWAN results (at sea the waves are "clipped" by depth induced breaking over the ridges and must be represented in SWAN computation). This requires not only that these ridges should be well represented on the input grid and after interpolation on the computational grid. This can be achieved by choosing the grid lines of the input grid along the ridges (even if this may require some slight "shifting" of the ridges) and by choosing the computational grid to be identical to the input grid. Otherwise, the ridge may be "lost" in the interpolation from the bottom-input grid to the computational grid.

In SWAN, the bathymetry, current, water level, wind and bottom friction may be time-varying, in which case there is a time window associated with the input (this need not be identical with the computational, output or other input windows). It is best to make input windows such that they entirely contain the computational window.

SWAN assumes zero values for input at times before the earliest begin time of the input parameters (e.g. wind speed). SWAN assumes constant values (the last values) at times after the end time of each input parameter.

For input one should use a time step that is small enough that time variations in the bathymetry, current, water level, wind and bottom friction are well resolved.

Since the locations of input and other grids are formulated in terms of numbers instead of maps, errors may easily occur. The PLOTGEO command has options to plot bathymetry together with locations of grids. This can be used to check the locations.

4.7.2 Computational Grids and Boundary / Initial / First Guess Conditions

The user must choose the computational spatial grid. The orientation (direction) can be chosen arbitrarily.

The boundaries of the computational spatial grid in SWAN are either land or water. With land there is no problem because the land does not generate waves and absorbs all incoming wave energy. Water boundaries pose a problem, because wave conditions are often unknown and SWAN assumes that no waves can enter and exit the area freely. These assumptions obviously contain errors that propagate into the model. Boundaries must therefore be chosen far enough away from the area where reliable computations are needed so as not to affect the computational results. This can be established by varying the location of the boundaries and inspecting the effect on the results. Sometimes the waves at these boundaries can be estimated with a certain degree of reliability, such as when the results of another model run (nested computations) or observations are available. If model results are available along the boundaries of the computational spatial grid, they are usually from a coarser resolution than the computational spatial grid under consideration. The resolution of the forcing will obviously have an impact on the simulation and must be chosen such that variability of forcing is well described.

Non-specification of boundary conditions is equivalent to specification of zero incident energy at that boundary. This can obviously lead to error. Coarse specifications of incident energy (e.g., by nesting) is preferable to uniform specification, which is in turn preferable to non-specification. If observations are available, they can be used as input at the boundaries, or some portion thereof.

A special case occurs near the coast, where it is often possible to identify an up-wave boundary (with proper wave information) and two lateral boundaries (with partial or no wave information). The affected areas with errors are typically regions with the apex at the corners of the water boundary with wave information, spreading towards shore at an angle of 30° to 45° for wind-sea conditions to either side of the imposed mean wave direction. The angle is less for swell conditions and is essentially the one-sided width of the directional distribution of wave energy. For propagation of short crested waves (wind-sea conditions) an example is given in **Fig. 4.7-1**. For this reason the lateral boundaries should be sufficiently far away from the area of interest to avoid the propagation of this error into this area. Such problems do not occur if the lateral boundaries contain proper wave information over their entire length, e.g., obtained from a previous SWAN computation or if the lateral boundaries are coast.

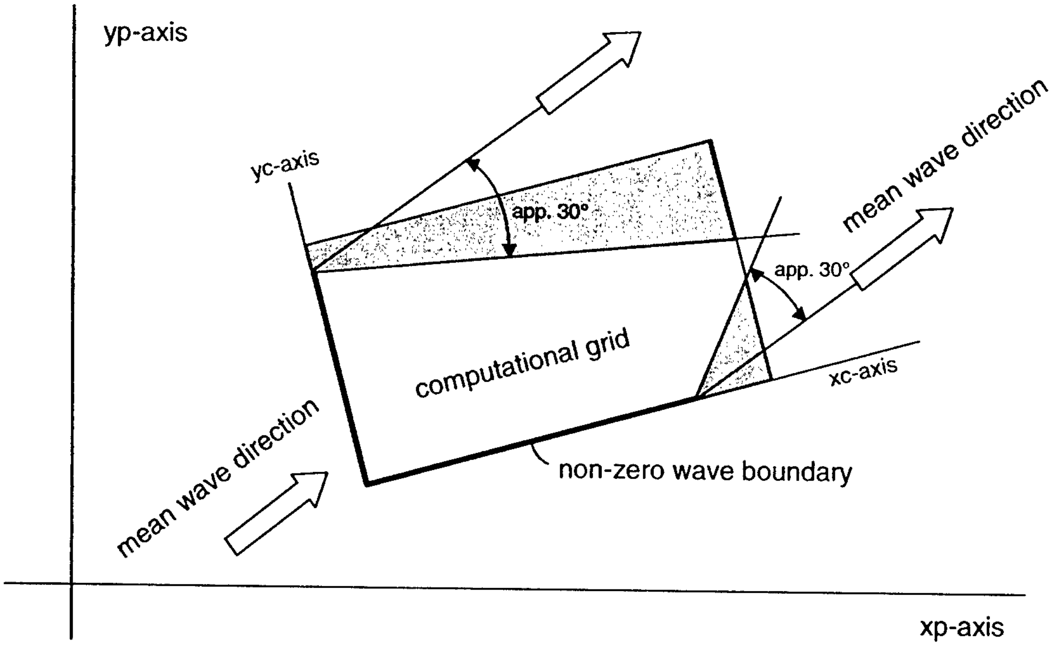


Fig. 4.7-1. Disturbed regions in the computational grid due to erroneous boundary conditions are indicated with shaded areas.

If the computational grid extends outside the input grid, the reader is referred to **Section 4.7.1** (input grid) to find the assumptions of SWAN on depth, current, water level, wind and bottom friction in the non-overlapping area.

The spatial resolution of the computational grid should be sufficient to resolve relevant details of the wave field. It is usually a good choice to take the resolution of the computational grid approximately equal to that of the input (bottom/current) grid.

SWAN may not use the entire user-provided computational grid if the user defines exception values on the depth grid (see command INPGRID, **Section 4.12.7.1**) or the computational grid (see command CGRID, **Section 4.12.6.1**). In such cases, a computational grid point is either:

- "wet" (i.e. the grid point is included in the computations because it represents water; this may vary with time-dependent or wave-induced water levels),
- "dry" (i.e. the grid point is excluded from the computations because it represents land which may vary with time-dependent or wave-induced water levels), or
- "exceptional" (i.e. the grid point is permanently excluded from the computations because it is so defined by the user).

If "exceptional" grid points occur in the computational grid, then SWAN filters the entire computational grid as follows:

- each grid line between two adjacent wet computational grid points (a wet link) without an adjacent, parallel wet link is removed,

- each wet computational grid point that is linked to only one other wet computational grid point is removed,
- SWAN deletes each wet computational grid point that has no wet links after the first filter (repeated until all such wet computational grid points are removed).

The effect of this filter is that if exception values are used for the depth grid or the computational grid, one-dimensional water features are ignored in the SWAN computations (results at these locations with a width of about one grid step would be unrealistic anyway). This is obvious in the output of SWAN, as there are no results on these features that are visible in relevant plots. It must be noted that the actual code in SWAN does not use the above procedure but another with the same effect. If no exception values are used, this filter is not applied because one-dimensional features may appear or disappear due to changing water levels (flooding may create them, or drying may reduce two-dimensional features to one-dimensional features).

When output is requested along a boundary of the computational grid it may differ from the boundary conditions that are imposed by the user. The reason is that SWAN accepts only the user-imposed incoming wave components and replaces the user-imposed outgoing components with computed outgoing components (propagating to the boundary from the interior region). Moreover, SWAN has an option to only compute within a pre-set directional sector (pre-set by the user). Wave components outside this sector are totally ignored by SWAN (with no replacements). The user is informed by means of a WARNING in the output print file. See **Section 4.9** when the computed significant wave height differs by more than 10% (the default) from the user-imposed significant wave height (see commands BOUND..., **Section 4.12.8**). The user can set the actual value of this difference (see command SET, **Section 4.12.4.2**).

The user must choose the **computational window** in time. If the initial condition is not well prescribed by the user (e.g., a user typically indicates a zero-state or parameterized initial condition), the computational window should start early enough so that the initial state of SWAN has propagated through the computational area. Note that a stationary computation can precede a nonstationary computation in a single SWAN simulation (i.e. a single command file with multiple "COMP" lines. See below). This is sometimes done in order to create a more realistic initial condition for the nonstationary computation.

The user should give the computational time step. Since SWAN is based on implicit numerical schemes, it is not limited by the Courant stability criterion (which couples time and space steps). In this sense, the time step in SWAN is unlimited. However, the accuracy of the results of SWAN is obviously affected by the time step. Generally the time step in SWAN should be small enough to resolve the time variations of the computed wave field itself. Usually, it is enough to consider the time variations of the driving fields (wind, currents, water depth, wave boundary conditions), but be careful. Relatively small time variations in depth (e.g. by tide) can result in relatively large variations in the wave field. At the same time, a large Courant number (>> 1) will lead to increased numerical error (especially true if using the S&L scheme).

The first guess conditions of a stationary run of SWAN are default determined with the 2nd generation mode of SWAN. The initial condition of a nonstationary run of SWAN is by default a JONSWAP spectrum with a directional distribution centered around the local wind direction.

The user must provide the **computational spectral grid**. In frequency space it is defined by a minimum and maximum frequency and the frequency resolution, which is proportional to the frequency itself (e.g. $\Delta f = 0.1 f$). In the frequency domain the lowest frequency [fl], the highest frequency [fh] and the number of frequencies [nfreq] must be chosen (see command CGRID, **Section 4.12.6.1**). The value of [fl] must be somewhat smaller than 0.7 times the value of the lowest peak frequency expected. The value of [fh] must be at least 2.5 to 3 times the highest peak frequency expected; usually [fh] is chosen less than or equal to 1 Hz.

SWAN has the option to make computations that are nested in WAM or WAVEWATCH III. In such runs SWAN interpolates the spectral grid of WAM or WAVEWATCH III to the (user provided) spectral grid of SWAN. The WAM Cycle 4 source term in SWAN has been returned for a highest prognostic frequency (explicitly computed by SWAN) of 1 Hz. It is therefore recommended that for cases where wind generation is important and WAM Cycle 4 formulations are chosen, the highest prognostic frequency be 1 Hz. WAM Cycle 4 is not usually recommended. In directional space the directional range is the full 360° unless the user specifies a limited directional range. This may be more efficient (requiring less computer time and/or space) when waves travel toward a coast within a limited sector of 180°. The number of discrete directions that is provided by the user determines the directional resolution. For wind seas with a directional spreading of typically 30° on either side of the mean wave direction, a resolution of 10° is sufficient, whereas for swell with a directional spreading of less than 10°, a resolution of 2° or less may be required. If the user is confident that no energy will occur outside a certain directional sector (or is willing to ignore this energy), then the computations by SWAN can be limited to the directional sector that does contain energy. This may often be the case when waves propagate to shore within a sector of 180° around some mean wave direction.

Use the following discretization in SWAN for applications in coastal areas:

Table 4.7-1. Discretization in SWAN.

Direction resolution:			
Wind sea conditions:	$\Delta\theta$	=	15° - 10°
Swell conditions	$\Delta\theta$	=	5° - 2°
Frequency range:	f_{min}	=	0.04 Hz
	f_{max}	=	1.00 Hz
Spatial resolution: (Δx, Δy)	Δx, Δy	=	50 – 1000 m

The numerical schemes in the SWAN model require a minimum number of two discrete grid points in each spatial direction. If the ILU-CGSTAB solver is used (see command NUMERIC, **Section**

4.12.10.2), then the minimum number of directional bins is three per directional quadrant and the minimum number of frequencies will be four.

Interpolation of spectra

The interpolation of spectra in SWAN, both in space and time, is a slight modification of the procedure used in WAM. This procedure is not a simple (spectral) bin-by-bin interpolation because that would cause reduction of the spectral peak if the peaks of the original spectra do not coincide. It is an interpolation where the spectra are first normalized by average frequency and direction then interpolated and then transformed back.

The average frequencies of the two origin spectra are determined using the frequency moments of the spectra:

$$m_{i,k} = \int N_i(\sigma,\theta)\sigma^{\mathrm{k}} d\sigma\, d\theta\,,$$

with $i = 1, 2$ (the two origin spectra) and k = 0, 1 (the zero- and first frequency moments of these spectra). Then

$$\overline{\sigma} = m_{i,1} / m_{i,0}\,.$$

The average frequency for the interpolated spectrum is calculated as

$$\overline{\sigma} = (w_1 m_{1,1} + w_2 m_{2,1}) / (w_1 m_{i,0} + w_2 m_{2,0})\,,$$

where w_1 is the relative distance (in space or time) from the interpolated spectrum to the first origin spectrum $N_1(\sigma, \theta)$ and w_2 is the same for the second origin spectrum $N_2(\sigma, \theta)$. Therefore, $w_1 + w_2 = 1$.

The average directions of the two origin spectra are determined using directional moments of the spectra:

$$m_{i,x} = \int N_i(\sigma,\theta)\cos(\theta) d\sigma\, d\theta \qquad \text{and}$$

$$m_{i,y} = \int N_i(\sigma,\theta)\sin(\theta) d\sigma\, d\theta\,,$$

with $i = 1, 2$. The average direction is then

$$\overline{\theta}_{\mathrm{i}} = a\tan(m_{i,y} / m_{i,x})\,.$$

The average direction of the interpolated spectrum is calculated as

$$\bar{\theta}_i = a\tan[(w_1 m_{1,y} + w_2 m_{2,y})/(w_1 m_{1,x} + w_2 m_{2,x}).$$

Finally the interpolated spectrum is calculated as follows:

$$N(\sigma,\theta) = w_1 N_1[\bar{\sigma}_1 \sigma / \bar{\sigma}, \theta - (\bar{\theta} - \bar{\theta}_1)] + w_2 N_2[\bar{\sigma}_2 \sigma / \bar{\sigma}, \theta - (\bar{\theta} - \bar{\theta}_2)].$$

4.7.3 Output Grids

SWAN can provide output on uniform, rectilinear spatial grids that are independent of the input grids and of the computational grid. In a computation with a curvilinear computational grid, curvilinear output grids are available in SWAN but not for isoline or vector plots. An output grid must be specified by the user, but it is wise to choose a resolution that is fine enough to show relevant spatial details. It must be pointed out that the information on an output grid is obtained from the computational grid through bilinear interpolation (output always at computational time level). This implies that some inaccuracies are introduced by this interpolation. It also implies that bottom or current information on a (output) plot has been obtained by interpolating twice: once from the input grid to the computational grid and once from the computational grid to the output grid. If the input, computational and output grids are identical, then no interpolation errors occur.

In the regions where the output grid does not cover the computational grid SWAN assumes output values equal to the exception value; e.g. for Hs the exception value is -9. The exception values of output quantities can be changed by means of the QUANTITY command (see **Section 4.13.2.1**).

The user may reproduce the input fields (bathymetry etc.) with commands FRAME and PLOT... (See **Sections 4.13.1.1** and **4.13.2**).

In nonstationary computations output can be requested at regular intervals beginning with a given time and always at computational times.

4.8 Activating Physical Processes

SWAN contains a number of physical processes (see **Section 5.0**) that add or subtract wave energy to or from the wave field. The processes included are wind growth, whitecapping, bottom friction, depth-induced wave breaking, obstacle transmission, nonlinear wave-wave interactions (quadruplets and triads) and wave-induced set-up. SWAN can run in several modes, indicating the level of parameterization.

Wind growth*	Activated by commands GEN1, GEN2 or GEN3*.
Whitecapping*	Activated by commands GEN1, GEN2 or GEN3*.
Quadruplets*	Activated by command GEN3*.
Triads	Activated by command TRIAD.
Bottom friction	Activated by command FRICTION.
Depth-induced breaking*	Activated by command BREAKING*.

Obstacle transmission	Activated by command OBSTACLES.
Wave-induced set-up	Activated by command SETUP.

* Active by default, can be deactivated with command OFF.

For the first SWAN runs, it is strongly advised to use the default values of the model coefficients. First it should be determined whether or not a certain physical process is relevant to the result. If this cannot be decided by means of a simple hand computation, one can perform a SWAN computation with and without the physical process included in the computations, using the standard values chosen in SWAN.

After it has been established that a certain physical process is important, it may be worthwhile to modify coefficients. In the case of wind growth one may at first try to vary the wind velocity. In the bottom friction the best coefficient to vary is the friction coefficient. Switching off the depth-induced breaking term is usually unwise, since ii leads to unacceptably high wave heights near beaches (the computed wave heights 'explode' due to shoaling effects).

4.9 Print file and error messages

SWAN always creates a print file. The filename is usually identical to the name of the command file of the computations, with the extension (.SWN) replaced with (.PRT). Otherwise, the filename depends on the user's batch file.

The print file contains an echo of the command file and error messages. These messages are usually self-explanatory. If more assistance is required, users may address the SWAN discussion group on the SWAN homepage (http://swan.ct.tudelft.nl). Active WISE members may contact the authors of SWAN for assistance. The print file also contains computational results if the user requests them with either command BLOCK or command TABLE.

4.10 The Millennium Problem

SWAN can run in the nonstationary mode for dates between the years 0 and 9999 if ISO-notation is used in the input (recommended), or between the years 1911 and 2010 if two-digit code for years is used (formats 2-6 in every command that contains moments in time).

Be careful when nesting SWAN in WAM, as WAM does not use ISO-notation.

4.11 Input / Output and Storage

4.11.1 General

SWAN is one single computer program, consisting of one executable file (e.g., swan.exe), a command file (e.g. a file named "INPUT" or a file with extension .swn, see swan.edt, Appendix C), and a run file (swan.bat or swan.unix). The names of the files provided by the user should comply with the rules of file identification of the computer system on which SWAN is run (see **Section 4.12.3**). In addition, SWAN does not permit file names longer than 36 characters. Moreover, *the maximum length of the lines in the input files for SWAN is 120 positions.*

The user should provide SWAN with the following input information:

- A command file containing the user selected instructions to run SWAN,
- File(s) containing bottom, current, friction, and wind (if relevant),
- File(s) containing the wave field at the model boundaries (if relevant).

The input and output to a number of test problems is provided on the SWAN internet-site in the test subdirectory. The files with extension .swn are the command files for these tests. The files with extension .bot are the bottom files for these tests. These files can be used to make a configuration test of SWAN one's computer. Compare the results with those in the test documentation. **Note:** The results need not be identical up to the last digit. In some test cases SWAN may generate an error message concerning the size of the data pool. It explains how large the pool actually is and how large it should be for this calculation. There are two solutions to this problem, either enlarge the pool or reduce the grid resolution(s).

4.11.2 Input / Output Facilities

To assist in making the command file, an edit file is available to the user (swan.edt; see Appendix C). In its original form this file consists only of comments. In the edit file all commands in this user's manual are reproduced as a mnemonic for making the actual command file, so a user does not need to consult the user's manual every time to check the correct spelling of keywords, the order of data, etc. The user is advised to copy the swan.edt file (the copy file should have a different name) and then type in commands between the comment lines of the edit file.

SWAN is fairly flexible with regards to output. Output is available for many different wave and wave-related parameters such as wave-induced stresses or bottom orbital motion). However, SWAN produces output only at the user's request. The instructions for controlling output are separated into four categories:

- Definitions of the geographic location(s) of the output. Output locations may be either on a regular geographic grid, or along user specified lines (e.g. a given depth contour line) or at individual output locations.
- Times for which this output is requested (only in nonstationary runs).
- Type of output quantity such as waves, currents, etc.
- Medium to which output should be transferred (plot or disk).

Plot output is quick-look output only, used to show spatial variations of bottom, current or wave related parameters in rectangular geographic output regions and spectra in individual geographic points.

This information (and much more) is available in data files.

4.11.2.1 Output Files

SWAN will generate a number of output files:

- Print Files: Error messages appear in a file with the name PRINT, which can be renamed by the user with a batch (DOS) or script (Unix) command. In the DOS batch file and the Unix script on the SWAN web site the file PRINT is renamed to the name of the command file, with the extension .swn replaced by .prt. Files with extension .prt are also referred to as print files.

  SWAN always creates a print file. Usually the name of this file is identical to the name of the command file of the computations with the extension (.swn) replaced with (.prt). Otherwise, it depends on the user's batch file.

  The print file contains an echo of the command file and error messages as well as computational results if the user so requests (with command BLOCK or TABLE (see **Sections 4.13.2.2** and **4.13.2.3**)). These messages are usually self-explanatory (If not, active WISE members can contact the authors of SWAN). Other users should address the discussion group on the SWAN web site).

- Numerical Output: Output from commands such as BLOCK and TABLE appears in files with user provided names.

- Plot Files: the PLOT… commands generates one or more plot files. If the user does not specify a filename the plot file has the name PLF..., where the run number as defined in the command PROJECT as NR appears on the dots.

- Error Files: A file called Errfile containing the error messages is created only when SWAN produces error messages. Existence of this file is an indication that results must be carefully examined.

- Grid Point Error Files: A file called ERRPTS containing the grid-points, where specific errors occurred during the calculation, such as non-convergence of the iterative matrix-solver. Existence of this file is an indication to study that grid-point spectrum more carefully.

4.11.3 Storage Requirements

The core memory for SWAN is determined at the installation of SWAN on the user's computer system. Reinstalling SWAN with another size of core memory can change it. The user cannot altar the size of this memory but considering the choice for the computation of the nonlinear four wave-wave interactions can influence the efficiency of using it. Calculating these interactions per sweep, instead of per iteration, decreases the amount of required memory by a factor of 2/3 (see **Section 5.0** and command GEN3, **Section 4.12.9.3**), but increases the computing time. Calculation per sweep is the default in SWAN.

Only when SWAN is given a task that exceeds the available primary storage should the user attempt to use the available storage more efficiently. The largest of these requirements determines whether SWAN can carry out a task within the available space or not.

The required storage capacity in SWAN depends on the number of grid points in x- and y-direction (mxc*myc) and the number of points in frequency and directional space (msc*mdc). Another important restriction is given by the way the nonlinear four wave-wave interactions are calculated. A rough estimate of the required storage capacity is given in **Tables 4.11-1** and **4.11-2** below. **Note:** The required storage is primarily determined by four-dimensional arrays (e.g., A^t (x, y, σ, θ), A^{t+1} (x, y, σ, θ)).

Table 4.11-1. Estimate of required storage capacity of SWAN for Stationary Run (with default SORDUP scheme), or Stationary or Nonstationary Runs (with BSBT scheme with one update of the wave field per time step).

			Required memory floating point numbers single precision
Spectra:			mxc*myc*msc*mdc
Quadruplet interaction:			
Fully explicit	iquad in command GEN3	add	negligible
	360°-sector	add	mxc*myc*msc*mdc
Semi-implicit	90°-sector	add	negligible

Table 4.11-2. Estimate of required storage capacity of SWAN for Nonstationary Run (with default S&L scheme or with BSBT).

			Required memory floating point numbers single precision
Spectra:			2*mxc*myc*msc*mdc
Quadruplet interaction:			
Fully explicit	[iquad] = 2	add	negligible
	[iquad] = 3	add	mxc*myc*msc*mdc
Semi-implicit	[iquad] = 1	add	negligible

The value of [iquad] refers to implicit or explicit computations of the quadruplet wave-wave interactions (see command GEN3, **Section 4.12.9.3**, default [iquad] = 2). Some additional arrays are used for the bottom, current, wind, wave number, different propagation velocities and some auxiliary arrays, however, these are negligible compared to the rather large four-dimensional arrays mxc*myc*msc*mdc.

4.12 Description of Commands

4.12.1 List of Available Commands

The following commands are available to SWAN users.

<u>Start-up commands</u> **(Section 4.12.4)**
Block (a)

PROJECT	Title of the problem to be computed.
SET	Sets values of certain general parameters.
MODE	Requests a stationary / nonstationary or 1D-mode / 2D-mode of SWAN.
COORD	For choosing between Cartesian and spherical coordinates.

<u>General commands</u> **(Section 4.12.5)**
Block (b)

TEST	Requests the output of intermediate results for testing purposes.
POOL	Requests status of computer memory.

<u>Model Description Commands</u> **(Section 4.12.6)**
Block (c)
Computational grid commands (**Section 4.12.6.1**)

CGRID	Defines dimension of computational grid.
READGRID	Reads a curvilinear computational grid.

Block (d)
Input field commands (**Section 4.12.7**)

INPGRID	Defines dimension of bottom grid, etc.
READINP	Reads input fields.
WIND	Activates constant wind option.

Block (e)
Boundary/initial conditions (**Section 4.12.8**)

BOUND...	Defines boundary conditions for the computational spatial grid.
	BOUNDPAR1
	BOUNDPAR2
	BOUNDSPEC
	BOUNDNEST1
	BOUNDNEST2
	BOUNDNEST3
INITIAL	Initializes the wave field for a computation.

Block (f)
Physics commands (**Section 4.12.9**)

GEN	SWAN first-, second- or third-generation physics.	
	GEN1	SWAN runs in first-generation mode.
	GEN2	SWAN runs in second-generation mode.
	GEN3	SWAN runs in third-generation mode.
BREAKING	Activates dissipation by depth-induced wave breaking.	
FRICTION	Activates dissipation by bottom friction.	
TRIAD	Activates three wave-wave interactions.	
OBSTACLE	Defines characteristics of sub-grid obstacles.	
SETUP	Activates the computation of the wave-induced set-up.	
OFF	Deactivates certain physical processes.	

Block (g)
Numerics commands (**Section 4.12.10**)

PROP	To choose the numerical propagation scheme.
NUMERIC	Sets some of the numerical properties of SWAN

Output commands (**Section 4.13**)
Block (h)
Output location commands (**Section 4.13.1**)

FRAME	Defines an output frame (a regular grid).
GROUP	Defines an output group (for regular and curvilinear grids).
CURVE	Defines an output curve.
RAY	Defines a set of straight output lines (rays).
ISOLINE	Defines a depth or bottom contour for output along that contour.
POINTS	Defines a set of individual output points.
NGRID	Defines a nested grid.
LINE	Defines a set of lines in plots.
SITES	Defines a set of place names in plots.

Block (i)
Commands to write or plot output quantities (**Section 4.13.2**)

QUANTITY	Defines properties of output quantities.
BLOCK	Requests a block output (geographic distribution).
TABLE	Requests writing of a table (set of locations).
SPECOUT	Requests spectral output.
NESTOUT	Requests spectral output for subsequent nested computations.
PLOT...	Requests plotting of a figure.
	PLOTGEOGR
	PLOTSTAR
	PLOTPP
	PLOTSPEC

Compute, hotfile and stop commands (**Section 4.13.3**)
Block (j)
Compute, hotfile and stop commands

COMPUTE	Starts a computation.
HOTFILE	Stores results for subsequent SWAN run.
STOP	End of user's input.

4.12.2 Command Sequence

SWAN executes the above command blocks (a,...,j) in the above sequence except (b), (g) and (j). The commands of the blocks (b) and (g) may appear anywhere before block (j), except that TEST POINTS must come after READINP BOTTOM. The commands of block (j) may appear anywhere in the command file (all commands after COMPUTE are ignored by SWAN, except HOTFILE and STOP). A sequence of blocks (h) is permitted (all commands will be executed, without overriding). Also a sequence of blocks such as (i) is permitted (all commands will be executed, without overriding). Within the blocks the following sequence is to be used:

In block (a):	No prescribed sequence in block.
In block (b):	No prescribed sequence in block.
In block (c):	READGRID after CGRID.
In block (d):	READINP after INPGRID (repeat both in this sequence for each quantity).
In block (e):	BOUNDPAR1 before BOUNDPAR2, otherwise no prescribed sequence in block.
In block (f):	Use only one GEN command in command file. Use command OFF only after a GEN command (note that GEN3 is default).
In block (g):	No prescribed sequence in block.
In block (h):	ISOLINE after RAY (ISOLINE and RAY can be repeated independently).
In block (i):	No prescribed sequence in block.

In block (j): HOTFILE immediately after COMPUTE, STOP after COMPUTE.

Note: A repetition of a command may override an earlier occurrence of that command. Many commands provide the user with the opportunity to assign values to coefficients of SWAN (e.g. the bottom friction coefficient). If the user does not select such an option, SWAN will use a default value. Some commands cannot be used in 1-D mode (see individual command descriptions below).

4.12.3 Command Syntax and Input and Output Limitations

The command syntax is given in Appendix A.
Rules of file identification:

- The maximum length of the input lines is 120 characters.
- The maximum length of the file names is 36 characters.
- The maximum length of the plot titles is 36 characters.
- The maximum number of file names is 79. This can be extended (edit the file swaninit to change highest unit number from 99 to a higher number).
- The maximum number of plot files is 20. This can be extended (edit the file ocpids.for). This does not limit the number of plots, as a large number of plots can reside in one plot file.

4.12.4 Start-up Commands

4.12.4.1 PROJECT

```
PROJECT     'name'      'nr'

            'title1'

            'title2'

            'title3'
```

The PROJ command requires the user to define a number of strings to identify the SWAN run (project name e.g., an engineering project) in the print and plot files.

'name' Name of the project, at most 16 characters long.
Default: blanks.

'nr' Run identification (to be provided as a character string; e.g. the run number) to distinguish a run from other runs for the same project. It is at most four characters long. It is the only required information in this command.

'title1' String of at most 72 characters provided by the user to appear in the output of

the program for the user's convenience.
Default: blanks.

'title2' Same as 'title1'.

'title3' Same as 'title1'.

4.12.4.2 SET

```
SET   [level]   [nor]   [depmin]   [maxmes]   [maxerr]  [grav]   [rho]      &

                                  |  NAUTICAL  |
      [inrhog]  [hsrerr]   <                    >  [pwtail]  [froudmax]  [printf]  [prtest]
                                  | -> CARTESIAN |
```

With this optional command the user assigns values to various general parameters.

[level] Increase the water level that is constant in space and time. [level] is the value of this increase (in m). For a variable water level, reference is made to the commands INPGRID and READINP (See **Sections 4.12.7.1 and 4.12.7.2**). Default: [level] = 0.

[nor] Direction of north with respect to the x-axis (measured counter-clockwise); Default [nor] = 90, i.e., the x-axis of the problem coordinate system points east. When spherical coordinates are used (see command COORD, **Section 4.12.4.4**) the value of [nor] may not be modified.

[depmin] Threshold depth (in m). In the computation, any positive depth smaller than [depmin] is made equal to [depmin]. Default: [depmin] = 0.05.

[maxmes] Maximum number of error messages incurred (not necessarily the number of errors!) during the computation beyond which the computation is terminated. During the computational process messages are written to the print file, e.g., if the action density or the frequency in a point becomes negative. Such negative values are a sign that instabilities may have occurred). Default: [maxmes] = 200.
Note: When a number [maxmes] of error messages, have been written to the PRT file (see **Section 4.9**), the computation will stop. If necessary make [maxmes] larger using command SET, and rerun the program.

[maxerr] SWAN checks input data during pre-processing. Depending on the severity of the errors encountered, SWAN does not start computations. The user can influence the error level above which SWAN will not start

computations (the computations will continue at the level indicated). The error level [maxerr] is coded as follows:
= 1 Warnings;
= 2 Errors (possibly automatically repaired or repairable by SWAN);
= 3 Severe errors.
Default: [maxerr] = 1.

[grav]

Gravitational acceleration (in m/s^2).
Default: [grav] = 9.81.

[rho]

Water density (in kg/m^3).
Default: [rho] = 1025.

[inrhog]

Indicates whether the user requires output based on variance or based on true energy (See **Section 4.6**).
= 0 Output based on variance;
= 1 Output based on true energy.
Default: [inrhog] = 0.

[hsrerr]

Relative difference between the user-imposed significant wave height and the significant wave height computed by SWAN (anywhere along the computational grid boundary) above which a WARNING will be given. This relative difference normalized with the user-provided significant wave height. This warning will be given for each boundary grid point where the problem occurs along with the x- and y-index number of the computational grid (See **Section 4.13.2**). To suppress these warnings (in particular for nonstationary computations), set [hsrerr] at a very high value or use command OFF BNDCHK.

NAUTICAL

Indicates that the Nautical convention for wind and wave direction (SWAN input and output) will be used instead of the default Cartesian convention. (For definition see **Section 4.6**).

CARTESIAN

Indicates that the Cartesian convention for wind and wave direction (SWAN input and output) will be used. (For definition see **Section 4.6**).

[pwtail]

Power of high frequency tail. Defines the shape of the spectral tail above the highest prognostic frequency [fhigh] (see command CGRID, **Section 4.12.6.1**). The energy density is assumed to be proportional to frequency to the power [pwtail]. Default values depend on formulations of physics:

command GEN1,	[pwtail] = 5
command GEN2,	[pwtail] = 5
command GEN3 KOMEN,	[pwtail] = 4
command GEN3 JANSEN,	[pwtail] = 5

If the user wishes to use another value, then this SET command should be located in the command file after the GEN1, GEN 2 or GEN3 command (these will otherwise override the SET command as regards [pwtail]).

[printf] Unit reference number of the print file. Initially [printf] is equal to four. If it is changed to six, all print output would be written to the screen. This is useful if print output is lost due to an abnormal end of the program and information about the reason will print to file.

[prtest] Unit reference number of the test output file. Initially [prtest] is equal to four. If it is changed to six, all test output will be written to the screen. This is useful if test print output is lost due to abnormal end of the program, and information about the reason should appear in the test print file.

4.12.4.3 MODE

```
        | -> STATIONARY      |
MODE   <                      >            &
        |    NONSTATIONARY   |

        | -> TWODIMENSIONAL  |
       <                      >
        |    ONEDIMENSIONAL  |
```

With this optional command the user selects that the SWAN run be either stationary or nonstationary and one-dimensional (1-D mode) or two-dimensional (2-D mode). Selecting nonstationary allows for the following (see command COMPUTE, **Section 4.13.3.1**):

- one nonstationary computation,
- a sequence of stationary computations,
- a mix of (a) and (b).

The default option of MODE is STATIONARY TWODIMENSIONAL.

4.12.4.4 COORDINATES

```
               | -> CARTESIAN                    |
COORDINATES   <                  | -> CCM |       >  REPEATING
               |    SPHERICAL   <          >     |
                    |  QC  |
```

The command COORDINATES allows for choosing between Cartesian and spherical coordinates.

A nested SWAN run must use the same coordinate system as the coarse grid SWAN run.

CARTESIAN
All locations and distances are in m. Coordinates are given with respect to x-and y-axes chosen by the user in the various commands.

SPHERICAL
All coordinates of locations and geographical grid sizes are given in degrees. The variable x is longitude, x = 0 is the Greenwich meridian and x > 0 is east of this meridian. The variable y is latitude, y > 0 is the Northern Hemisphere. Input and output grids must be oriented with their x-axes to the east. Mesh sizes are in degrees. All other distances are in m.

CCM
The Central Conformal Mercator projection. This command defines the projection method in case of spherical coordinates. The horizontal and vertical scales are uniform over the area shown, measured in terms of cm/degree. The center (and only the center) of the scale is identical to that of the conventional Mercator projection. The area in the projection center is therefore exactly conformal.

QC
Quasi-Cartesian project method, where the horizontal and vertical scales are equal to one another in terms of cm/degree.

Note: SPHERICAL coordinates may also be used for relatively small areas, say 10 or 20 km horizontal dimension. This may be useful if one obtains the boundary conditions by nesting in an oceanic model that is naturally formulated in spherical coordinates.

Also note that for spherical coordinates regular grids must always be oriented E-W, N-S, using [alpc] = 0, [alpinp] = 0, [alpfr] = 0 (see commands CGRID, INPUT GRID, and FRAME, **Sections 4.12.6.1**, **4.12.7.1** and **4.13.1.1**).

4.12.5 General Commands

4.12.5.1 TEST

```
                                              | -> IJ  < [i] [j] > |
TEST      [itest]     [itrace]     POINTS     <                    >        &
                                              |   XY  < [x] [y] >  |

          (PAR `fname')  (S1D `fname')  (S2D `fname')
```

If SWAN produces unexpected results, this optional command can be used to instruct the program to produce intermediate results during a SWAN run (test output). A TEST command may alternate between commands in the file during the execution to change the level of test output during a SWAN run. A TEST command controls the test output until the next TEST command. Subsequent TEST commands may have a level 0, which will suppress the test output.

[itest]
The level of test output. For values under 100 the amount is usually reasonable, for values above 200 it can be very large. For values of [itest] up to 50 the user can interpret the test output. Higher values of [itest] result in output that can only be interpreted by those who have the program source listing at their disposal. Default: [itest] = 30

[itrace]
SWAN writes a message (name of subroutine) to the print file at the first [itrace] entries of each subroutine. Default: [itrace] = 0

POINTS
Produces a detailed print out during the model run for a grid point on the computational grid. Output at a maximum of 50 grid points is possible. This option can be used only after the bathymetry has been read (See command READINP BOTTOM, **Section 4.12.7.2**).

IJ
The test points are defined by means of grid indices.

[i], [j]
Grid indices of a test point. Values of [i] range between 0 and [mxc] (See command CGRID, **Section 4.12.6.1**), values of [j] between 0 and [myc] (inclusive).

XY
The test points are defined in terms of problem coordinates. SWAN will determine the nearest grid points. Output will be made for this selected grid point.

[x], [y]
Coordinates of a test point. Problem coordinates (in meters (m)) for Cartesian coordinates, or longitude and latitude (in degrees) for spherical coordinates. See command COORD, **Section 4.12.4.4**).

PAR
Integral parameters for test points. These are written: HSIGN, TM01, (for definitions see **Section 4.14**) and Swind, Swcap, Ssurf, Sfric, Snl3 and Snl4, which are the integrals over frequency and direction of the respective source terms of wind input, whitecapping, depth-induced breaking, bottom friction, absolute value of the triad wave-wave interactions, and absolute value of the quadruplet wave-wave interactions.

S1D
If the keyword S1D appears variance densities and six source terms* will be printed as 1-D spectral (frequency) output. The file definition is given in

Appendix D. The output will be made after every iteration for MODE STATIONARY and after every time step for MODE NONSTATIONARY (see command MODE, **Section 4.12.4.3**).

`fname' Name of the file to which the output is written.
Default: fname = SWSRC1D.

S2D If the keyword S2D appears, variance densities and six source terms* as computed will be printed as 2-D (frequency and direction) spectral output. The file description is given in Appendix D. The output will be made after every iteration for MODE STATIONARY and after every time step for MODE NONSTATIONARY (see command MODE, **Section 4.12.4.3**).

`fname' Name of the file to which the output is written. Default: fname = SWSRC2D.

* The source terms written due to the presence of the keyword S1D or S2D are source terms of variance density. The six source terms are: wind input, whitecapping, bottom friction, breaking, 3-wave interactions and 4-wave interactions.

4.12.5.2 POOL

```
POOL
```

This optional command can be used to obtain information on the numerical data pool, which contains almost all numerical data in SWAN. The program will answer by providing the amount of data in use when the POOL command is received by the program, the amount of data expected to be needed later in the run, and the actual size of the pool, which is determined at the implementation of the program; expressed in number of reals, number of bytes = reals x 4 or reals x 8. An example of the output follows:

```
Data pool size defined at installation:                                   2000000
Occupied memory after executing last command before this instance of POOL:  425230
Estimated additional memory needed after this instance of POOL:             125906
TOTAL estimated memory for this run:                                        551136
```

4.12.6 Computational Grid Commands

4.12.6.1 CGRID

```
        | -> REGULAR [xpc] [ypc] [alpc] [xlenc] [ylenc] [mxc] [myc] |
CGRID  <                                                             >   &
        |  CURVILINEAR [mxc] [myc] (EXCEPTION  [xexc] [yexc])        |
         | -> CIRCLE              |
        <                         > [mdc] [flow] [fhigh] [msc]
         |   SECTOR [dir1] [dir2] |
```

This required command allows the user to define the geographic location, size, resolution and orientation of the computational grid in the problem coordinate system (see **Section 4.7.2**) for a uniform, rectilinear computational grid or to define the size of a curvilinear grid (see **Section 4.6**). The origin of the grid and the direction of the positive x-axis may be chosen arbitrarily by the user. CGRID must be used for nested runs and for a nested run the geographic and spectral range (directional sector inclusive) and resolution may differ from the previous run (zeros are used outside these ranges).

REGULAR This option indicates that the computational grid is to be taken as uniform and rectangular.

CURVILINEAR This option indicates that the computational grid is to be taken as curvilinear. The user must provide the coordinates of the grid points with command READGRID.

[xpc] Geographic location of the origin of the computational grid in the problem coordinate system (x-coordinate, in m). (See command COORD, **Section 4.12.4.4**). Default: [xpc] = 0.0

[ypc] Geographic location of the origin of the computational grid in the problem coordinate system (y-coordinate, in m). (See command COORD, **Section 4.12.4.4**). Default: [ypc] = 0.0

[alpc] Direction of the positive x-axis of the computational grid in degrees (Cartesian convention). In 1-D mode, [alpc] should be equal to the direction [alpinp] (see command INPGRID, **Section 4.12.7.1**). Default: [alpc] = 0.0

[xlenc] Length of the computational grid in x-direction (in m). For spherical coordinates [xlenc] is in degrees.

[ylenc] Length of the computational grid in y-direction (in m). In 1-D mode, [ylenc] = 0. For spherical coordinates, [ylenc] is in degrees.

[mxc] Number of meshes in computational grid in x-direction for a uniform, rectilinear grid or x^*-direction for a curvilinear grid (this number is one less than the number of grid points in this domain), i.e., Δx = [xclen]/[mxc] for a uniform, rectilinear grid.

[myc] Number of meshes in computational grid in y-direction for a uniform, rectilinear grid or y^*-direction for a curvilinear grid (this number is one less

than the number of grid points in this domain), i.e., $\Delta y = [yclen]/[myc]$ for a uniform, rectilinear grid. In 1-D mode, [myc] should be [myc] = 0.

EXCEPTION

Only available in the case of a curvilinear grid. If certain grid points are to be ignored during the computations, such as land points that remain dry, (no flooding), thus saving computer time and memory, then this can be indicated by identifying the grid points in the file that contain the grid point coordinates. (See command READGRID, **Section 4.12.6.2** and as an alternative see command INPGRID BOTTOM, **Section 4.12.7.1**).

[xexc]

Indicates that a grid point is to be ignored in the computations. This value is provided by the user at the location of the x-coordinate considered in the file of the x-coordinates; see command READGRID, **Section 4.12.6.2**. The value is required if EXCEPTION is used.

[yexc]

Indicates that a grid point is to be ignored in the computation. This value is provided by the user at the location of the y-coordinate considered in the file of the y-coordinates; see command READGRID, **Section 4.12.6.2**. The value is required if EXCEPTION is used. Default: [yexc] = [xexc].

CIRCLE

Indicates that the spectral directions cover the full circle. This option is default.

SECTOR

Allows for only spectral wave directions in a limited directional sector to be considered. The range of this sector is given by [dir1] and [dir2]. If the quadruplet interactions are to be computed (see command GEN3, **Section 4.12.9.3**), then SECTOR should be 30° wider on each side than the directional sector occupied by the spectrum (everywhere in the computational grid). If not, then these computations will be inaccurate. If the directional distribution of the spectrum is symmetric around the center of the SECTOR, then the computed quadruplet wave-wave interactions are effectively zero in the 30° range on either end of the SECTOR. The problem can be avoided by not activating the quadruplet wave-wave interactions (use command GEN1 or GEN2) or, if activated (with command GEN3), by subsequently de-activating them with command OFF QUAD.

[dir1]

The direction of the right-hand boundary of the sector when looking outward from the sector (required for option SECTOR) in degrees.

[dir2]

The direction of the left-hand boundary of the sector when looking outward from the sector (required for option SECTOR) in degrees.

[mdc]

Number of meshes in θ-space. For CIRCLE, this is the number of subdivisions of the 360 degrees of a circle, so $\Delta\theta = [360°]/[mdc]$ is the spectral directional resolution. In the case of SECTOR $\Delta\theta$ = {[dir2] -

[dir1]}/[mdc]. If the ILU-CGSTAB solver is used (see command NUMERIC, **Section 4.12.10.2**), then the minimum number of directional bins is three per directional quadrant.

[flow] Lowest discrete frequency that is used in the calculation (in Hz).

[fhigh] Highest discrete frequency that is used in the calculation (in Hz).

[msc] One less than the number of frequencies. This defines the grid resolution in frequency-space between the lowest discrete frequency [flow] and the highest discrete frequency [fhigh]. This resolution is not constant, since the frequencies are distributed logarithmically: $f_{i+1} = \gamma f_i$, where γ is a constant. The value of [msc] depends on the frequency resolution, Δf, which the user requires. Since the frequency distribution on the frequency axis is logarithmic, the relationship is:

$$\Delta f = \left(\left(\frac{[fhigh[}{[flow]} \right)^{1/[msc]} - 1 \right) f$$

Conversely, if the user chooses the value of $\Delta f / f$, then the value of [nfreq] is given by:

If the ILU-CGSTAB solver is used (see command NUMERIC, **Section 4.12.10.2**) then the minimum number of frequencies is four. It must be observed that the DIA of the quadruplet interactions is based on a frequency resolution of $\Delta f / f = 0.1$. The actual resolution in the computations should therefore not deviate too much from this.

For illustration of the quantities [xpc], [ypc] and [alpc] see **Fig. 4.12-1** below.

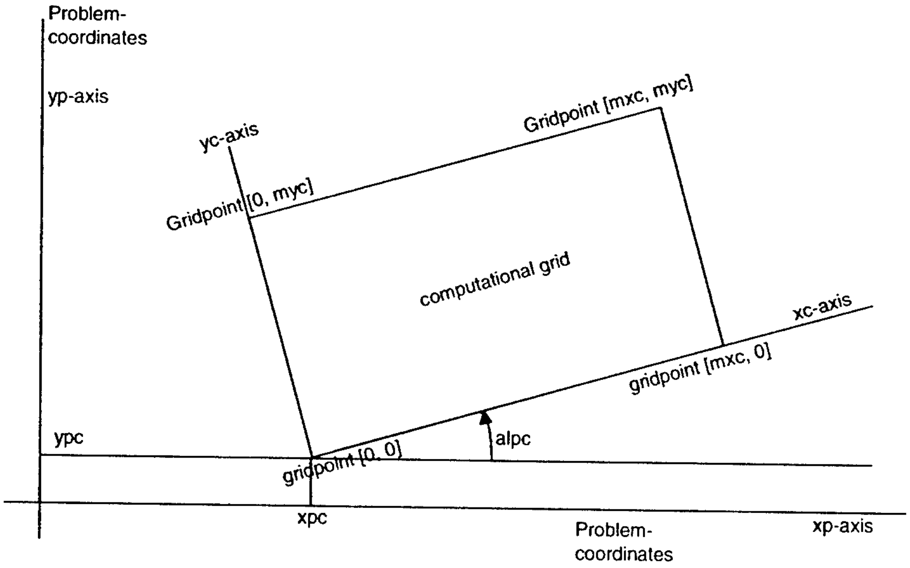


Fig. 4.12-1. Coordinates of the origin [xpc] and [ypc], the orientation [alpc], and the grid point numbering of the computational grid with respect to the problem coordinates system. Note that in case of spherical coordinates the xc- and xp-axes both point east.

4.12.6.2 READGRID COORDINATES

```
READGRID COORDINATES   [fac] 'fname' [idla] [nhedf] [nhedvec]        &

                        | -> FREE                          |
                        |                                  |
                        |                  | 'form' |      |
                       <   FORMAT         <          >     >
                        |                  | [idfm] |      |
                        |                                  |
                        |  UNFORMATTED                     |
```

READGRID COORDINATES requires that the user control the reading of the coordinates of the computational grid points. These coordinates must be read from a file as a vector (x-component, y-component = x-coordinate, y-coordinate of each single grid point). READGRID must follow a command CGRID CURVILINEAR. READGRID COORDINATES is required if the computational grid is curvilinear but not allowed for a regular grid. See command READINP in **Section 4.12.7.2** for the description of the READGRID options. READGRID cannot be used in 1-D mode.

4.12.7 Input Grids and Data Commands

4.12.7.1 INPGRID

```
          | BOTTOM          |
                   |                     |
                   | WLEVEL              |
                   |                     |
                   |  | CURRENT          |
                   | <                   |
                   |  | VX               |
                   |  | VY               |
                   |                     |
INPGRID          (<                      >)                                  &
                   | FRICTION            |
                   |                     |
                   |  | WIND             |
                   | <                   |
                   |  | WX               |
                   |  | WY               |

                 | -> REGULAR [xpinp] [ypinp] [alpinp] [mxinp] [myinp] [dxinp] [dyinp]   |
                <                                                                         > &
                 |    CURVILINEAR [stagrx] [stagry] [mxinp] [myinp]                      |

                 (EXCEPTION [excval])                                                    &
                                                             | -> SEC |
                 (NONSTATIONARY [tbeginp] [deltinp]         <     MIN  >  [tendinp])
                                                             |    HR  |
                                                             |    DAY |
```

OPTION CURVILINEAR IS NOT FOR 1-D MODE.

INPGRID is a required command in which the user defines the geographic location, size and orientation of an input grid as well as the time characteristics of the variable if it is not stationary. Non-stationary variables should be given in a sequence of fields, one for each time step. The actual reading of bottom level values, currents, etc., from the file is controlled by the command READINP. INPGRID must precede READINP.

Grids may vary for bottom level (BOTTOM), current (CURRENT), bottom friction coefficient (FRICTION) and wind velocity (WIND). If the current velocity components are available on different grids, then option VX, VY can define these different grids for the x- and y-component of the current (the grids must have identical orientation). Different grids for VX and VY may be useful if a current model using a staggered grid generates the data. The same holds true for wind velocity components. If INPGRID is given without any of the keywords BOTTOM, WIND etc. it is assumed that all the input grids are the same.

The current and wind vectors are defined with the x- and y-component of the current or wind vector with respect to the x-axis of the input grid. For wind velocity and friction coefficient it is also possible to use a constant value over the computational field (see commands WIND and FRICTION, **Sections 4.12.7.3** and **4.12.9.5**). No grid definition for wind and friction is required.

See **Section 4.7** for more information on grids.

BOTTOM

Defines the input grid of the bottom level. (For the definition of the bottom level see command READINP, **Section 4.12.7.2**).

WLEV

Water level relative to datum level, positive upward (in m).

CURRENT

Defines the input grid of the current field (same grid for x- and y-components).

VX

Defines the input grid of the x-component of the current field (different grid than y-component but same orientation).

VY

Defines input grid of the y-component of the current field (different grid than x-component but same orientation).

FRICTION

Defines input grid of the bottom friction coefficient (defined in command FRICTION, not to be confused with this option FRICTION).

WIND

Defines input grid of the wind field (same grid for x- and y-component). If neither of the commands WIND and READINP WIND is used it is assumed that there is no wind.

WX

Defines input grid of the x-component of the wind field (different grid than x-component but same orientation).

WY

Defines input grid of the y-component of the wind field (different grid than y-component but same orientation).

REGULAR

Means that the input grid is regular, rectangular.

CURVILINEAR

Means that the input grid is curvilinear; this option is available only if the computational grid is curvilinear as well. The input grid is identical (which is default) to the computational grid, or it is staggered in x- and/or y-direction. NOT for 1-D mode.

For a REGULAR grid:

[xpinp]

Geographic location (x-coordinate) of the origin of the input grid in problem coordinates in meters if Cartesian coordinates are used and degrees if spherical coordinates are used, (see command COORD,

Section 4.12.4.4). Default: [xpinp] = 0.

[ypinp] Geographic location (y-coordinate) of the origin of the input grid in problem coordinates in meters if Cartesian coordinates are used and degrees if spherical coordinates are used, (see command COORD, **Section 4.12.4.4**). Default: [ypinp] = 0.

[alpinp] Direction of the positive x-axis of the input grid (in degrees, Cartesian convention). (**Section 4.12.4.4**). Default: [alpinp] = 0.

[mxinp] Number of meshes in x-direction of the input grid (this number is one less than the number of grid points in this direction).

[myinp] Number of meshes in y-direction of the input grid (this number is one less than the number of grid points in this direction). In 1-D mode, [myinp] should be set equal to zero.

[dxinp] Mesh size in x-direction of the input grid, in meters for Cartesian coordinates, and degrees if spherical coordinates are used, (see command COORD, **Section 4.12.4.4**).

[dyinp] Mesh size in y-direction of the input grid, in meters for Cartesian coordinates and degrees if spherical coordinates are used, (see command COORD, **Section 4.12.4.4**). In 1-D mode, [dyinp] may have any value. Default: [dyinp] = [dxinp].

For a CURVILINEAR input, which has not been fully tested for spherical coordinates:

[mxinp] Number of meshes in x'-direction of the input grid (one less than the number of grid points in this direction). Default: [mxinp] = [mxc].

[myinp] Number of meshes in y'-direction of the input grid (one less than the number of grid points in this direction). Default: [myinp] = [myc].

[stagrx] Staggered x'-direction with respect to computational grid. Default: 0. **Note:** e.g. [stagrx] = 0.5 means that the grid points are shifted a half step in x'-direction; in many flow models x-velocities are defined in points shifted a half step in x'-direction.

[stagry] Staggered in y'-direction with respect to computational grid. Default: [stagry] = 0.
Note: e.g., [stagry] = 0.5 means that the grid points are shifted a half step in y'-direction; in many flow models y-velocities are defined in points shifted a half step in y'-direction.

EXCEPTION

If land points remain dry during the computations (no flooding), then these points can be ignored (saving computer time and memory) by giving bottom level values equal to [excval] in these grid points. NOT for 1-D mode.

[excval]

Exception value, which is required if EXCEPTION is used (see command EXCEPTION).

NONSTATIONARY

The variable (given in a time sequence of fields) is nonstationary. NOT for 1-D mode.

[tbeginp]

Begin time of the first field of the variable. The format is:

= 1	ISO notation	19870530.153000
= 2	(as in HP compiler):	`30-May-87 15:30:00'
= 3	(as in Lahey compiler):	05/30/87.15:30:00
= 4		15:30:00
= 5		87/05/30 15:30:00'
= 6	as in WAM	8705301530

This format is installation dependent. Refer to the installation manual on the SWAN web site. Default is ISO.

[deltinp]

Time interval between fields. The unit is indicated in the next option.

SEC

Unit seconds.

MIN

Unit minutes.

HR

Unit hours.

DAY

Unit days.

[tendinp]

End time of the last field of the variable. The format is:

= 1	ISO notation	19870530.153000
= 2	(as in HP compiler):	`30-May-87 15:30:00'
= 3	(as in Lahey compiler):	05/30/87.15:30:00
= 4		15:30:00
= 5		`87/05/30 15:30:00'
= 6	as in WAM	8705301530

This format is installation dependent. Refer to the installation manual on the SWAN web site. Default is ISO.

4.12.7.2 READINP

```
            | BOTTOM     |
            |            |
            | WLEVEL     |
            |            |
            | CURRENT    |
            |            |       | 'fname1'          |
READINP   < WIND         > [fac] <                   > [idla]    &
            |            |       | SERIES 'fname2'   |
            | FRICTION   |
                                 | -> FREE                         |
                                 |                                 |
                                 |                    | 'form' |   |
           [nhedf] ([nhedt]) ([nhedvec]) <  FORMAT    <        >   >
                                 |                    | [idfm] |   |
                                 |                                 |
                                 |  UNFORMATTED                    |
```

With this required command the user controls the reading of values of the indicated variables from file. Command READINP must follow command INPGRID.

If the variables are in one file, then the READINP commands should be given in the same sequence in which the variables appear in the file.

BOTTOM With this option, the user indicates that bottom levels (m) are to be read from file (bottom level positive downward relative to an arbitrary horizontal datum level). The sign of the input can be changed with option [fac] = -1. (See below).

WLEV With this option, the user indicates that water levels (m) are to be read from file (water level positive upward relative to the same datum level as used in option BOTTOM). Sign of input can be changed with option [fac] = -1. If the water level is constant in space and time, use the command SET to add this water level to the water depth.

CURRENT Rectilinear (curvilinear) input grid. With this option, the user indicates that the x- and y-components are to be read from one and the same READINP command. Under this option SWAN reads the x-components (x^*-components) first, and then the y-components (y^*-components), (see below). x^*- and y^*-components are taken along the directions of the grid lines of the curvilinear grid! If the current velocity is relatively large, e.g., the Froude number $U / \sqrt{gd}$ is larger than 0.8, it will be reduced such that the Froude number becomes equal to 0.8.

FRICTION

With this option the user indicates that friction coefficient is to be read from file for Collins: cfw and for Madsen: Kn, no space- or time-variable coefficient for the JONSWAP expression (see command FRICTION, **Section 4.12.9.5**). If the coefficients are constant in space and time, see command FRICTION.

WIND

Rectilinear (curvilinear) input grid: with this option the user indicates that the x- and y-component (x^*- and y^*-component) are to be read from one and the same file (with one READINP command). Under this option SWAN reads first all x-components (x^*-components), and then all y-components (y^*-components), (see below). x^*- and y^*-components are taken along the directions of the grid lines of the curvilinear grid. If the wind is constant (see command WIND, **Section 4.12.7.3**).

[fac]

SWAN multiplies all values that are read from file with FAC. For instance, if the bottom levels are given in unit decimeter, one should make [fac] = 0.1 to obtain levels in m. To change sign of bottom level use a negative value of [fac]. Default: [fac] = 1.

'fname1'

Name of the file with the values of the variable.

SERIES

With this option (only for mode NONSTATIONARY) the user indicates that the names of the files containing the nonstationary variable(s) are located in a separate file with name 'fname2'.

'fname2'

Contains the names of the files where the variables are given. These names are to be given in proper time sequence. SWAN reads the next file when the previous file end has been encountered. In these files the input should be given in the same format as in the above file 'fname1' (that implies that a file should start with the start of an input time step).

[idla]

Prescribes the order in which the values of bottom levels and currents should be given in the file.

= 1 SWAN reads the map from left to right starting in the upper-left-hand corner of the map (it is assumed that the x-axis of the grid is pointing to the right and the y-axis upwards). A new line in the map should start on a new line in the file.

= 2 As [idla] = 1 but a new line in the map need not start on a new line in the file.

= 3 SWAN reads the map from left to right starting in the lower-left-hand corner of the map. A new line in the map should start on a new line in the file.

= 4 As [idla] = 3 but a new line in the map need not start on a new line in the file.

= 5 SWAN reads the map from top to bottom starting in the lower-left-hand corner of the map. A new column in the map should start on a new line in the file.

= 6 As [idla] = 5 but a new column in the map need not start on a new line in the file. (Default: [idla] = 1).

[nhedf]

The number of header lines at the start of the file. The text in the header lines is reproduced in the print file created by SWAN (see **Section 4.11.2.1**). The file may begin with more header lines than [nhedf] because the start of the file is often also the start of a time step and possibly also of a vector variable (each having header lines, see below, [nhedt] and [nhedvec]). Default: [nhedf] = 0.

[nhedt]

Number of header lines in the file at the start of each time level. **Note:** Used only if variable is time-dependent. A time step may start with more header lines than [nhedt] because the variable may be a vector variable that has its own header lines (see below [nhedvec]). Default: [nhedt] = 0.

[nhedvec]

For each vector variable: number of header lines in the file at the start of each component (e.g., x- or y-component). Default: [nhedvec] = 0.

FREE

With this option the user indicates that the values are to be read with free format. Free format is a standard of FORTRAN. The free format conventions in reading from a file are almost the same as the conventions for the command syntax given in Appendix A. The most important differences are:

1. There are no continuation marks. Reading continues until the required number of data has been read or until a slash (/) is encountered.
2. Input lines can be longer than eighty characters (depending on the operating system of the computer).
3. Comments are not allowed.

With free format empty fields, repetition factors, and closure of a line by a slash, may be used.

FORMAT

Indicates that fixed format (FORTRAN convention) is to be used when reading the values from file. The format can be defined either by giving the format number [IDFM] or the format string 'FORM'.

'form' A user-specified format string according to FORTRAN convention, e.g. '((10X, 12F5.0))'.

[idfm] This format number is interpreted as follows:

= 1 Format according to BODKAR convention (a standard of the Ministry of Transport and Public Works in the Netherlands); Format string: (10X, 12F5.0).

= 5 Format (16F5.0), i.e. an input line consists of 16 fields of five places each.

= 6 Format (12F6.0), i.e. an input line consists of 12 fields of six places each.

= 8 Format (10F8.0), i.e. an input line consists of 10 fields of eight places each.

UNFORMATTED A form of reading without conversion (binary files). Not recommended for ordinary use.

If the file does not contain a sufficient number of data (i.e. less than the number of grid points of the input grid), SWAN will write an error message to the print file. If [itest] > 0 (see command TEST, **Section 4.12.5.1**), SWAN will reproduce the data in the print file using the layout according to [idla] = 1. The echo of the data to print file is also made if the READINP command is embedded between two TEST commands in the command file as follows:

```
TEST 120
READINP ....
TEST 0
```

To obtain only contour line plots or vector plots of input quantities (see command PLOTGEOGR, **Section 4.13.2.6**) without computations, e.g. to verify input, one should run SWAN with [itermx] = 0 (with command NUMERIC).

4.12.7.3 WIND

```
WIND [vel] [dir]
```

With this optional command the user indicates that the wind is constant.

[vel] Wind velocity at 10-m elevation (m/s).

[dir] Wind direction at 10-m elevation (in degrees, Cartesian or Nautical convention. See command SET, **Section 4.12.4.2**)

Both quantities are required if this command is used. Note that SWAN converts U_{10} to U_* (see **Section 5.1.4**).

4.12.8 Boundary and Initial Conditions Commands

4.12.8.1 BOUNDPAR1

```
                              |-> JONSWAP [gamma]   |
                              |                     |    |-> PEAK |
BOUNDPAR1 SHAPESPEC        <      PM                >   <          >    &
                              |                     |    |  MEAN  |
                              |   GAUSS [sigfr]     |
                              |                     |
                              |   BIN               |

              DSPR            |-> POWER          |
                           <                     >
                              |   DEGREES        |
```

This command BOUNDPAR1 SHAPESPEC defines the shape of the spectra (both in frequency and direction) at the boundary of the computational grid in case of parametric spectral input (see command BOUNDPAR2, **Section 4.12.8.2**).

JONSWAP JONSWAP spectrum will be used.

[gamma] Peak enhancement parameter of the JONSWAP spectrum. Default: [gamma] = 3.3.

PM Pierson-Moskowitz spectrum will be used.

GAUSS A Gaussian-shaped frequency spectrum will be used.

BIN Energy is located in one frequency bin (the frequency bin closest to the PER value of command BOUNDPAR2).

[sigfr] Width of the Gaussian frequency spectrum expressed as a standard deviation in Hz.

PEAK Default: the peak period (for definition see **Section 4.14**) is used as characteristic wave period.

MEAN Tm01 (for definition see **Section 4.14**) is used as the characteristic wave period.

DSPR Option for expressing the width of the directional distribution (the distribution function itself is $\cos^m(\theta-\theta_{peak})$.

POWER — Default: The directional width is expressed with the power m itself. **Note:** The directional resolution should accommodate the directional width; see command CGRID, **Section 4.12.6.1**).

DEGREES — The directional width is expressed in terms of the directional standard deviation of the $\cos^m(\theta-\theta_{peak})$ distribution (for definition see **Section 4.14**). **Note:** The directional resolution should accommodate the directional width; see command CGRID, **Section 4.12.6.1**.

If this command is not used, the JONSWAP option will be used by SWAN with [gamma] = 3.3 and POWER for the directional width.

4.12.8.2 BOUNDPAR2

```
                            | NORTH    |
                            | NW       |
                            | WEST     |
                            | SW       |    | -> CCW      |
              | -> SIDE     < SOUTH    >   <               >   |
              |             | SE       |    | CLOCKWISE |      |
              |             | EAST     |                       |
              |             | NE       |                       |
BOUNDPAR2   <                                                  >   &
                            | -> XY      <  [x]  [y]   > |     |
              | SEGMENT   <                                >   |
                            |   IJ       <  [i]  [j]   > |

              | CONSTANT PAR   [hs]  [per]  [dir]  [dd]        |
              |                                                |
            <                                                  >
              |                                                |
              | VARIABLE PAR < [len]  [hs]  [per]  [dir]  [dd] >   |
```

Command BOUNDPAR2, consisting of two parts, defines parametric spectra at the boundary. The first part defines the boundary side or segment where the spectra will be given. The second defines the spectral parameters of these spectra. Only the incoming wave components of these spectra are used by SWAN. The fact that complete spectra are calculated at the model boundaries from the spectral parameters should not be misinterpreted. Only the incoming components are effective in the computation.

There are two ways to define the part of the boundary at which the spectra are imposed. The first way (SIDE) is easiest if the boundary is one full side of the computational grid. The second

(SEGMENT) can be used if the boundary segment goes around the corner of the grid, or if the segment is only part of one side of the grid.

Command BOUNDPAR2 can be given a number of times, such as to define incident wave fields on various sides or segments of the boundary. One BOUNDPAR2 command can be used for only one side or one contiguous segment.

SIDE

The boundary is one full side of the computational grid (in 1-D cases either of the two ends of the 1-D grid).

NORTH, ...

Indicates on which side the boundary condition is applied; N means the boundary is the north edge (if present) of the computational area, likewise for W (west), S (south), E (east), NW (northwest), NE (northeast), SW (southwest) and SE (southeast). The side does not have to face exactly the given direction because the nearest direction of the normal to the side is taken. For curvilinear grids the direction is determined as the normal to the sum of the vectors joining the grid points on the boundary. If there is an interruption in the boundary due to the occurrence of exception values then the interruption is ignored in the summation. **Note**: For Cartesian coordinates the direction of the problem coordinate system must be defined by the user (see command SET..[north], **Section 4.12.4.2**). Default: the positive x-axis points east.

CCW, CLOCKWISE

See description of [len] below; these options are only effective if the option VARIABLE is used.

SEGMENT

SEGMENT is used if SIDE is not, i.e. either the boundary segment goes around a corner of the grid, or the segment is only part of one side of the grid. The distance along the segment (see [len] below) is measured from the first point of the segment (see XY or IJ).

XY

The segment is defined by a series of points of problem coordinates; these points do not have to coincide with grid points. The (straight) line connecting two points must be close to grid lines of the computational grid (the maximum distance is one tenth of the length of the straight line). This option is default.

[x], [y]

Problem coordinates of a point of the boundary segment (see command COORD, **Section 4.12.4.4**).

IJ

The segment is defined by means of a series of computational grid points given in terms of grid indices (origin at 0,0); not all grid points on the segment have to be mentioned. If two points are on the same grid line, intermediate points are assumed to be on the segment as well.

[i], [j]

Grid indices of a point of the segment. Values of [i] range between 0 and [mxc] (see command CGRID, **Section 4.12.6.1**), values of [j] between 0 and [myc] (inclusive).

CONSTANT

With this option the wave spectra are constant along the side or segment.

VARIABLE

With this option the wave spectra can vary along the side or segment. The incident wave field is prescribed at a number of points of the side or segment. Their distance from the starting point of the side or segment characterizes the points. SWAN calculates the wave spectra for grid points on the boundary of the computational grid by the spectral interpolation technique described in **Section 4.7.2**.

PAR

The wave spectra are defined by means of the following spectral parameters (see command BOUNDPAR1, **Section 4.12.8.1** for spectral shape).

[hs]

The significant wave height (in m).

[per]

The characteristic period of the energy spectrum (relative frequency; which is equal to absolute peak frequency in the absence of currents). [per] is the value of the peak period (in s) if option PEAK is chosen in command BOUNDPAR1. [per] is the value of the mean period if option MEAN is chosen in command BOUNDPAR1.

[dir]

The peak wave direction (θ_{peak}, direction in degrees; constant over frequencies).

[dd]

Coefficient of directional spreading; a $\cos^m(\theta)$ distribution is assumed. [dd] is interpreted as the directional standard deviation in degrees if the option DEGREES is chosen in the command BOUNDPAR1.
[dd] is interpreted as the power m if the option POWER is chosen in the command BOUNDPAR1.

[len]

The distance from the first point of the side or segment to the point along the side or segment for which the incident wave spectrum is prescribed. **Note:** The points do not have to coincide with grid points of the computational grid. [len] is the distance in m, not in grid steps. The values of [len] should be given in ascending order. The length along a SIDE is measured in clockwise or counterclockwise direction, depending on the options CCW or CLOCKW. The option CCW is default. For a SEGMENT the length is measured from the indicated begin point of the segment.

4.12.8.3 BOUNDSPEC

```
                                |NORTH        |
                                |NW           |
                                |WEST         |
                                |SW           |    |-> CCW      |
                    |-> SIDE    < SOUTH       >   <              >   |
                    |           |SE           |    |CLOCKWISE|       |
                    |           |EAST         |                      |
                    |           |NE           |                      |
BOUNDSPEC  <                                                          >    &
                    |           |-> XY        < [x] [y]   > |        |
                    |SEGMENT    <                             >      |
                                |   IJ        < [i] [j]   > |

                    |CONSTANT FILE   'fname'  [seq]             |
                    |                                           |
                   <                                             >
                    |                                           |
                    |VARIABLE FILE < [len]  'fname'  [seq] >    |
```

Command BOUNDSPEC, which consists of two parts, defines spectra at the boundary. The first part defines the boundary side or segment where the spectra will be given. The second defines the source of these spectra (files). Note that only the incoming wave components of these spectra are used by SWAN. The fact that complete spectra are calculated from the spectral parameters should not be misinterpreted. Only the incoming components are effective in the computation.

This BOUNDSPEC command can be given a number of times; i.e. to define incident wave fields on various segments of the boundary. One BOUNDSPEC command can be used for only one contiguous segment.

SIDE See command BOUNDPAR2, **Section 4.12.8.2**.

NORTH, ... See command BOUNDPAR2, **Section 4.12.8.2**.

CCW, CLOCKWISE See command BOUNDPAR2, **Section 4.12.8.2**.

SEGMENT See command BOUNDPAR2, **Section 4.12.8.2**.

XY See command BOUNDPAR2, **Section 4.12.8.2**.

[x], [y] See command BOUNDPAR2, **Section 4.12.8.2**.

IJ See command BOUNDPAR2, **Section 4.12.8.2**.

[i], [j]	See command BOUNDPAR2, **Section 4.12.8.2**.

CONSTANT	See command BOUNDPAR2, **Section 4.12.8.2**.

VARIABLE	See command BOUNDPAR2, **Section 4.12.8.2**.

FILE	The incoming wave data are read from a file. There are three types of files:

1. TPAR files containing nonstationary wave parameters,
2. Files containing stationary or nonstationary 1-D spectra (usually from measurements),
3. Files containing stationary or nonstationary 2-D spectra (from other computer programs or other SWAN runs).

A TPAR file is used for only one location. The string TPAR is on the first line of the file and subsequent lines each contain five numbers including Time (ISO notation), Hs, Period (average or peak period depending on the choice given in command BOUND SHAPE), Peak Direction (Nautical or Cartesian, depending on command SET), Directional spread (in degrees or as power of Cos depending on the choice given in command BOUND SHAPE).

Example of a TPAR file:

```
TPAR
19920516.1300  4.2   12.    -110.   22.
19920516.1800  4.2   12.    -110.   22.
19920517.0000  1.2   8.     -110.   22.
19920517.1200  1.4   8.5    -80.    26
19920517.2000  0.9   6.5    -95.    28
```

The structure of the files containing 1-D or 2-D spectra is described in Appendix D (there is no relation to the definition of the boundary file generated by WAM or WAVEWATCH III). 1-D and 2-D files can be used for stationary and nonstationary boundary conditions and for one or more locations. The spectral frequencies (and directions in the case of a 2-D spectrum) do not have to coincide with the frequencies and directions used in the present SWAN run. In a nested run, however, SWAN will interpolate to these frequencies and directions. The coordinates of locations in the 1-D and 2-D files are ignored when SWAN reads this file (SWAN uses the geographical information in the BOUNBSPEC command instead).

'fname'	Name of the file containing the boundary condition.

[seq]	Identification number of geographic location in the file (see Appendix D).

It is useful for files that contain spectra for more than one location. Default: [seq] = 1.
Note: A TPAR file always contains only one location, therefore, [seq] must always be one.

[len] See command BOUNDPAR2, **Section 4.12.8.2**.

4.12.8.4 BOUNDNEST1

```
                                   | -> CLOSED |
BOUNDNEST1   NEST  'fname'    <                >
                                   |   OPEN    |
```

With this optional command a nested SWAN run can be completed with the boundary conditions obtained from a coarse grid SWAN run, generated in a previous SWAN run with command NESTOUT. This should not to be confused with the BOUNDNEST1 option NEST. For a nested SWAN run the CGRID command is given to define the computational grid prior to giving the BOUNDNEST1 command. The computational grid for SWAN in geographic space is the area bounded by the SWAN coarse run nest (SWAN boundary points of the nest). This implies that the boundaries of the SWAN coarse run nest and the boundaries of the SWAN nested computational area should be almost identical (see below). The spectral frequencies and directions of the coarse grid run do not have to coincide with the frequencies and directions used in the nested SWAN run (for definition see command CGRID, **Section 4.12.6.1**). SWAN will interpolate to these frequencies and directions in the nested run (see **Section 4.7.2**).

Use command NGRID to generate the nest boundary in the coarse grid run. For the nested run, use the command CGRID with identical geographical information except the number of meshes, which will be much higher for the nested run.

A nested SWAN run must use the same coordinate system as the coarse grid SWAN run. A curvilinear grid may be used in the nested grid but the boundaries of this nest should conform to the rectangular course grid nest boundaries.

NEST Indicates that the boundary conditions are to be retrieved from a file created by a previous SWAN run (the present SWAN run is a nested run). The spectral frequencies (and directions, for a 2-D spectrum) of the previous run do not have to coincide with the frequencies and directions used in the present SWAN run (see command CGRID, **Section 4.12.6.1**). SWAN will interpolate the energy densities to these frequencies and directions (see **Section 4.7.2**).

'fname' Name of the file containing the boundary conditions for the present run, created by the previous SWAN coarse grid run. This file is structured according to the rules given in Appendix D for 2-D spectra.

CLOSED The boundary represented in the file is a closed rectangle. This is always the case if the NESTOUT command was used to generate the boundary condition file.

OPEN The boundary represented in the file is not a closed rectangle. This is always the case if the NESTOUT command was used to generate the boundary condition file.

If the user wishes to view the location of the nested grid, e.g. in relation to the bathymetry (defined by 'sname' = 'BOTTGRID'), the command PLOTGEOGR should be used. If the name of the nested grid is 'sname' = 'NG', the command should be:

PLOTGEOGR 'BOTTGRID' ISO BOTTOM LOCATION 'NG' (see command PLOTGEOGR, **Section 4.13.2.7**).

The BOUNDNEST command is not available for 1-D computations. In such cases, the commands SPECOUT and BOUNDSPEC may be used for the same purpose. Use command CGRID to define the computational grid.

4.12.8.5 BOUNDNEST2 WAMNEST

```
                                                  |-> CRAY   |
                                   | UNFORMATTED <            >   |
                                   |              | WKSTAT |      |
                                   |                              |
BOUNDNEST2 WAMNEST 'fname'        <                                > [xgc]  [ygc]
                                   |                              |
                                   | FREE                         |
```

BOUNDNEST2 WAMNEST is an optional command (not fully tested) in which a nested SWAN run can be carried out with the boundary conditions obtained from a coarse grid WAM Cycle 3 and 4 run. For a nested SWAN run the user must give the CGRID command to define the computational grid before using BOUNDNEST2. The computational grid for SWAN in geographic space is the area bounded by the WAM coarse run nest (WAM boundary points of the nest). This implies that the boundaries of the WAM nest and the boundaries of the SWAN computational area should be (nearly) identical (see below). The spectral frequencies and directions of the coarse grid run do not have to coincide with the frequencies and directions used in the nested SWAN run (CGRID). SWAN will interpolate to these frequencies and directions in the nested run (see **Section 4.7.3**).

Note: SWAN will accept a WAM output location only if the SWAN grid point on the nest boundary lies within a rectangle between two consecutive WAM output locations. The width between the output locations must be equal to 0.1 times the distance on either side of the line between the WAM output locations.

Option BOUNDNEST is not available for 1-D computations.

A nested SWAN run may use either Cartesian or spherical coordinates. A curvilinear grid may be used in the nested grid but the boundaries of this nest should conform to the rectangular course grid nest boundaries.

WAM output files are unformatted (binary). This implies that WAM and SWAN must run on the same computer. The option FREE is available in command BOUNDNEST when WAM and SWAN run on different types of machines because binary files do not transfer properly. The distributed version of WAM does not support the required free format nesting output. WAM users must modify WAM so that WAM output files can be read in free format, i.e. with at least a blank or comma between numbers.

'fname' A file name that contains all WAM filenames containing the nested boundary conditions in time-sequence (usually one file per day). For example, the contents of 'fname' may appear as:

```
CBO9212010000
CBO9212020000
CBO9212030000
....
etc.
```

SWAN will read the boundary data from these WAM files one after the other.

UNFORMATTED Indicates that the WAM files are binary.

CRAY Input will be read from file created by the CRAY version of WAM.

WKSTAT Input will be read from file created by the WORKSTATION version of WAM.

FREE The user indicates that the WAM files may be read with free format. These files are not generated standard by WAM.

[xgc] If SWAN is used with Cartesian coordinates [xgc] is the longitude of the southwest corner of the SWAN computational grid (in degrees). If the southwest corner of the nest in the WAM computation is on land this value is required. SWAN ignores [xgc] if it is used with spherical coordinates. Default: The location of the first spectrum encountered in the nest file.

[ygc] If SWAN is used with Cartesian coordinates [ygc] is the latitude of the southwest corner of SWAN computational grid (in degrees). If the

southwest corner of the nest in the WAM computation is on land this value is required. SWAN ignores [ygc] if it is used with spherical coordinates. Default: The location of the first spectrum encountered in the nest file.

4.12.8.6 BOUNDNEST3 WWIII

```
                                   | -> CLOSED   |
BOUNDNEST3   WWIII   'fname'   <                  >  [xgc] [ygc]
                                   |    OPEN     |
```

With this optional command (not fully tested) a nested SWAN run can be carried out with the boundary conditions obtained from a coarse grid WAVEWATCH III run. For this nested SWAN run the user must give the CGRID command before the BOUNDNEST3 command. The computational grid for SWAN in geographic space is the area bounded by the WAVEWATCH III nest (WAVEWATCH III boundary points of the nest). Therefore the boundaries of the WAVEWATCH III nest and the SWAN computational area should be (nearly) identical (see below). The spectral frequencies and directions of the coarse grid run do not have to coincide with the frequencies and directions used in the nested SWAN run (CGRID command); SWAN will interpolate to these frequencies and directions in the nested run (see **Section 4.7.2**).

The output files of WAVEWATCH III (Version 1.18, as distributed by NOAA) must be created with the WAVEWATCH III post-processor as output transfer files with

WW_3 OUTP (output type 1 sub type 3)

at the locations along the nest boundary (i.e. computational grid points in WAVEWATCH III). These locations are equal to the corner points of the SWAN nested grid and are optionally distributed between the nested SWAN corner points. The boundary of the WAVEWATCH III nested grid need not be closed and may cover land. WAVEWATCH III will output the locations in sequence, moving along the nest boundary either clockwise or counter-clockwise). SWAN will accept WAVEWATCH III output locations only if the SWAN nested boundary grid lies within a rectangle between two consecutive WAVEWATCH III output locations with a width equal to 0.1 times the distance between these output locations on either side of the line between these WAVEWATCH III output locations.

A nested SWAN run may use either Cartesian or spherical coordinates. A curvilinear grid may be used in the nested grid but the boundaries of this nest should conform to the rectangular course grid nest boundaries.

'fname' Name of the file that contains the spectra computed by WAVEWATCH III.

CLOSED The boundary condition represented in the file is defined on a closed rectangle.

OPEN
The curve on which the boundary condition is given, is not closed.

[xgc]
If SWAN is used with Cartesian coordinates, [xgc] is the longitude of the southwest corner of SWAN computational grid (in degrees). This value is required.
If SWAN is used with spherical coordinates, SWAN ignores [xgc].
Default: The location of the first spectrum encountered in the nest file.

[ygc]
If SWAN is used with Cartesian coordinates, [ygc] is the latitude of the southwest corner of SWAN computational grid (in degrees). This value is required.
If SWAN is used with spherical coordinates; SWAN ignores [ygc].
Default: The location of the first spectrum encountered in the nest file.

4.12.8.7 INITIAL

```
               | -> DEFAULT                |
               |                           |
               |   ZERO                    |
INITIAL   <                                 >
               |   PAR [hs] [per] [dir] [dd]   |
               |                           |
               |   HOTSTART `fname'        |
```

This command can be used to specify the initial values for a stationary (INITIAL HOTSTART only) or nonstationary computation. The initial values thus specified override the default initialization (see **Section 4.7.2**). It is possible to obtain an initial state by carrying out a previous stationary or nonstationary computation.

DEFAULT
The initial spectra are computed from the local wind velocities using the deep-water growth curve of Kahma and Calkoen (1992) cut off at values of significant wave height and peak frequency from Pierson and Moskowitz (1964). The average spatial step size over the model area is used as fetch with local wind. The shape of the spectrum is default JONSWAP with a $\cos^2$ directional distribution (options are available: see command BOUNDPAR1 SHAPE, **Section 4.12.8.1**).

ZERO
The initial spectral densities are all 0; note that if waves are generated in the model only by wind, waves can become non-zero only by the presence of the "A" term in the growth model (see the keyword AGROW in command GEN3 **Section 4.12.9.3**).

PAR — The spectra in the entire computational area are generated from integral parameters [hs], etc., in the same way as done for the boundary using the command BOUNDPAR2.

[hs] — The significant wave height.

[per] — Characteristic wave period of the energy spectrum (either peak or mean period, as determined by the options PEAK and MEAN in the command BOUNDSPEC SHAPE).

[dir] — The peak wave direction (direction in degrees, Nautical or Cartesian convention, see command SET, **Section 4.12.4.2**).

[dd] — The coefficient of directional spreading; a $\cos^m (\theta)$ distribution is assumed. See the options DEGREES and POWER in the command BOUNDSPEC SHAPE **Section 4.12.8.3**.

HOTSTART — Initial wave field is read from file. This file was generated in a previous SWAN run by means of the HOTFILE command. If the previous run was nonstationary, the time found on the file will be assumed to be the initial run time. The computational grid (both in geographical space and in spectral space) must be identical to the one in which the initial wave field was computed.

`fname' — Name of the file containing the initial wave field.

4.12.9 Physics Commands

4.12.9.1 GEN1

```
GEN1 [cf10] [cf20] [cf30] [cf40] [edmlpm] [cdrag] [umin] [cfpm]
```

With this command the user indicates that SWAN should run in first-generation mode (see **Section 5.1**).

[cf10] — Controls the linear wave growth. Default: [cf10] = 188.

[cf20] — Controls the exponential wave growth. Default [cf20] = 0.59.

[cf30] — Controls the exponential wave growth. Default [cf30] = 0.12.

[cf40] — Controls the dissipation rate, i.e., the time decay scale. Default [cf40] = 250.

[edmlpm] Maximum non-dimensionless energy density of the wind sea part of the spectrum according to Pierson Moskowitz (1964). Default [edmlpm] = 0.0036.

[cdrag] Drag coefficient. Default [cdrag] = 0.0012.

[umin] Minimum wind velocity (relative to current; all wind speeds are taken at 10 m above sea level). Default [umin] = 1.

[cfpm] Coefficient that determines the Pierson Moskowitz frequency.
Default [cfpm] = 0.13.

4.12.9.2 GEN2

```
GEN2 [cf10] [cf20] [cf30] [cf40] [cf50] [cf60] [edmlpm] [cdrag] [umin] [cfpm]
```

GEN2 indicates that SWAN should run in second-generation mode (see **Section 5.1**). The variables are identical to those in the GEN1 command except that [cf50] and [cf60] are added.

[cf50] Controls the spectral energy scale of the limit spectrum.
Default [cf50] = 0.0023.

[cf60] Controls the spectral energy scale of the limit spectrum.
Default [cf60] = -0.223.

4.12.9.3 GEN3

```
        | JANSSEN [cds1] [delta]          |
GEN3  <                                   >          &
        | KOMEN   [cds2] [stpm]           |
                                          |

      (QUADRUPL [iquad] [lambda] [Cnl4] [Csh1] [Csh2] [Csh3] ) (AGROW [a])
```

With this command the user indicates that SWAN should run in third-generation mode for wind input, quadruplet interactions and whitecapping (see **Section 5.1**). Triads, bottom friction and depth-induced breaking are not activated by GEN3. The option GEN3 KOMEN is default. The DIA of the quadruplet interactions is always active with this command but can be deactivated (see command OFF, **Section 4.12.9.9**) and is a poor approximation for long-crested waves and frequency resolutions differing widely from 10% (see command CGRID, **Section 4.12.6.1**).

JANSSEN — Linear growth: Cavaleri and Malanotte-Rizzoli (1981), activated only if the keyword AGROW is present (see below).

Exponential growth: Janssen (1989, 1991).

[cds1] — Coefficient for determining the rate of dissipation (=C_{ds}/s_{PM}^4 in Eq. 16). Default [cds1] = 4.5.

[delta] — Coefficient that determines the dependency of the whitecapping on wave number (mix with Komen et al. (1984) formulation). Default [delta] = 0.5 (see **Section 5.1**).

KOMEN — Linear growth: Cavaleri and Malanotte-Rizzoli (1981), activated only if the keyword AGROW is present (see below).

Exponential growth: Komen et al. (1984).

[cds2] — Coefficient for determining the rate of dissipation (= C_{ds} in Eq. 16). Default [cds2] = 2.36e-5

[stpm] — Value of the wave steepness for a Pierson-Moskowitz spectrum (= s_{PM}^2 with s_{PM} from Eq 16). Default [stpm] = 3.02e-3.

QUADRUPL — With this option the user can influence the computation of nonlinear quadruplet wave interactions. Default: the quadruplets are included in the computations. Quadruplets may be deactivated with command OFF QUAD.

[iquad] — The quadruplets can be integrated by three different numerical procedures:

= 1 Semi-implicit computation of the nonlinear transfer per sweep;
= 2 Fully explicit computation of the nonlinear transfer per sweep;
= 3 Fully explicit computation of the nonlinear transfer per iteration.
Default: [iquad] = 2.

[lambda] — Coefficient for quadruplet configuration (see Eq. 27, **Section 5.1.4.5.1**).

[Cnl4] — Proportionality coefficient for quadruplet interactions (see Eq. 30, **Section 5.1.4.5.1**).

[Csh1] — Coefficient for shallow water scaling (see Eq. 32, **Section 5.1.4.5.1**).

[Csh2] — Coefficient for shallow water scaling (see Eq. 32, **Section 5.1.4.5.1**).

[Csh3] — Coefficient for shallow water scaling (see Eq. 32, **Section 5.1.4.5.1**).

AGROW — If this keyword is used, the wave growth term of Cavaleri and Malanotte (1981) is activated. If this keyword is NOT used, the wave growth term of Cavaleri and Malanotte (1981) is NOT activated. In nonstationary runs SWAN will start with an initial state generated by command INIT (default if not given by the user). In stationary runs SWAN will start with a first guess.

[a] — If the wave growth term of Cavaleri and Malanotte (1981) is activated, [a] is the proportionality coefficient in that term (see **Section 5.1.4**).
Default [a] = 0.0015.

4.12.9.4 BREAKING CONSTANT

```
BREAKING CONSTANT [alpha] [gamma]
```

With this command the user can influence depth-induced wave breaking in shallow water in the SWAN model (see **Section 5.1**).

If this command is not used, SWAN will account for wave breaking anyhow (with default options and values). If the user wants to specifically ignore wave breaking, he should use the command OFF BREAKING.

CONSTANT — Indicates that a constant breaker parameter is to be used.

[alpha] — Proportionality coefficient of the rate of dissipation. Default [alpha] = 1.0.

[gamma] — The ratio of maximum individual wave height over depth. Default [gamma] = 0.73.

4.12.9.5 FRICTION

```
                   | -> JONSWAP [cfjon]
                   |
FRICTION          <    COLLINS   [cfw]
                   |
                   |   MADSEN   [kn]
```

With this optional command the user can activate bottom friction (see **Section 5.1**). If this command is not used, SWAN will not account for bottom friction.

In SWAN the three different formulations available are Hasselmann et al. (1973, JONSWAP), Collins (1972), and Madsen et al. (1988). The default option is JONSWAP.

JONSWAP — Indicates that the semi-empirical expression derived from the JONSWAP results for bottom friction dissipation (Hasselmann et al., 1973, JONSWAP) should be activated. This option is default.

[cfjon] Coefficient of the JONSWAP formulation. [cfjon] is equal to 0.038 m^2s^{-3} for swell conditions and equal to 0.067 m^2s^{-3} for wind sea conditions. Default: [cfjon] = 0.067 m^2s^{-3}

COLLINS Indicates that the expression of Collins (1972) should be activated.

[cfw] Collins bottom friction coefficient, (see **Section 5.1**). Default: [cfw] = 0.015. Note that [cfw] is allowed to vary over the computational region. In that case, use the commands INPGRID FRICTION and READINP FRICTION to define and read the friction data. The command FRICTION is still required to define the type of friction expression. The value of [cfw] in this command is then not required and will be ignored.

MADSEN Indicates that the expression of Madsen et al. (1988) should be activated.

[kn] Equivalent roughness length scale of the bottom. Default: [kn] = 0.05 m. Note that [kn] is allowed to vary over the computational region. In that case, use the commands INPGRID FRICTION and READINP FRICTION to define and read the friction data. This command FRICTION is still required to define the type of friction expression. The value of [kn] in this command is then not required (it will be ignored).

4.12.9.6 TRIAD

```
TRIAD [trfac] [cutfr]
```

TRAID allows the user to activate the triad wave-wave interactions in the SWAN model (see **Section 5.1**).

[trfac] The value proportionality coefficient. Default: [trfac] = 0.1.

[cutfr] Controls the maximum frequency that is considered in the triad computations. The value of [cutfr] is the ratio of this maximum frequency over the mean frequency. Default: [cutfr] = 2.2.

4.12.9.7 OBSTACLE

```
                | TRANSM [trcoef]             |
OBSTACLE  <                                   > (REFL [reflc]) LINE < [xp] [yp]
                | DAM [hgt] [alpha] [beta]    |
```

With this optional command the user provides the characteristics of a (line of) sub-grid obstacle(s) through which waves are transmitted or against which waves are reflected (possibly both at the same time). Reflection is only spectral reflection. The obstacle is sub-grid in the sense that it is narrow compared to the spatial meshes. Its length should be at least one mesh length.

The location of the obstacle is defined by a sequence of corner points of a line. The obstacles interrupt the propagation of the waves from one grid point to the next wherever the obstacle line is located between two neighboring grid points. The resolution of the obstacle is therefore equal to the computational grid spacing. In order for an obstacle to be effective it must be located such that it crosses at least one grid line. This is always the case when an obstacle is larger than one mesh length.

The computation of transmission and reflection is problematic if an obstacle runs exactly through one or more grid points. SWAN will move the obstacle over a small distance (0.01 of the mesh size) if this occurs. The reflection results will be incorrect if more than one obstacle crosses the same grid line between two neighboring grid points. SWAN is not able to detect this so the user must check that the model fulfills this condition. The command is NOT used for 1-D mode.

The obstacle locations may be plotted with option LINES in the PLOTGEOGR command (providing isoline plots or vector plots).

TRANSM With this option the user indicates that the transmission coefficient is a constant.

[trcoef] Constant transmission coefficient formulated in terms of wave height, i.e. ratio of transmitted significant wave height over incoming significant wave height. Default: [trcoef] = 0.0 (no transmission = complete blockage)

DAM With this option the user indicates that the transmission coefficient depends on the incident wave conditions at the obstacle and on the obstacle height, which may be submerged (see **Section 5.1**).

[hgt] The elevation of the top of the obstacle above reference level, which is the same reference level as for bottom. Use a negative value if the top is below that reference level. This value is required if OBSTACLE is used.

[alpha], [beta]
Coefficients determining the transmission coefficient (see **Section 5.1**). Default: [alpha] = 2.6 and [beta] = 0.15.

REFL
If this keyword is present the obstacle will reflect wave energy, possibly in combination with transmission. Reflections will be computed only if the spectral directions cover the full 360°, i.e., if in the command CGRID the option CIRCLE is activated.

[reflc]
Constant reflection coefficient, formulated in terms of wave height, i.e. the ratio of reflected significant wave height over incoming significant wave height. [reflc] must be >= 0 and <= 1. Default: [reflc] = 1, if the keyword REFL is present.

LINE
With this required keyword the user defines the location of the obstacle(s).

[xp], [yp]
Coordinates of a corner point of the line that defines the location of the obstacle(s) (if Cartesian coordinates are used in m, if spherical coordinates are used in degrees, see command CGRID, **Section 4.12.4.4**). At least two corner points must be provided.

Note: On plotting near obstacles: see **Section 4.13.2** on plot output.

4.12.9.8 SETUP

SETUP [supcor]

If this optional command is given, the wave-induced set-up is computed and accounted for in the wave computations, and during the computation it is added to the depth that is obtained from the READ BOTTOM and READ WLEVEL commands.

[supcor]
By default the wave-induced setup is computed with an added constant such that the setup is zero in the deepest point of the model grid. The user can modify the constant by the value of [supcor]. The user can thus impose a setup in any one point (and only one) in the grid by first running SWAN, then reading the setup in that point and adding or subtracting the required value of [supcor] (in m; positive if the setup has to rise).
Default: [supcor] = 0.

4.12.9.9 OFF

```
                 | WINDGROWTH   |
                 |              |
                 | QUADRUPL     |
                 |              |
                 | WCAPPING     |
OFF             <                >
                 | BREAKING     |
                 |              |
                 | REFRAC       |
                 |              |
                 | FSHIFT       |
                 |              |
                 | BNDCHK       |
```

With this optional command the user can change the default inclusion of various physical processes. This command is recommended for research purposes and not for operational use.

WINDGROWTH Switches off wind growth (in commands GEN1, GEN2 and GEN3).

QUADRUPL Switches off quadruplet wave-wave interactions (in command GEN3).

WCAPPING Switches off whitecapping (in command GEN3).

BREAKING Switches off depth-induced breaking dissipation. Caution: wave heights may diverge in very shallow water.

REFRAC Switches off refraction (action transport in θ-direction).

FSHIFT Switches off frequency shifting in frequency space (action transport in σ-space).

BNDCHK Switches off the checking of the difference between imposed and computed significant wave height at the boundary of the computational grid (see also command SET, **Section 4.12.4.2**).

4.12.10 Numerics

4.12.10.1 PROP

```
          |BSBT                                  |
PROP     <                       |SEC    |       |
          |GSE [waveage]        < MIN     >      |
          |                      | HR    |       |
          |                      | DAY   |       |
```

The PROP command is used for choosing (a) the BSBT scheme, used for stationary and nonstationary computations, (b) the default S&L scheme (nonstationary) or the default SORDUP scheme (stationary cases), or (c) the wave age, for the default nonstationary S&L scheme. With computational nonstationary S&L, if the run time limitations require CFL >>1 for the dominant wave period, BSBT is recommended rather than the default S&L.

BSBT — The BSBT scheme will be used in the computations.

GSE — Garden-sprinkler-effect is to be counteracted in the S and L propagation scheme (default for nonstationary computations) by adding a diffusion term to the basic equation. This may affect the numerical stability of SWAN (see **Section 5.2**).

[waveage] — The time interval used to determine the diffusion, which counteracts the so-called garden-sprinkler effect. The default value of [waveage] equals zero, or no added diffusion. The value of [waveage] should correspond to the travel time of the waves over the computational region.

Note: All schemes can be used in combination with curvilinear grids. With the higher order schemes (S&L and SORDUP) it is important to use a gradually varying grid, otherwise there may be a severe loss of accuracy. If sharp transitions in the grid cannot be avoided it is safer to use the BSBT scheme.

4.12.10.2 NUMERIC

```
                                                      |-> STAT [mxitst]       |
NUMERIC  ( ACCUR [drel] [dhoval] [dtoval] [npnts]    <                         > [limiter] )   &
                                                      |NONSTAT [mxitns]       |
         ( DIRIMPL [cdd] [cdlim]                                )                               &

         ( SIGIMPL [css] [prec] [eps1] [eps2] [outp] [niter]    )                               &

         ( SETUP [prec] [eps2] [outp] [niter]                   )
```

This optional command allows the user to influence some numerical properties of SWAN.

ACCUR — Influences the criterion for terminating the iterative procedure in the SWAN computations (both stationary and nonstationary mode). SWAN terminates the iterations if:

a) The change in the local significant wave height (H_s) from one iteration to the next is less than
   (1) fraction [drel] of that height or
   (2) fraction [dhoval] of the average significant wave height (average over all wet grid points).

And

b) The change in the local mean wave period ($RPER = RT_{m01}$) from one iteration to the next is less than
   (1) fraction [drel] of that period or
   (2) fraction [dtoval] of the average mean wave period (average over all wet grid points).

And

c) Conditions (a) and (b) are fulfilled in more than fraction [npnts]% of all wet grid points.

[drel] — Default [drel] = 0.02 [-].

[dhoval] — Default [dhoval] = 0.02 [-].

[dtoval] — Default [dtoval] = 0.02 [-].

[npnts] — Default [npnts] = 98. [-].

STAT — The maximum number of iterations in a stationary computation.

[mxitst] — The maximum number of iterations for stationary computations. The computation stops when this number is exceeded. Default [mxitst] = 15. Note that [mxitst] can be set to 0 if one wants to check the input to the model without making computations.

NONST — The maximum number of iterations per time step in a nonstationary computation.

[mxitns] — The maximum number of iterations per time step for nonstationary computations. The computation moves to the next time step when this number is exceeded. Default: [mxitns] = 1. Note that [mxitns] can be set to zero if one wants to check the input to the model without making computations.

[limiter] Determines the maximum change per iteration (stationary and nonstationary runs; note that default one iteration equals one upgrade is used per time step) of energy density per spectral (α,θ)-bin in terms of a fraction of the omni-directional Phillips level (**Section 5.1**). Default: [limiter] = 0.1. If command OFF QUAD is used, then SWAN deactivates the limiter.

DIRIMPL This option is used to influence the numerical scheme for refraction.

[cdd] A value of [cdd] = 0 corresponds to a central scheme and has the largest accuracy (diffusion ˜ 0) but the computation may more easily generate spurious fluctuations. A value of [cdd] = 1 corresponds to an upwind scheme and it is more diffusive and therefore preferable if (strong) gradients in depth or current are present. Default: [cdd] = 0.5.

[cdlim] If the spatial discretization of the bathymetry or the currents is too coarse, the waves may turn too far (more than 90 degrees, say) over one spatial grid step. The computational results will then be very inaccurate. In such a case SWAN can limit the maximum turning of the waves over one spatial grid to 90 degrees to obtain robust (but not necessarily correct) results.
< 0 Then no limiter is used (this is default).
= 0 Refraction is off (same effect as command OFF).
= 4 Waves turning limited to about 90 degrees over one spatial grid step.

SIGIMPL Controls the accuracy of computing the frequency shifting. The type of preconditioning, required accuracy to terminate the iteration ([eps1] and [eps2]), the amount of required output and maximum number of iterations of the linear iterative ILU-CGSTAB solver (used in the computations in the presence of currents or time varying depth) can be chosen by the user.

SETUP Controls the accuracy of computing the wave-induced setup. Type of preconditioning, required accuracy to terminate the iteration ([eps2]), the amount of required output and maximum number of iterations of the linear iterative ILU-CGSTAB solver can be chosen by the user.

[css] A value of [css] = 0 corresponds to a central scheme and has the largest accuracy (diffusion ˜ 0) but the computation may more easily generate spurious fluctuations. A value of [css] = 1 corresponds to an upwind scheme and it is more diffusive and therefore preferable if (strong) gradients in the currents are present. Default: [css] = 0.5.

[prec] Value for preconditioner of the ILU-CGSTAB solver (see **Section 5.2**)
= 0 No preconditioner.
= -1 Diagonal preconditioner.
= -3 ILU preconditioner.
= 3 Default [prec] (most robust option regarding the preconditioner).

[eps1] [eps2] Criterion to terminate the iterative solver. The criterion is based on:

$$\|AN_k - b\|_2 \le [eps1] + [eps2]\|AN_{k-1} - b\|_2 + [eps2]\|b\|_2$$

Where A is a matrix, N is the action density, b is the right hand vector, k is the iteration number. (in case of the SETUP option, [eps1] = 0.
Defaults: [eps1] = 1.e-6 and [eps2] = 1.e-4

[outp] Output for ILU-CGSTAB solver.
= 0 No output.
= 1 Additional information about the iteration process is written to the print file.
= 2 Gives a maximal amount of output concerning the iteration process.
Default = 0.

[niter] Maximum number of iterations for the ILU-CGSTAB solver (per geographic grid point). Default [niter] = 20.

4.13 Output Commands

There are two categories of output commands, each with two classes:

1. Locations
   1a. Commands defining sets of output locations at which the user requires output. Each set is indicated with a name ('sname' in this manual) which must be unique and not more than eight characters long.

   Types of sets of output points:
   FRAME To define a set of output locations on a regular grid.
   GROUP To define a set of output locations on a regular or curvilinear grid.
   CURVE To define a set of output locations along a curve.
   RAY To define a set of output locations along a depth or bottom contour line (with ISOLINE).
   ISOLINE To define a set of output locations along a depth or bottom contour line (with RAY).
   POINTS To define a set of isolated output locations.
   NGRID To define a set of output locations for a nested grid to be used in a subsequent SWAN run.

   Commands FRAME, GROUP, RAY, ISOLINE and NGRID cannot be used in 1-D mode. If one gives one name for two sets of output locations, the first set is lost (first in the sequence in the command file). SWAN reserves three special types (BOTTGRID, CGRID and TESTPNTS) for use (see below), but the user may not define sets with these names.

   1b. Commands defining additional geographical information to be used in plot output:

LINE Defines a line to indicate coastlines or contours of a certain object. LINE can be used in a plot generated with command PLOTGEOGR.

SITES Plot (geographic) names in a plot generated with command PLOTGEOGR.

Commands LINE and SITES cannot be used in 1-D mode.

2. Write / plot

2a. Commands defining data file output (write) at the above defined set(s) of output locations:

BLOCK Write spatial distributions (only for FRAME and GROUP).

TABLE Write output for (set of) output location(s).

SPECOUT Write to data file the variance / energy (see command SET, **Section 4.12.4.2**) density spectrum for (set of) output location(s).

NESTOUT Write to data file two-dimensional action density spectra (relative frequency) along the boundary of a nested grid (see command NGRID, **Section 4.13.1.7**) to be used in a subsequent SWAN run.

Commands BLOCK and NESTOUT cannot be used in 1-D mode.

2b. Commands defining plot output at the above defined sets of output locations (and auxiliary plot output):

PLOTGEOGR A plot of a spatial distribution is required (contour line plot and / or vector plot).

PLOTSTAR Show the directional distribution of the action transport.

PLOTPP Show the locations of points where the problems occur (problem points).

PLOTSPEC Polar plot of a 2-D variance/energy density spectrum, for (a set) of output location(s).

Commands PLOTGEOGR, PLOTSTAR and PLOTPP cannot be used in 1-D mode.

Note: The plot output of the model is not adapted to obstacles. For the plot, computational results may be interpolated from points at both sides of the obstacle. Results for output points near an obstacle must therefore be interpreted with caution. The effect is, for instance, that the isolines of the wave height near the obstacle appear rather "ragged".

4.13.1 Output Location Commands

4.13.1.1 FRAME

```
FRAME 'sname' [xpfr] [ypfr] [alpfr] [xlenfr] [ylenfr] [mxfr] [myfr]) ([scale])
```

With this optional command the user defines output on a rectangular grid in a regular, rectangular frame. FRAME cannot be used in 1-D mode.

If the set of output locations is identical to a part of the computational grid, then the user can use the alternative command GROUP.

'sname' Name of the frame defined by this command.

[xpfr] X-coordinate of the origin of the frame in problem coordinates. Cartesian coordinate units are in m and spherical coordinate units are in degrees (see command COORD, **Section 4.12.4.4**).

[ypfr] Y-coordinate of the origin of the frame in problem coordinates. Cartesian coordinate units are in m and spherical coordinate units are in degrees (see command COORD, **Section 4.12.4.4**).

[alpfr] Direction of the x-axis of the frame, in degrees for Cartesian convention, and [alpfr] must be zero in case of spherical coordinates).

[xlenfr] Length of the frame in x-direction. Cartesian coordinate units are in m and spherical coordinate units are in degrees (see command COORD, **Section 4.12.4.4**).

[ylenfr] Length of the frame in y-direction. Cartesian coordinate units are in m and spherical coordinate units are in degrees (see command COORD, **Section 4.12.4.4**).

[mxfr] Number of meshes in x-direction of the rectangular grid in the frame (one less than the number of grid points in this direction). Default: [mxf] = 20.

[myfr] Number of meshes in y-direction of the rectangular grid in the frame (one less than the number of grid points in this direction). Default: [myf] = 20.

[scale] Controls the horizontal scale of plotting (if plotting is required): length in cm on paper per m or degree in reality. Default: Plot as large as possible within size of 18 x 24 cm^2 (x- and y-direction respectively).

Some output may be required on a frame that is identical with the input (bottom/current) grid or with the computational grid (e.g., for test purposes or to avoid interpolation errors in the output). These frames need not be defined when using the FRAME command. The frames are always generated automatically by SWAN under the names 'sname' = 'BOTTGRID' (for the bottom/current grid) and 'sname' = 'CGRID' (for the computational grid).

4.13.1.2 GROUP

```
GROUP 'sname' SUBGRID [ix1] [ix2] [iy1] [iy2]
```

With this optional command the user defines a group of output locations on a rectangular or curvilinear grid that is identical to (part of) the computational grid (rectilinear or curvilinear). GROUP may be convenient for use in obtaining output that is not affected by interpolation errors, which would occur when an output grid is used that is not identical with (part of) the computational grid.

GROUP cannot be used in 1-D mode.

Note: SWAN cannot generate geographical plots for GROUP on a curvilinear computational grid (see command PLOTGEOGR, **Section 4.13.2.6**).

Command CGRID should precede command GROUP. The subgrid contains those points (ix, iy) of the computational grid for which:

[ix1] <= ix <= [ix2] and [iy1] <= iy <= [iy2] .

For convenience the size of the group, the corner coordinates and the angle with the problem coordinate system are written to print file. The origin of the computational grid is (ix = 0, iy = 0).

'sname' Name of the set of output locations defined by this command.

[ix1] Lowest grid index of subgrid in terms of computational grid in ix-direction.

[iy1] Lowest grid index of subgrid in terms of computational grid in iy-direction.

[ix2] Highest grid index of subgrid in terms of computational grid in ix-direction.

[iy2] Highest grid index of subgrid in terms of computational grid in iy-direction.

Limitations: [ix1] >= 0, [ix2] <= [mxc], [iy1] >= 0, [iy2] <= [myc] ([mxc] and [myc] as defined in the command CGRID).

4.13.1.3 CURVE

CURVE 'sname' [xp1] [yp1] < [int] [xp] [yp] >

With the optional command CURVE, the user defines output along a curved line. The curve is actually a broken line, defined by the user with its corner points. The values of the output quantities along the curve are interpolated from the computational grid. This command may be used more than once to define more curves.

'sname' Name of the curve.

[xp1], [yp1] Problem coordinates of the first point of the curve. If Cartesian coordinates are used, the units are in m. If spherical coordinates are used, the units are in degrees (see command COORD, **Section 4.12.4.4**).

[int] SWAN will generate output at [int]-1 equidistant locations between two subsequent corner points of the curve (including the two corner points of the curve).

[xp], [yp] Problem coordinates of a corner point of the curve. Repeat the group [int] [xp] [yp] in proper order if there are more corner points on the curve.

4.13.1.4 RAY

```
RAY      'rname' [xp1]  [yp1]  [xq1]  [yq1]            &
         < [int]  [xp]  [yp]  [xq]  [yq] >
```

RAY is an optional command with which the user provides SWAN with information to determine output locations along the depth contour line(s) defined subsequently in command ISOLINE (see below). The model determines the locations as the intersections of the depth contour line(s) and the set of straight rays defined by command RAY. These rays are characterized by a set of master rays defined by their start and end positions ([xp], [yp]) and ([xq], [yq]). Between each pair of sequential master rays SWAN generates [int]-1 intermediate rays by linear interpolation of the start and end positions.

Note: The rays defined here are not synonymous with wave rays (e.g. as obtained from conventional refraction computations).

'rname' Name of the set of rays defined by this command.

[xp1], [yp1]
[xq1], [yq1] Problem coordinates of the begin and end points of the first master ray.

[int] Number of subdivisions between the previous master ray and the following master ray defined by the following data (number of subdivisions is one more than the number of interpolated rays).

[xp], [yp];
[xq], [yq] Problem coordinates of the begin and end points of each subsequent master ray. For Cartesian coordinates the units are m and for spherical coordinates the units are degrees (see command COORD, **Section 4.12.4.4**).

RAY cannot be used in 1-D mode.

4.13.1.5 ISOLINE

```
                                      | -> DEPTH          |
ISOLINE  'sname'   'rname'           <                     >  [dep]
                                      |   BOTTOM          |
```

With this optional command the user defines a set of output locations along one depth or bottom level contour line (in combination with command RAY).

'sname' Name of the set of output locations defined by this command.

'rname' Name of the set of rays (as defined in command RAY).

[dep] The depth (in m) of the depth contour line along which SWAN generates output locations. If the keyword DEPTH appears in front of the value the true depth is used, if the keyword BOTTOM appears the water level is ignored, i.e. the depth with respect to datum level is used.

The set of output locations along the depth contour lines created with this command is of the type CURVE.

ISOLINE cannot be used in 1-D mode.

4.13.1.6 POINTS

```
                         | < [xp] [yp] >    |
POINTS  'sname'         <                    >
                         | FILE 'fname'     |
```

With this optional command the user defines a set of individual output locations (points). The coordinates of these points are given in the command itself or read from a file (option FILE).

'sname' Name of the points.

[xp], [yp] Problem coordinates of one output location.
If Cartesian coordinates are used, the units are in m.
If spherical coordinates are used, the units are degrees (see command COORD, **Section 4.12.4.4**).

4.13.1.7 NGRID

```
NGRID 'sname' [xpn] [ypn] [alpn] [xlenn] [ylenn] [mxn] [myn]
```

NGRID cannot be used in 1-D mode.

If the user wishes to carry out nested SWAN run(s) a separate, coarse-grid SWAN run is required. With this optional command NGRID, the user defines in the present coarse-grid run a set of output locations along the boundary of the subsequent nested computational grid. The set of output locations thus defined is of the type NGRID.

Command NESTOUT is required after NGRID to generate some data for the (subsequent) nested run. Commands BLOCK and PLOT... are not used because a set of locations of the type NGRID does not represent a geographic region, only its outline.

'sname' Name of the set of output locations along the boundaries of the following nested computational grid defined by this command.

[xpn] Geographic location of the model grid origin of the coarse-grid run in the problem coordinate system (x-coordinate). If Cartesian coordinates are used, the units are m. If spherical coordinates are used, the units are degrees (see command COORD, **Section 4.12.4.4**).

[ypn] Geographic location of the model grid origin of the coarse-grid run in the problem coordinate system (y-coordinate). If Cartesian coordinates are used, the units are m. If spherical coordinates are used, the units are degrees (see command COORD, **Section 4.12.4.4**).

[alpn] Direction of the positive x-axis of the computational grid of this coarse-grid run (in degrees, Cartesian convention).

[xlenn] Length in the x-direction of the nested grid If Cartesian coordinates are used, the units are m. If spherical coordinates are used, the units are degrees (see command COORD, **Section 4.12.4.4**).

[ylenn] Length in y-direction of the nested grid. If Cartesian coordinates are used, the units are m. If spherical coordinates are used, the units are degrees (see command COORD, **Section 4.12.4.4**).

[mxn] Number of meshes of the output grid in the x-direction of this grid (this number is one less than the number of grid points in this direction). [mxn] does not have to be equal to the number of meshes in the nested computation, as SWAN will interpolate the required information. Default: [mxn] is chosen such that the mesh size of the output grid is roughly equal to the mesh size of the coarse grid, but at least one.

[myn] Number of meshes of the output grid in the y-direction of this grid (this number is one less than the number of grid points in this direction). [myn] does not have to be equal to the number of meshes in the nested computation, as SWAN will interpolate the required information. Default: [myn] is chosen such that the mesh size of the output grid is (roughly) equal to the mesh size of the coarse grid, but at least one.

If the user wishes to view the location of the nested grid, e.g. in relation to the bathymetry in the computational region of the coarse run (e.g. defined by 'sname' = 'CGRID'), the command PLOTGEOGR should be used. If the name of the nested grid is 'sname' = 'NG', the syntax should be:

PLOTGEOGR 'CGRID' ISO BOTTOM LOCATION 'NG' (see command PLOTGEOGR, **Section 4.13.2.6**).

4.13.1.8 LINE

```
           | -> CONTINUOUS          |
LINE      <                          >  (COLOR [ipen]) < [xp] [yp] >
           |    DASHED [pat] [len] |  |
```

With this optional command the user defines a line that is plotted with option LINES (see the PLOT... commands in **Section 4.13.2**). The line is defined by the corner points of a sequence of straight lines. LINE is provided to facilitate orientation on the plots. One can indicate coastlines, contours of certain landmarks, etc. The command LINE may be repeated.

CONT Indicates the type of line will be a continuous line.

DASH A dashed line will be drawn.

[pat] Pattern type number (integer). Default: [pat] = 3.

[len] Pattern length in cm (real). Default: [len] = 1.

COLOR To be used if a color other than black is desired.

[ipen] Color number (integer). Default: [ipen] = 2.

[xp], [yp] Coordinates of a corner point of the line in the problem coordinate system. If Cartesian coordinates are used, the units are in m. If spherical coordinates are used, the units are degrees (see command COORD, **Section 4.12.4.4**).

LINE cannot be used in 1-D mode.

4.13.1.9 SITES

```
                                          | REGION   |
SITES      < 'pname' [xp] [yp] [size]    <           > >
                                          | TOWN     |
```

SITES is an optional command in which the user defines names of towns, regions etc., that can be plotted (with option SITES in command PLOTGEOGR). The command SITES may be repeated.

'pname' The name of a town, region etc., with a maximum 16 characters long.

[xp],[yp] X- and y-coordinates of the location where the name must be plotted in the problem coordinate system. If Cartesian coordinates are used, the units are in m. If spherical coordinates are used, the units are degrees (see command COORD, **Section 4.12.4.4**).

[size] The size of the characters in the plot (in cm). Default: [size] = 0.28 cm.

REGION The name is plotted with the point of reference in the middle of the name (if space allows). This is intended for names of regions, islands, etc.

TOWN The first character is placed at the point of reference (if space allows). This is intended for names of towns, etc.

SITES cannot be used in 1-D mode.

4.13.2 Write or Plot Data Commands

For definitions of output parameters see **Section 4.14**.

WARNING:
When integral parameters are computed by the user from the output spectrum of SWAN, differences with the SWAN-computed parameters may occur. The reasons are:

(a) SWAN accepts at the boundaries of the computational grid only the user-imposed incoming wave components, and it replaces the user-imposed outgoing wave components with computed components (propagating to the boundary from the interior region).

(b) During the computation of the parameters, SWAN adds an analytical (diagnostic) high-frequency tail to the discrete spectrum.

(c) SWAN has an option to only compute within a pre-set directional sector (pre-set by the user). Wave components outside this sector are totally ignored by SWAN (no additions or replacements).

This is particularly relevant along the boundaries of SWAN where the user-imposed integral parameters (boundary conditions) may differ from the SWAN-computed parameters. The user is informed by means of a WARNING in the output print file, (see **Section 4.11.2.1**) when the computed significant wave height differs by more than 10%, say, from the user-imposed significant wave height (command BOUNDPAR2). The actual value of this difference can be set by the user (see command SET, **Section 4.12.4.2**).

It should be noted that the plot output of the model is not adapted to obstacles. For the plot, computational results may be interpolated from points at both sides of the obstacle. Results for output points near an obstacle must therefore be interpreted with caution. The effect is that the isolines of the wave height near the obstacle appear rather "ragged".

4.13.2.1 QUANTITY

```
                 |..........|
QUANTITY        <            >   `short' `long' [lexp] [hexp] [excv]
                 |..........|

      [power]                        (For output quantities PER, RPER and WLEN) &

      [ref]                                 (For output quantity TSEC)         &

      [fswell]                              (For output quantity HSWELL)       &

       |-> PROBLEMCOORD       |
      <                        >            {For directions (DIR, TDIR, PDIR)
       |  FRAME               |        and vectors (FORCE, WIND, VELOCITY,
                                                   TRANSPORT)}
```

With this command the user can influence,

(a) the naming of output quantities,
(b) the accuracy of writing output quantities,
(c) the definition of some output quantities, and
(d) reference direction for vectors.

```
 |...|
<     >
 |...|
```

The output parameters are the same as in command BLOCK.

`short'

User preferred short name of the output quantity (e.g. the name appearing in the heading of a table written by SWAN). If this option is not used, SWAN will use a realistic name.

`long' Long name of the output quantity (e.g. the name appearing in the heading of a block output written by SWAN). If this option is not used, SWAN will use a realistic name.

[lexp] Lowest expected value of the output quantity.

[hexp] Highest expected value of the output quantity. The highest expected value is used by SWAN to determine the number of decimals in a table with heading. The QUANTITY command can be used in case the default number of decimals in a table is unsatisfactory.

[excv] In case there is no valid value (e.g., wave height in a dry point), this exception value of the output quantity is written in a table or block output.

The following data are accepted only in combination with selected output quantities:

[power] Power p appearing in the definition of PER, RPER and WLEN (see **Section 4.14**). Note that the value for [power] given for PER affects also the value of RPER. The power for WLEN is independent of that of PER or RPER. Default: [power] = 1.

[ref] Reference time used for the quantity TSEC. Default: Starting time of the first computation.

[fswell] Upper limit of frequency range used for computing the quantity HSWELL (see **Section 4.14**). Default: [fswell] = 0.1 Hz.

PROBLEMCOORD Vector components are relative to the x- and y-axes of the problem coordinate system (see command COORD, **Section 4.12.4.4**). If Cartesian direction convention is used directions are counter-clockwise relative to the positive x-axis of the problem coordinate system (see command SET, **Section 4.12.4.2**). If Nautical direction convention is used the directions are relative to north (clockwise; see command SET, **Section 4.12.4.2**).

FRAME If output is requested on sets created by command FRAME or automatically (CGRID (see **Section 4.12.6.1**) or BOTTGRID (see **Section 4.12.7.1**)), vector components are relative to the x- and y-axes of the frame coordinate system (see command FRAME, **Section 4.13.1.1**). If Cartesian direction convention is used directions are counter-clockwise relative to the positive x-axis of the problem coordinate system (see command SET, **Section 4.12.4.2**). If Nautical direction convention is used the directions are relative to north (clockwise; see command SET, **Section 4.12.4.2**).

Examples:

```
        QUANTITY Xp hexp = 100                 $ for simulations of lab experiments.
```

```
QUANTITY Hs excv = -9.                      $ to change the exception value for Hs.
QUANTITY Hswell fswell = -0.08              $ to change the value of [fswell].
QUANTITY Per short = 'Tm-1,0' power = 0.    $ to redefine average wave period.
QUANTITY Transp Frame                       $ to obtain vector components and
                                            direction with respect to frame.
```

4.13.2.2 BLOCK

```
                     | -> HEADER      |                                              |
BLOCK  'sname'   <                    >  'fname' (LAY-OUT [idla])
                     |   NOHEADER     |
                | HSIGN      |
                |            |
                | HSWELL     |
                |            |
                | DIR        |
                |            |
                | PDIR       |
                |            |
                | TDIR       |
                |            |
                | TM01       |
                |            |
                | RTM01      |
                |            |
                | RTP        |
                |            |
                | PER        |
                |            |
                | RPER       |
                |            |
                | TM02       |
                |            |
                | FSPR       |
                |            |
                | DSPR       |
                |            |
                | DEPTH      |
                |            |
                | VEL        |
                |            |
                | FRCOEF     |
                |            |
                | WIND       |                                        | -> SEC |
               <             > [unit] >    (OUTPUT [tbegblk] [deltblk])    <     MIN  >
                | DISSIP     |                                        |     HR   |
                |            |                                        |     DAY  |
                | QB         |
                |            |
                | TRANSP     |
|              |
| FORCE        |
|              |
| UBOT         |
|              |
| URMS         |
|              |
| WLENGTH      |
|              |
| STEEPNESS    |
|              |
| DHSIGN       |
|              |
| DRTM01       |
|              |
| LEAK         |
|              |
| XP           |
|              |
| YP           |
|              |
| DIST         |
```

BLOCK cannot be used in 1-D mode. BLOCK indicates that one or more spatial distributions should be written to a file.

'sname' Name of FRAME or GROUP (see command FRAME or GROUP, **Sections 4.13.1.1** or **4.13.1.2**).

HEADER Indicates that the output should be written to a file with header lines. The text of the header indicates run identification (see command PROJECT, **Section 4.12.4.1**), time, frame name or group name ('sname'), variable and unit. The number of header lines is eight.

NOHEADER With this option the user indicates that the output should be written to a file without header lines.

'fname' Name of the data file where the output is to be written to.
Default for option HEADER is the print file (see **Section 4.11.2.1**). In case of NOHEADER the filename is required.

LAY-OUT With this option the user can prescribe the layout of the output to file with the value of [idla].

[idla] See command READINP in **Section 4.12.7.2** (options are [idla] = 1, 3, 4).

Option 4 is recommended for postprocessing by MATLAB. Default [idla] = 1.

For definitions of the output quantities (Hsign etc.) see **Section 4.14**.

Note: For output on (or near) a boundary of the computational grid or near an obstacle, see the warning at the introduction of this section (**4.13.2**).

The wave parameters in the SWAN output are computed from the wave spectrum over the prognostic portion of the spectrum with the diagnostic tail added. This value may therefore deviate slightly from values computed by the user from the output spectrum of SWAN not containing the diagnostic tail.

TIME — Time for which the values on the same line are valid. Useful only in case of nonstationary computations.

HSIGN — Significant wave height (in m).

HSWELL — Swell wave height (in m).

DIR — Mean wave direction for Cartesian convention relative to the x-axis of the problem coordinate system (counter-clockwise direction; Cartesian or Nautical convention, see command SET, **Section 4.12.4.2**). A possible exception occurs in the case of output with BLOCK command in combination with command FRAME (see command QUANTITY, **Section 4.13.2.1**).

PDIR — Peak wave direction in degrees for Cartesian convention relative to the x-axis of the problem coordinate system (counter-clockwise). A possible exception occurs in the case of output with BLOCK command in combination with command FRAME (see command QUANTITY, **Section 4.13.2.1**).

TDIR — Direction of energy transport in degrees for Cartesian convention relative to the x-axis of the problem coordinate system (counter-clockwise). A possible exception occurs in the case of output with BLOCK command in combination with command FRAME, (see command QUANTITY, **Section 4.13.2.1**).

TM01 — Mean absolute wave period (in s).

RTM01 — Mean relative wave period (in s).

RTP — Peak period (in s) of the variance density spectrum (relative frequency spectrum).

PER Mean absolute wave period (in s).

RPER Mean relative wave period (in s).

TM02 Mean absolute zero crossing period (in s).

FSPR The normalized width of the frequency spectrum.

DSPR Directional spreading of the waves (in degrees).

DEPTH Water depth (in m) (not the bottom level!).

VEL Current velocity (in m/s).

FRCOEF Friction coefficient (equal to [cfw] or [kn] in command FRICTION).

WIND Wind velocity (in m/s).

DISSIP Energy dissipation due to bottom friction, wave breaking and whitecapping (in W/m^2 of m^2/s, depending on command SET).

QB Fraction of breaking waves due to depth-induced breaking.

TRANSP Transport of energy (vector; W/m or m^3/s, depending on command SET).

FORCE Wave induced force per unit surface area (vector; in N/m^2).

UBOT The rms value of the maxima of the orbital velocity near the bottom (in m/s). Output only if command FRICTION is used. If one wants to output UBOT with friction neglected in the computations, then he/she should use the command FRICTION with the value of the friction set at zero, e.g., FRICTION COLLINS 0.

URMS The rms value of the orbital velocity near the bottom (in m/s). If one wants to output UBOT but friction is to be neglected in the computations, then one should use the command FRICTION with the value of the friction set at zero, e.g., FRICTION COLLINS 0.

WLEN Average wavelength (in m).

STEEPNESS Average wave steepness (dimensionless).

DHSIGN The difference in significant wave height as computed in the last two iterations. This is not the difference between the computed values and the final limit of the iteration process. It is at most an indication of this difference.

DRTM01 The difference in average wave period (RTMO1) as computed in the last two iterations. This is not the difference between the computed values and the final limit of the iteration process. It is at most an indication of this difference.

LEAK Numerical loss of energy equal to across boundaries = [dir1] and = [dir2] of a directional sector (see command CGRID, **Section 4.12.6.1**).

XP User instructs SWAN to write the x-coordinate in the problem coordinate system of the output location.

YP User instructs SWAN to write the y-coordinate in the problem coordinate system of the output location.

DIST If output has been requested along a curve (see command CURVE, **Section 4.13.1.3**) then the distance along the curve can be obtained with this command TABLE.
Definition: DIST is the distance along the curve measured from the first point on the curve to the output location on the curve in meters.

[unit] This controls the scaling of output. The program divides computed values by [unit] before writing to file, so the user should multiply the written value by [unit] to obtain the proper value.
Default if HEADER is selected, value is written as a five-position integer. SWAN takes [unit] such that the largest number occurring in the block can be printed.
Default if NOHEADER is selected, values are printed in floating-point format, [unit] = 1.

OUTPUT The user requests output at various times. If the user does not use this option, the program will give BLOCK output for the last time step of the computation.

[tbegblk] Begin time of output, the format is:

= 1	ISO notation	19870530.153000
= 2	(as in HP compiler):	`30-May-87 15:30:00'
= 3	(as in Lahey compiler):	05/30/87.15:30:00
= 4		15:30:00
= 5		87/05/30 15:30:00'
= 6	as in WAM	8705301530

This format is installation dependent. See installation manual on the SWAN web site. Default is ISO.

[deltblk] Time interval of output, the unit is indicated in the next option.

SEC Unit seconds.

MIN Unit minutes.

HR Unit hours.

DAY Unit days.

4.13.2.3 TABLE

```
                        | ->HEADER        |
                        |                 |
TABLE 'sname'     <      NOHEADER         > 'fname'                          &
                        |                 |
                        |  INDEXED        |
               | ...    |                                  | -> SEC     |
          <  <           >   >   (OUTPUT [tbegtbl] [delttbl]   <     MIN        >)
               | ...    |                                  |    HR      |
                                                           |    DAY     |
```

TABLE is an optional command indicating that for each output location set to 'sname' (see commands POINTS, CURVE, FRAME, or GROUP, **Section 4.13.1.6**, **4.13.1.3**, **4.13.1.1** or **4.13.1.2**), one or more variables should be written to a file. The keywords HEADER and NOHEADER determine the appearance of the table. The filename determines the destination of the data.

'sname' Name of the set of POINTS, CURVE, FRAME or GROUP (see **Section 4.13.1.6**, **4.13.1.3**, **4.13.1.1** or **4.13.1.2**).

HEADER Output is written in fixed format to file with headers giving name of variable and unit per column. A disadvantage of this option is that the data are written in fixed format. Numbers too large to be written will be shown as ****. The number of header lines is four.

NOHEADER Output is written in floating point format to file and has no headers. It is intended primarily for processing by other programs. With some spreadsheet programs, the HEADER option works better.

INDEXED A table on file is produced which can be used directly (without editing) as input to ARCVIEW, ARCINFO, etc. The user should give two TABLE commands, one to produce a file with XP and YP as output quantities and the other with HS, PER or other output quantities for processing in ARCVIEW or ARCINFO.

The first column of each file produced by SWAN with this command is the sequence number of the output point. The last line of each file is the word END.

'fname' Name of the data file to which the output is to be written. Default for option HEADER is output to the print file (see **Section 4.11.2.1**). In case of NOHEADER, the filename is required.

|...|
< > The output parameters are the same as in command BLOCK.
|...|

OUTPUT The user requests output at various times. If the user does not use this option, the program will give TABLE output for the last time step of the computation.

[tbegtbl] Begin time of output, the format is:

= 1	ISO notation:	19870530.153000
= 2	(as in HP compiler):	`30-May-87 15:30:00'
= 3	(as in Lahey compiler):	05/30/87.15:30:00
= 4		15:30:00
= 5		87/05/30 15:30:00'
= 6	as in WAM	8705301530

This format is installation dependent. See installation manual on the SWAN web site. Default is ISO.

[deltbl] Time interval of output, the unit is indicated in the next option.

SEC Unit seconds.

MIN Unit minutes.

HR Unit hours.

DAY Unit days.

See command BLOCK in **Section 4.13.2.2** for details except when the x- and y-components of the vectorial quantities VEL, FORCE and TRANSPORT are given with respect to the problem coordinate system.

The number of decimals in the table varies for the output parameters. The number depends on the value of [hexp], given in the command QUANTITY.

4.13.2.4 SPECOUT

```
                        |    SPEC1D   |      | -> ABSOLUTE  |
SPECOUT 'sname'        <              >   <                 > 'fname'          &
                        | -> SPEC2D   |      |   RELATIVE   |

                                          | -> SEC  |
        OUTPUT [tbegspc] [deltspc]      <     MIN   >
                                          |    HR   |
                                          |    DAY  |
```

With the optional command SPECOUT, the user indicates that for each output location set to 'sname' (see **Sections 4.13.1.1-3, 4.13.1.6**) the 1-D or 2-D variance/energy (see command SET, **Section 4.12.4.2**), density spectrum (either the relative or absolute frequency spectrum) is to be written to a data file. The name 'fname' is required in this command.

'sname' Name of the set of POINTS, CURVE, FRAME or GROUP (see **Sections 4.13.1.1-3, 4.13.1.6**).

SPEC2D Means that 2-D (frequency-direction) spectra are written to file according to the format described in Appendix D. The following data will be written to the file heading:

- x and y (problem coordinates) of the output point,
- number of spectral directions and frequencies,
- spectral directions (Nautical or Cartesian definition, see command SET, **Section 4.12.4.2**) in degrees,
- spectral frequencies in Hz.

The following data will be written to the file body:

- values of the variance/energy (see command SET, **Section 4.12.4.2**) density for each spectral bin.

Note: This output file can be used for defining boundary conditions for subsequent SWAN runs (command BOUNDSPEC).

SPEC1D Means that 1-D (frequency) spectra are written to file according to the format described in Appendix D.

The following data will be written to the file heading:

- x and y (problem coordinates) of the output point,
- number of spectral frequencies,
- spectral frequencies in Hz.

The following data will be written to the file body:

- values of the variance/energy (see command SET, **Section 4.12.4.2**) density (integrated over directions) for each spectral frequency,
- values of the average direction for each spectral frequency (Nautical or

Cartesian definition, see command SET, **Section 4.12.4.2**) in degrees,

- values of the directional spreading for each spectral frequency.

Note: This output file can be used for defining boundary conditions for subsequent SWAN runs (command BOUNDSPEC).

ABS

Means that spectra are computed as a function of absolute frequency (i.e. the frequency as measured in a fixed point).

REL

Means that spectra are computed as function of relative frequency (i.e. the frequency as measured when moving with the current).

'fname'

Name of the data file to which the output is written.

OUTPUT

The user requests output at various times. If the user does not use this option, the program will give SPECOUT output for the last time step of the computation.

[tbegspc]

Begin time of output, the format is:

= 1	ISO notation	19870530.153000
= 2	(as in HP compiler):	`30-May-87 15:30:00'
= 3	(as in Lahey compiler):	05/30/87.15:30:00
= 4		15:30:00
= 5		`87/05/30 15:30:00'
= 6	as in WAM	8705301530

This format is installation dependent. See installation manual on the SWAN web site. Default is ISO.

[deltspc]

Time interval of output, the unit is indicated in the next option.

SEC

Unit seconds.

MIN

Unit minutes.

HR

Unit hours.

DAY

Unit days.

4.13.2.5 NESTOUT

```
                                                    | -> SEC |
NESTOUT 'sname' 'fname' OUTPUT [tbegnst] [deltnst]  <    MIN  >
                                                    |    HR   |
                                                    |    DAY  |
```

NESTOUT cannot be used in 1D-MODE.

With this optional command the user indicates that the spectra along a nest boundary 'sname' (see command NGRID, **Section 4.13.1.7**) should be written to a data file with name 'fname'. This name is required in this command.

'sname' Name of the set of output locations as defined in command NGRID.

'fname' Name of the data file to where the output is written. The file is structured according to the description in Appendix D and information about the boundary location is written to this file. SWAN will use this as a check for the subsequent nested run.

OUTPUT The user requests output at various times. If the user does not use this option, the program will give NESTOUT output for the last time step of the computation.

[tbegnst] Begin time of output, the format is:

= 1	ISO notation	19870530.153000
= 2	(as in HP compiler):	`30-May-87 15:30:00'
= 3	(as in Lahey compiler):	05/30/87.15:30:00
= 4		15:30:00
= 5		`87/05/30 15:30:00'
= 6	as in WAM	8705301530

This format is installation dependent. See installation manual on the SWAN web site. Default is ISO.

[deltnst] Time interval of output, the unit is indicated in the next option.

SEC Unit seconds.

MIN Unit minutes.

HR Unit hours.

DAY Unit days.

4.13.2.6 PLOTGEOGR

```
                     |-> xxx              |
PLOTGEOGR 'sname'   <                      >     'title'      <                >   &
                     |  FILE   'fname'    |                   | COORD [marg] |
                                |...      |
                     ( ISO      <          > [step] [min] [max] )                   &
                                |...      |

                                |...      |
                     ( VEC      <          > [scale] [dist] )                       &
                                |...      |

                     (CMESH) [ipen])                                                &

                     (SITES [ipen])               (LINES [ipen])                    &

                     (LOCATION ('sname') [ipen])                                    &

                                                  |-> SEC |
                     (OUTPUT [tbegnst] [deltnst] <    MIN  >
                                                  |   HR  |
                                                  |   DAY |
```

Option PLOTGEOGR cannot be used in 1-D mode.

With the optional command PLOTGEOGR the user indicates that a plot of a spatial distribution is required (contour line plot and/or vector plot). It is not possible to combine two isoline plots or two vector plots. For a combined contour line/vector plot the ISO option must precede the VEC option. CMESH, SITES and LOCATIONS can be combined with both vector and isoline plots.

The user should read **Section 4.7** to be aware of interpolation errors.

'sname' Name of the frame (must be a frame; see command FRAME, **Section 4.13.1.1**).

FILE With this option the user can define a filename for the plot file. In different PLOT... commands different plot files may be used. If the option FILE is used, a filename must be given for 'fname'. If this option is not used then the program will write the plot data to the same plot file as mentioned in the last PLOT... command. If a filename is not given in the first PLOT... command SWAN will create a plot file with the name PLF..., where the run number as defined in the command PROJECT as 'NR' appears on the dots (only for PLF...).

'fname' Name of the file where the plot output is to be written to (see **Section 4.12.3**).

xxx SWAN will assume a name for the plot file.

'title' String to be plotted in the caption of the plot. It has a maximum 36 character length. The project name and run number given in command PROJECT are also plotted. If the user does not give 'title', SWAN will generate a text indicating which quantity and which frame is plotted. The maximum length of the plot title is 36 characters.

COORDINATES If the keyword COORD is present the program will plot values of problem coordinates at the left and the lower side of the figure. The coordinates will be plotted only if the x-axis of the frame indicated by 'sname' is parallel to the x-axis of the problem coordinate system. If not, an error message results.

[marg] The size of the margin used to plot the coordinates (in cm). Default: [marg] = 1.5.

ISO Indicates that a contour line plot is requested for a scalar variable (i.e. not a vector, not a direction (DIR etc.)). The keywords identifying the variable are the same as in command BLOCK.

|... |
< >
|... |

The output parameters are the same as in command BLOCK. The scalar and vector parameters are separated here but DIR and PDIR are considered to be scalars in this context.

[step] The difference in value for two neighboring contour lines. If [step] is not specified, SWAN will make [step] equal to the difference between [min] and [max] divided by tem.

[min] The minimum value for which a contour line is drawn. If [min] is not specified then SWAN will make [min] equal to 0 if [step] is specified. If [step] is not specified, SWAN will make a rounded value near the smallest value occurring in the grid.

[max] The maximum value for which a contour line is drawn. If [max] is not specified then SWAN will make [max] equal to 10*[step] if [step] is specified, or a rounded value near the largest value occurring in the grid if [step] is not specified.

VEC Indicates that a vector plot is requested for a vector or direction variable. The keyword identifying the variable must follow the keyword VEC, as they are the same as in BLOCK. The length of the vectors is proportional to the magnitude of the variable (except for direction vectors, e.g. DIR, PDIR or TDIR). The start position of the vector is a grid point of the output frame.

[scale] Length of the vectors (in cm on paper) equal to the value of the vector (in SI units).

[dist] Every [dist]-th vector is plotted in the plot (in x- and y-direction), starting with the first vector in the lower left-hand corner of the FRAME.
Default: If option ISO is used in the command (i.e. vector and contour line plot combined), then [dist] = 5. If option ISO is not used in the command (i.e. only a vector plot is required), then [dist] = 1.

CMESH With this option the user indicates that a plot of the computational grid is to be plotted.

[ipen] Pen number to be used for the plot of the computational grid. If not given, the same pen is used as for the rest of the picture (usually black).

SITES Names of towns, regions, etc., as defined with the command SITES are plotted.

[ipen] Pen number to be used for the display of the SITES. If not given, the same pen is used as for the rest of the picture (usually black).

LINES Line(s) as defined with the command LINE is/are plotted.

[ipen] Pen number to be used for the display of the LINES. If not given, the same pen is used as for the rest of the picture (usually black).

LOCATION The locations of output point sets will be plotted in the figure. If a name is specified (see below), the corresponding point set is displayed. If no name is specified, all point sets will appear.

'sname' Name of a point set of which the location is to be plotted in the figure (second occurrence in the command scheme). If no 'sname' is given, all output point sets will be displayed.

[ipen] Pen number to be used for the display of the LOCATIONS. If not given, the same pen is used as for the rest of the picture (usually black).

OUTPUT The user requests output at various times. If the user does not use this option, the program will give PLOTGEOGR output for the last time step of the computation.

[tbegpltg] Begin time of output, the format is:

= 1	ISO notation	19870530.153000
= 2	(as in HP compiler):	30-May-87 15:30:00'
= 3	(as in Lahey compiler):	05/30/87.15:30:00
= 4		15:30:00

```
=5                              `87/05/30 15:30:00'
=6     as in WAM                8705301530
```

This format is installation dependent. See installation manual on the SWAN web site. Default is ISO.

[deltpltg] Time interval of output, the unit is indicated in the next option.

SEC Unit seconds.

MIN Unit minutes.

HR Unit hours.

DAY Unit days.

4.13.2.7 PLOTSTAR

```
                        |-> xxx             |           |->              |
PLOTSTAR 'sname'                            > 'title'  <                 >      &
                        |   FILE 'fname'    |           | COORD [marg]   |

                      |... |
          ( ISO      <     > [step] [min] [max] )                               &
                      |... |

          ( STAR       [scale] [dist] [bundle] )                                &

          (CMESH)      [ipen])                                                  &

          (SITES       [ipen]) (LINES [ipen])                                   &

          (LOCATION ('sname') [ipen])                                           &

                                          |-> SEC  |
          (OUTPUT [tbegnst] [deltnst]    <    MIN  >
                                          |   HR   |
                                          |   DAY  |
```

Option PLOTSTAR cannot be used in 1-D mode.

This PLOT... command can be used to show the directional distribution of the action transport (integrated over frequency) optionally superimposed on the scalar variables of command PLOTGEOGR.

Keywords and numerical input have the same meaning as in the PLOTGEOGR command, except for the items mentioned below.

```
 |...|
<     >        The output parameters are scalar parameters as in command PLOTGEOGRISO.
 |...|
```

STAR — Means that in a number of grid points in the output frame, lines are drawn which have the direction of the action transport associated with a spectral bundle, and whose length is proportional to the same action transport. The action transport is defined in Wm.

[scale] — Length of the vectors (in cm on paper) equal to the value of the vector (in SI units).

[dist] — See command PLOTGEOGR, **Section 4.13.2.6**.

[bundle] — Number of spectral directions that are represented by one arrow in the plot, i.e. the arrow shown represents the average action transport of a bundle of spectral directions. [bundle] is an integer. It must be a divisor of [mdc], the number of spectral directions in the computation. Default: [bundle] = 2.

4.13.2.8 PLOTPP

```
                        | -> xxx            |
PLOTPP 'sname'         <                     >  'title'                      &
                        | FILE 'fname'      |

                            | FROUDE        |
           ( PROBLEMS      <                 >  [symsiz]                     &
                            | PCONV         |

           (SITES [ipen])   (LINES [ipen])                                   &

           (LOCATION  ('sname') [ipen])                                      &

                                          | -> SEC  |
           (OUTPUT [tbegnst] [deltnst]   <     MIN   >
                                          |    HR   |
                                          |    DAY  |
```

Option PLOTPP cannot be used in 1-D mode.

SWAN will report that numerical problems have occurred, should they arise during the computation. This optional command shows the locations of the problem points by indicating that a plot of problem points is required. The points will be shown in the plot with a symbol specific to the nature of the problem. SITES and LOCATIONS can be combined with plots of problem points. It is not possible to combine this plot with isoline or vector plots.

Keywords and numerical input have the same meaning as in the PLOTGEOGR command, except for the items mentioned below.

FROUDE — If the current velocity is relatively large, with a Froude number larger than 0.8, it will be reduced such that the Froude number becomes equal to 0.8. (see command READINP CURRENT, **Section 4.12.7.2**).

PCONV — If the linear equation solver, used in the case of currents, does not converge in a grid point, the computation is not terminated and a warning is printed.

If none of the above alternatives is mentioned, both types of problem points will be shown.

[symsiz] — The size of the symbols as they appear in the plot, in cm. Default: [symsiz] = 0.18 (cm).

4.13.2.9 PLOTSPEC

```
                          | -> xxx                  |
PLOTSPEC 'sname'          <                         >    'title'                 &
                          |   FILE 'fname'          |

                          | -> NORMALIZED     |
       SPECTRUM           <                         >      ( < [ch] > )          &
                          |    NONORMALIZED|

                     | -> [fmax] [fmid]    |      | ->  ABS    |
       ( FREQ     <                        >   <               >)                &
                     |   NORMALIZED        |      |     REL    |

                                        | -> SEC |
       (OUTPUT [tbegplts] [deltplts])   <    MIN >
                                        |    HR  |
                                        |    DAY |
```

With this optional command the user instructs SWAN to produce a polar plot of the 2-D spectrum at POINTS or along a CURVE (see **Section 4.13.1.6** or **4.13.1.3**).

'sname' Name of the points or curve (see **Section 4.13.1.6** or **4.13.1.3**).

FILE See command PLOTGEOGR, **Section 4.13.2.6**.

'fname' Name of the file where the plot output is written to (see **Section 4.12.3**).

xxx SWAN will assume a name for the plot file.

'title' String to be plotted in the caption of the plot, maximum 36 characters long. The project name and the run number given in the command PROJECT are plotted. The maximum length of the plot title is 36 characters. Default: title = ' '.

SPECTRUM Indicates that a polar plot of the spectrum is requested.

NORMALIZED The variance/energy densities (see command SET, **Section 4.12.4.2**) in the 2-D plot will be normalized, i.e. the ratio of the variance/energy density and the peak variance/energy density is plotted; consequently the values of [ch] (see below) must be smaller than one. Note that the 1-D spectrum is always normalized.

NONORMALIZED The variance/energy densities (see command SET, **Section 4.12.4.2**) in the 2-D plot will not be normalized. Note that the 1-D spectrum is always normalized.

[ch] Contour levels for the variance/energy (see command SET, **Section 4.12.4.2**) density. In contrast with the plot as described in the commands PLOTSTAR ISO and PLOTGEOGR ISO the contour levels are not equidistant. Because of the large variation in values of variance/energy densities (see command SET, **Section 4.12.4.2**) one is advised to distribute the levels in a logarithmic manner, e.g., 0.01, 0.03, 0.1, 0.3, 1., 3., 10. (in m 2 /Hz/Rad).
Note: If no values are given for [ch] the program will use 0.02, 0.1 and 0.5 (it is assumed that these are normalized values).

FREQUENCY The part after this keyword serves to define the frequency scale of the figure.

[fmax] The maximum frequency for which the variance/energy density (see command SET, **Section 4.12.4.2**) is plotted. Default [fmax] = 0.5*[fhigh] (see command CGRID, **Section 4.12.6.1**). SWAN will plot a dashed circle for this frequency. **Note:** The program will also plot a circle for the frequency [flow] (see command CGRID, **Section 4.12.6.1**).

[fmid] An additional circle will be plotted at frequency [fmid]. [fmid] must be

less than [fmax]. Default: no circle.

NORMALIZED — The frequency scale is normalized with the peak frequency. If this keyword does not appear, non-normalized values for the frequency are assumed.

ABSOLUTE — The spectral density is defined along the absolute frequency axis, i.e., the frequency as measured in a geographically fixed location. This is the default option.

RELATIVE — The spectral density is defined along the relative frequency axis, i.e., the frequency as measured in a point moving with the current.

OUTPUT — The user requests output at various times. If the user does not use this option, the program will give PLOTSPEC output for the last time step of the computation.

[tbegplts] — Begin time of output, the format is:

= 1	ISO notation	19870530.153000
= 2	(as in HP compiler):	\`30-May-87 15:30:00'
= 3	(as in Lahey compiler):	05/30/87.15:30:00
= 4		15:30:00
= 5		87/05/30 15:30:00'
= 6	as in WAM	8705301530

This format is installation dependent. See installation manual on the SWAN web site. Default is ISO.

[deltplts] — Time interval of output, the unit is indicated in the next option.

SEC — Unit seconds.

MIN — Unit minutes.

HR — Unit hours.

DAY — Unit days.

4.13.3 Compute, Hotfile and Stop

4.13.3.1 COMPUTE

```
                      | STATIONARY [time]                                      |
COMPUTE      ( <                                                                  >)
                      |                                  | -> SEC  |             |
| -> NONSTAT [tbegc] [deltc]   <  MIN  >     [tendc]    |
                               |  HR   |
                               |  DAY  |
```

COMPUTE orders SWAN to start the computation(s).

If the SWAN mode is stationary (see command MODE, **Section 4.12.4.3**), then the only command given is COMPUTE. If the SWAN mode is nonstationary (see command MODE, **Section 4.12.4.3**), then the computation can be either stationary (at the specified time: option STATIONARY) or nonstationary (over the specified period of time: [tbegc], etc.).

To verify input to SWAN such as water depth and wind fields, SWAN can be run without computations (zero iterations by using command NUM ACCUR itermx = 0).

In the case MODE NONSTATIONARY, several COMPUTE commands may appear, where the wave state at the end of one computation is used as initial state for the next one, unless a command INIT appears in between the two COMPUTE commands. This enables the user to make a stationary computation to obtain the initial state for a nonstationary computation and/or to change a boundary condition, the computational time step during a computation, etc.

STATIONARY A stationary computation is to be made.

[time] Time level for which the stationary run is to be made, the format is:

= 1	ISO notation	19870530.153000
= 2	(as in HP compiler):	`30-May-87 15:30:00'
= 3	(as in Lahey compiler):	05/30/87.15:30:00
= 4		15:30:00
= 5		`87/05/30 15:30:00'
= 6	as in WAM	8705301530

This format is installation dependent. See installation manual on the SWAN web site. Default is ISO.

NONSTATIONARY A nonstationary computation is to be made.

[tbegc] The start date and time of the nonstationary computation. For format, see [time]. Default: The time read from a hotfile (see command INIT HOTSTART, **Section 4.12.8.7**), the end time [tendc] of the previous nonstationary computation or the [time] of the previous stationary computation in the same SWAN run (if any).

[deltc] The time step of the nonstationary computation. The unit is indicated in the following options.

SEC Unit seconds.

MIN Unit minutes.

HR Unit hours.

DAY Unit days.

[tendc] The end time of the nonstationary computation. For format, see [time].

4.13.3.2 HOTFILE

```
HOTFILE 'fname'
```

This command can be used to write the entire wave field at the end of a computation to an initialization file, e.g. to be used as initial condition in a subsequent SWAN run (see command INITIAL HOTSTART, **Section 4.12.8.7**). This command must be entered immediately after a COMPUTE command.

The format of the initialization file is identical to the format of the files written by SPECOUT (option SPEC2D).

`fname' Name of the file to which the wave field is written.

4.13.3.3 STOP

```
STOP
```

This required command marks the end of the commands in the command file.
Note: The command STOP must always be the last command in the input file; any information in the input file beyond this command is ignored.

4.14 Definitions of Variables

In SWAN a number of variables mostly related to waves are used as input and output. The definitions of these variables are mostly conventional.

HSIGN Significant wave height (H_s in m), defined as:

$H_s = 4\sqrt{\int\int E(\omega,\theta) d\omega d\theta}$, where $E(\omega, \theta)$ is the variance density spectrum.

HSWELL Significant wave height (H_{swell} in m), associated with the low frequency part of the spectrum, defined as:

$H_{swell} = 4\sqrt{\int_0^{\omega_{swell}} \int_0^{2\pi} E(\omega,\theta) d\omega d\theta}$, where $E(\omega, \theta)$ is the variance density spectrum,

and $\omega_{swell} = 2\pi f_{swell}$. f_{swell} is 0.1 Hz by default. It can be changed in the command QUANTITY.

TM01 Mean absolute wave period (in s) of $E(\omega, \theta)$, defined as:

$$T_{m01} = 2\pi \left(\frac{\iint \omega E(\sigma,\theta) d\sigma d\theta}{\iint E(\sigma,\theta) d\sigma d\theta} \right)^{-1} = 2\pi \left(\frac{\iint \omega E(\omega,\theta) d\omega d\theta}{\iint E(\omega,\theta) d\omega d\theta} \right)^{-1}$$

where ω is the absolute radian frequency, determined by the Doppler shifted dispersion relation.

TM02 Mean absolute wave period (in s) of $E(\omega, \theta)$, defined as:

$$T_{m02} = 2\pi \left(\frac{\iint \omega^2 E(\omega,\theta) d\omega d\theta}{\iint E(\omega,\theta) d\omega d\theta} \right)^{-\frac{1}{2}} = 2\pi \left(\frac{\iint \omega^2 E(\sigma,\theta) d\sigma d\theta}{\iint E(\sigma,\theta) d\sigma d\theta} \right)^{-\frac{1}{2}}$$

where ω is the absolute radian frequency, determined by the Doppler shifted dispersion relation.

DIR Mean wave direction (in degrees, Cartesian or Nautical convention), as conventionally defined (Kuik et al., 1988).

$$[\mathrm{DIR}] = \arctan \left| \int \sin(\theta) E(\sigma,\theta) d\sigma d\theta \,/\, \int \cos(\theta) E(\sigma,\theta) d\sigma d\theta \right|$$

DIR is the direction normal to the wave crests. If currents are present, DIR is different from the direction of the energy transport TDIR.

PDIR — Peak direction of $E(\theta) = \int E(\omega,\theta)d\omega = \int E(\sigma,\theta)d\sigma$ (in degrees, Cartesian or Nautical convention).

TDIR — Direction of energy transport (in degrees, Cartesian or Nautical convention).

RTM01 — Mean relative wave period (in s) of $E(\sigma, \theta)$, defined as:

$$RT_{m01} = 2\pi\left(\iint \sigma E(\sigma,\theta)d\sigma d\theta \,/\, \iint E(\sigma,\theta)d\sigma d\theta\right)^{-1}$$

Equal to TM01 in the absence of currents.

RTP — Relative peak period (in s) of $E(\sigma)$ (equal to absolute peak period in the absence of currents).

PER — Average absolute period (in s) defined as:

$$T_{m,p-1,p} = 2\pi \int \omega^{p-1}E(\sigma,\theta)d\sigma d\theta \,/\, \int \omega^{p}E(\sigma,\theta)d\sigma d\theta$$

The power p can be chosen by using QUANTITY. If $p = 1$ (the default value in QUANTITY) PER and RPER are identical to TM01 and RTM01, respectively. If $p = 0$, the average period is obtained as used in WAM, i.e. T(m,-1,0).

RPER — Average relative period (in s) defined as:

$$T_m = 2\pi \int \sigma^{p-1}E(\sigma,\theta)d\sigma d\theta \,/\, \int \sigma^{p}E(\sigma,\theta)d\sigma d\theta$$

The power p can be chosen by using QUANTITY. If $p = 1$ (the default value in QUANTITY), PER and RPER are identical to TM01 and RTM01, respectively. If $p = 0$, the average period is obtained as used in WAM, i.e. Tm(-1,0).

FSPR — The normalized frequency width of the spectrum (frequency spreading), as defined by Battjes and van Vledder (1984):

$$FSPR = \left|\int_0^\infty E(\omega)e^{i\omega\tau}d\omega\right| / E_{tot} \qquad \text{for } \tau = T_{m02}$$

DSPR — The one-sided directional width of the spectrum (directional spreading or directional standard deviation, in degrees), defined as:

$$DSPR^2 = \left(\frac{180}{\pi}\right)^2 \int_0^{2\pi} \{2\sin((\theta-\bar{\theta})/2)\}^2 D(\theta)d\theta,$$

computed conventionally for pitch-and-roll buoy data (Kuik et al., 1988).

This is the standard definition for WAVEC buoys integrated over all frequencies:

$$(DSPR\cdot\frac{\pi}{180})^2 = 2\left(1-\left[(\int\sin(\theta)\frac{\int E(\sigma,\theta)d\sigma}{\int E(\sigma)d\sigma}d\theta)^2+(\int\cos(\theta)\frac{\int E(\sigma,\theta)d\sigma}{\int E(\sigma)d\sigma}d\theta)^2\right]^{1/2}\right)$$

[MS] SWAN input in the commands BOUNDPAR and BOUNDSPEC defined as the directional distribution of incident wave energy, D(θ) = A {cos(θ)}**[MS], at all frequencies. [MS] is not necessarily an integer but is, for this directional distribution, related to the one-sided directional spread of the waves (DSPR) as follows:

[MS]	DSPR (°)
1.	37.5
2.	31.5
3.	27.6
4.	24.9
5.	22.9
6.	21.2
7.	19.9
8.	18.8
9.	17.9
10.	17.1
15.	14.2
20.	12.4
30.	10.2
40.	8.9
50.	8.0
60.	7.3
70.	6.8
80.	6.4
90.	6.0
100.	5.7
200.	4.0
400.	2.9
800.	2.0

DISSIP Energy dissipation per unit time due to the sum of bottom friction, whitecapping and depth induced wave breaking (in W/m^2 of m^2/s, depending on command SET).

WLEN The mean wavelength,

$$WLEN = 2\pi \left(\frac{\int k^{p} E(\sigma,\theta) d\sigma d\theta}{\int k^{p-1} E(\sigma,\theta) d\sigma d\theta} \right)^{-1} ,$$

see command QUANTITY, **Section 4.13.2.1** (where $p = 1$ is default).

STEEPNESS Wave steepness, computed as:

STEEPNESS = HSIGN / WLEN

Qb Fraction of breakers in expression of Battjes and Janssen (1978), see **Section 5.1.**

TRANSP Energy transport with components $P_x = \iint \rho g c_x E(\sigma,\theta) d\sigma d\theta$ and $P_y = \iint \rho g c_y E(\sigma,\theta) d\sigma d\theta$, where x and y relate to the problem coordinate system. The exception is output with BLOCK in combination with FRAME, where x and y relate to the x-axis and y-axis of the output frame.

VEL Current velocity with components in x- and y-direction of the problem coordinate system. The exception is output with BLOCK in combination with FRAME, where x and y relate to the x-axis and y-axis of the output frame.

WIND Wind velocity with components in x- and y-direction of the problem coordinate system. The exception is output with BLOCK in combination with FRAME, where x and y relate to the x-axis and y-axis of the output frame.

FORCE Wave induced force per unit surface area (gradient of the radiation stresses) where x and y relate to the problem coordinate system. The exception is output with BLOCK in combination with FRAME, where x and y relate to the x-axis and y-axis of the output frame.

$$F_x = -\frac{\partial S_{xx}}{\partial x} - \frac{\partial S_{xy}}{\partial y}, \text{ and } F_y = -\frac{\partial S_{yx}}{\partial x} - \frac{\partial S_{yy}}{\partial y},$$

where S is the radiation stress tensor:

$$S_{xx} = \rho g \int [n \cos^2\theta + n - \tfrac{1}{2}] E d\sigma d\theta$$

$$S_{xy} = S_{yx} = \rho g \int n \sin\theta \cos\theta E d\sigma d\theta$$

$$S_{yy} = \rho g \int [n \sin^2\theta + n - \tfrac{1}{2}] E d\sigma d\theta$$

and n is the ratio of group velocity over phase velocity.

URMS Root-mean-square value of the orbital motion near the bottom $U_{rms} = \langle (U^2) \rangle^{1/2}$

UBOT Root-mean-square value of the maxima of the orbital motion near the bottom.
$U_{bot} = \sqrt{2} U_{rms}$.

LEAK Numerical loss of energy equal to $c_\theta E(\omega, \theta)$ across boundaries θ_1 = [dir1] and θ_2 = [dir2] of a directional sector (see command CGRID, **Section 4.12.6.1**).

TIME Full date-time string.

TSEC Time in seconds with respect to a reference time (see command QUANTITY, **Section 4.13.2.1**).

SETUP The elevation of mean water level (relative to still water level) induced by the radiation stress gradient of the waves.

Cartesian Direction convention: The direction is the angle between the vector and the positive x-axis, measured counter-clockwise (the direction to where the waves are going or to where the wind is blowing).

Nautical Direction convention: The direction + 180° of the vector from geographic North measured clockwise (the direction from where the waves are coming or from where the wind is blowing).

5.0 Functional Description

5.1 Model Formulation

5.1.1 General Formulation

The waves in SWAN are described with the two-dimensional wave action density spectrum, even when nonlinear phenomena dominates, such as in the surf zone. The rationale for using the spectrum in such highly nonlinear conditions is that even in such conditions it seems possible to predict with reasonable accuracy spectral distribution of the second order moment of the waves (although it may not be sufficient to fully describe the waves statistically). The action density spectrum rather than the energy density spectrum is considered in SWAN, since in the presence of currents, action density is conserved whereas energy density is not (e.g., Whitham, 1974). The independent variables are the relative frequency (as observed in a frame of reference moving with the action propagation velocity) and the wave direction, which is normal to the wave crest of each spectral component. The action density is equal to the energy density divided by the relative frequency. In SWAN, this spectrum may vary in time and space.

5.1.1.1 Action Balance Equation

The evolution of the wave spectrum in SWAN is described by the spectral action balance equation, which for Cartesian coordinates is (e.g., Hasselmann et al., 1973) defined by:

$$\frac{\partial}{\partial t}N+\frac{\partial}{\partial x}c_xN+\frac{\partial}{\partial y}c_yN+\frac{\partial}{\partial\sigma}c_\sigma N+\frac{\partial}{\partial\theta}c_\theta N=\frac{S}{\sigma} \qquad (1a)$$

The first term in the left-hand side of this equation represents the local rate of change of action density in time. The second and third term represent propagation of action in geographical space, with propagation velocities c_x and c_y in x- and y-space, respectively. The fourth term represents shifting of the relative frequency due to variations in depths and currents, with propagation velocity c_σ in σ-space. The fifth term represents depth-induced and current-induced refraction with propagation velocity c_θ in θ -space. The expressions for these propagation speeds are taken from linear wave theory (as seen in Whitham, 1974; Mei, 1983; Dingemans, 1997). The term S (= $S(\sigma,\theta)$) at the right hand side of the action balance equation is the source term in terms of energy density representing the effects of generation, dissipation and nonlinear wave-wave interactions. A brief summary of the formulations that are used for the various source terms in SWAN is given next. More details are given in **Section 5.1.3**.

In view of the use of SWAN at shelf sea or oceanic scale the user can choose to express the basic equation in spherical coordinates:

$$\frac{\partial}{\partial t}N+\frac{\partial}{\partial\lambda}c_\lambda N+(\cos\varphi)^{-1}\frac{\partial}{\partial\varphi}c_\varphi\cos\varphi N+\frac{\partial}{\partial\sigma}c_\sigma N+\frac{\partial}{\partial\theta}c_\theta N=\frac{S}{\sigma}\quad , \qquad (1b)$$

with longitude, λ and latitude, φ.

5.1.1.2 Wind Input

Transfer of wind energy to waves is described in SWAN with a resonance mechanism (Phillips, 1957) and a feedback mechanism (Miles, 1957). The corresponding source term for these mechanisms is commonly described as the sum of linear and exponential growth:

$$S_{in}(\sigma,\theta)=A+BE(\sigma,\theta), \qquad (2)$$

in which A describes linear growth and BE exponential growth. A and B depend on wave frequency and direction, and wind speed and direction. The effects of currents are accounted for by using the apparent local wind speed and direction. The expression for the term A is due to Cavaleri and Malanotte-Rizzoli (1981), with a filter to avoid growth at frequencies lower than the Pierson-Moskowitz frequency (Tolman, 1992a). Two optional expressions for the coefficient B are used in the model. The first is taken from an early version of the WAM model (known as WAM Cycle 3, the WAMDI group, 1988). It is due to Snyder et al. (1981), rescaled in terms of friction velocity *U* by Komen et al. (1984). The drag coefficient to relate *U* to the driving wind speed at 10 m elevation U_{10} is taken from Wu (1982). The second expression for *B* in SWAN is taken from the most recent version of the WAM model (known as WAM Cycle 4, Komen et al., 1994). It is due to Janssen (1991a), and it accounts explicitly for the interaction between the wind and the waves by considering atmospheric boundary layer effects and the roughness length of the sea surface. The corresponding set of equations is solved (as in the WAM model) with the iterative procedure of Mastenbroek et al. (1993).

5.1.1.3 Dissipation

The dissipation term of wave energy is represented by the summation of three different contributions: whitecapping, $s_{ds,w}(\sigma, \theta)$, bottom friction, $s_{ds,b}(\sigma, \theta)$, and depth-induced breaking, $s_{ds,br}(\sigma, \theta)$.

Whitecapping is primarily controlled by the steepness of the waves. In presently operating third-generation wave models (including SWAN) the whitecapping formulations are based on a pulse-based model (Hasselmann, 1974), as adapted by the WAMDI group (1988):

$$s_{ds,w}(\sigma,\theta) = -\Gamma\tilde{\sigma}\frac{k}{\tilde{k}}E(\sigma,\theta), \qquad (3)$$

where Γ is a steepness dependent coefficient, *k* is wave number, and $\tilde{\sigma}$ and $\tilde{k}$ denote a mean frequency and a mean wave number, respectively (cf. the WAMDI group, 1988). Komen et al. (1984) estimated the value of Γ by closing the energy balance of the waves in fully developed conditions. This implies that the Γ value depends on the wind input formulation used. Since two expressions are used for SWAN wind input, two values for Γ are also used. The first is due to Komen et al. (1984, as in Cycle 3 of the WAM model) and is used when the wind input coefficient *B* (Komen et al., 1984) is used. The second expression is an adaptation based on Janssen (1991a; as in Cycle 4 of the WAM model; see Janssen, 1991b; Günther et al., 1992) and used when the wind input term *B* of Janssen (1991a) is used. Young and Banner (1992) and Banner and Young (1994) have shown that the results of closing the energy balance in this manner depend critically on the choice of a high-frequency cut-off frequency above which a diagnostic spectral tail is used. In SWAN, this cut-off frequency is different that used in WAM. Differences in the growth rates between WAM and SWAN are therefore to be expected.

Depth-induced dissipation may be caused by bottom friction, bottom motion, percolation, or backscattering on bottom irregularities (Shemdin et al., 1978). For continental shelf seas with sandy bottoms, the dominant mechanism appears to be bottom friction (Bertotti and Cavaleri, 1994) which can be represented as:

$$S_{ds,b}(\sigma,\theta) = -C_{bottom}\frac{\sigma^2}{g^2\sinh^2(kd)}E(\sigma,\theta), \qquad (4)$$

in which C_{bottom} is a bottom friction coefficient that generally depends on the bottom orbital motion represented by U_{rms}. Many models have been proposed since the pioneering paper of Putnam and Johnson (1949). Hasselmann et al. (JONSWAP, 1973) suggested using an empirically obtained constant that seems to perform well in a variety of conditions as long as a suitable value is chosen (typically different for swell and wind sea; Bouws and Komen, 1983). A nonlinear formulation based on drag was proposed by Hasselmann and Collins (1968) and simplified by Collins (1972). More complicated, eddy viscosity models have been developed by Madsen et al. (1988) and by Weber (1989, 1991a, 1991b). Considering the large bottom condition variations in coastal areas (bottom material, bottom roughness length, ripple height etc.), there is no field data evidence to give preference to a particular friction model (Luo and Monbaliu, 1994). For this reason, the simplest of each of these friction models has been implemented in SWAN: the empirical JONSWAP model of Hasselmann et al. (1973), the drag law model of Collins (1972), and the eddy-viscosity model of Madsen et al. (1988). The effect of a mean current on the wave energy dissipation due to bottom friction is not taken into account in SWAN. The reasons for this are given by Tolman (1992b), who argues that state-of-the-art expressions vary too widely in their effects to be acceptable. He found that the error in computing a correct estimate of the bottom roughness length scale has a much larger impact on the energy dissipation rate than the effect of a mean current.

The process of depth-induced wave breaking is still poorly understood and little is known about its spectral modeling. In contrast to this, the total dissipation (i.e., integrated over the spectrum) due to this type of wave breaking can be well modeled with the dissipation of a bore applied to the breaking waves in a random field (Battjes and Janssen, 1978; Thornton and Guza, 1983). Laboratory observations (e.g., Battjes and Beji, 1992; Vincent et al. 1994; Arcilla et al., 1994 and Eldeberky and Battjes, 1996) show that the shape of initially uni-modal spectra propagating across simple (barred) beach profiles is insensitive to depth-induced breaking. This has led Eldeberky and Battjes (1995) to formulate a spectral version of the bore model of Battjes and Janssen (1978) that conserves the spectral shape. Expanding their expression to include directions, the expression used in SWAN is:

$$S_{ds,br}(\sigma,\theta) = \frac{D_{tot}}{E_{tot}} E(\sigma,\theta), \tag{5}$$

in which E_{tot} is the total wave energy and D_{tot}(which is negative) is the rate of dissipation of the total energy due to wave breaking. (Battjes and Janssen, 1978). Adding a quadratic dependency on frequency, as suggested by Mase and Kirby (1992), seems to have no noticeable effect on SWAN results. Chen and Guza (1997) inferred (from observations and simulations with a Boussinesq model) that the high-frequency levels are insensitive to such frequency dependency because increased nonlinear energy transfer compensates an increased dissipation at high frequencies; however, they did find the frequency dependency to be relevant in time domain. The value of D_{tot} depends critically on the breaking parameter $\gamma = H_{max}/d$, in which H_{max} is the maximum possible individual wave height in the local water depth. In SWAN, a constant value and a variable value are available. The constant value is $\gamma = 0.73$ (the mean value of the data set of Battjes and Stive, 1985).

SWAN can estimate wave transmission through a (line-) structure such as a breakwater or dam. Such an obstacle will affect the wave field in two ways, first by reducing the wave height locally all along its length, and second by causing diffraction around its end(s). The model is not able to account for diffraction. In irregular, short-crested wave fields, however, it seems that the effect of diffraction is small, except in a region less than one or two wavelengths away from the tip of the obstacle (Booij et al., 1993). Therefore, the model can reasonably account for waves around an obstacle if the directional spectrum of incoming waves is not too narrow. Since obstacles usually have a transversal area that is too small to be resolved by the bottom grid in SWAN, an obstacle is modeled as a line. If the crest of the breakwater is at a level where at least a portion of the waves can pass over, the transmission coefficient K_t is a function of wave height and the difference in crest level and water level. K_t is defined as the ratio of the significant wave height at the downwave side of the dam over the (significant) wave height at the upwave side (Goda et al., 1967):

$$K_t = 0.5\left[1-\sin\left(\frac{\pi}{2\alpha}\left(\frac{F}{H_i}+\beta\right)\right)\right] \quad \text{for } -\beta-\alpha<\frac{F}{H_i}<\alpha-\beta, \tag{6}$$

where $F = h\text{-}d$ is the freeboard of the dam, H_i is the incident significant wave height at the upwave side of the obstacle (dam), h is the crest level of the dam above the reference level, d is the mean water level relative to the reference level, and the coefficients α, β depend on the shape of the dam. Table 5.1-1 provides a summary of the α, β coefficients and the cases in which they are used.

Table 5.1-1. Coefficients α, β determined by the shape of the dam (Seelig, 1979).

Case	α	β
vertical thin wall	1.8	0.1
caisson	2.2	0.4
dam with slope 1:3/2	2.6	0.15

The above expression is based on experiments in a wave flume, so it is only valid for normal incidence waves. No data are available on oblique waves so it is assumed that the transmission coefficient does not depend on direction. A change in wave frequency may also be expected since the process above the dam is often highly nonlinear. Again, there is little information available, so it is assumed that the frequencies remain unchanged over an obstacle. Only the energy scale of the spectrum is affected, not the spectral shape.

5.1.1.4 Nonlinear Wave-wave Interactions

In deep water, quadruplet wave-wave interactions dominate the evolution of the spectrum. They transfer wave energy from the spectral peak to lower frequencies, thus moving the peak frequency to lower values and to higher frequencies where the energy is dissipated by whitecapping. In very shallow water, triad wave-wave interactions transfer energy from lower frequencies to higher frequencies, resulting in higher harmonics (Beji and Battjes, 1993). Low-frequency energy generation by triad wave-wave interactions is not considered here.

A full computation of the quadruplet wave-wave interactions is extremely time consuming and not convenient for any operational wave model. A number of techniques based on parametric methods or other approximations have been proposed to improve computational speed (see Young and Van Vledder, 1993 for a review). SWAN's computations are carried out with the Discrete Interaction Approximation (DIA) of Hasselmann et al. (1985). DIA has been deemed quite successful in describing the essential features of a developing wave spectrum (Komen et al., 1994), but for uni-directional waves, the approximation is not valid. In fact, the quadruplet interaction coefficient for these waves is nearly zero (G.Ph. van Vledder, personal communication, 1996). For finite-depth applications, Hasselmann and Hasselmann (1981) have shown that for a JONSWAP-type spectrum the quadruplet wave-wave interactions may be scaled with a simple expression. This expression is used in SWAN.

Abreu et al. (1992) first described triad wave-wave interactions in terms of a spectral energy source term. However, their expression is restricted to non-dispersive shallow water waves and is not suitable in many practical wind waves applications. A breakthrough came from Eldeberky and Battjes (1995), who transformed the amplitude portion of the Boussinesq model of Madsen and Sørensen (1993) into an energy density formulation and parameterized the biphase of the waves on the basis of laboratory observations (Battjes and Beji, 1992; Arcilla et al., 1994). A discrete triad approximation (DTA) for co-linear waves was subsequently obtained by considering only the dominant self-self interactions. Their model has been verified with flume observations of long-crested, random waves breaking over a submerged bar (Beji and Battjes, 1993) and over a barred beach (Arcilla et al., 1994). The model appeared successful in describing the essential features of the energy transfer, from the primary peak of the spectrum to the super harmonics. Eldeberky (1996) later derived a slightly different version, the Lumped Triad Approximation (LTA). LTA is used in SWAN.

5.1.2 First-, Second- and Third-generation Mode

SWAN operates in first-, second- and third-generation mode. The first- and second-generation modes are essentially those of Holthuijsen and de Boer (1988) as indicated above, with the first-generation having a constant Phillips constant of 0.0081 and the second-generation a variable Phillips constant. An overview of the options is given below in Table 5.1-2.

Table 5.1-2. SWAN Generation Mode Options.

		Generation mode of SWAN		
		1st	2nd	3rd
Linear wind growth:	Cavaleri & Malanotte-Rizzoli (1981) [modified]	x	x	
	Cavaleri & Malanotte-Rizzoli (1981)			x
Exponential wind growth:	Snyder et al. (1981) [modified]	x	x	
	Snyder et al. (1981)			x[1]

	Janssen (1989, 1991)			x[2]
Whitecapping:	Holthuijsen and de Boer (1988)	x[3]	x[4]	
	Komen et al. (1984)			x[1]
	Janssen (1991), Komen et al. (1994)			x[2]
Quadruplet interaction:	Hasselmann et al. (1985)			x
Triad interactions:	Eldeberky (1996)	x	x	x
Depth-induced breaking:	Battjes & Janssen (1978)	x	x	x
Bottom friction:	Hasselmann et al. JONSWAP (1973)	x	x	x
	Collins (1972)	x	x	x
	Madsen et al. (1988)	x	x	x
Obstacle transmission:	Seelig (1979)	x	x	x

[1] Gives the wind input and whitecapping formulations as used in WAM Cycle 3 (see **Section 4.12.9.3**).

[2] Gives the wind input and whitecapping formulations as used in WAM Cycle 4 (see **Section 4.12.9.3**).

[3] Pierson-Moskowitz spectrum as upper limit.

[4] Scaled Pierson-Moskowitz spectrum as upper limit.

5.1.3 Wave-induced Set-up

The computation of the wave-induced set-up for a geographic 1-D case is based on the vertically integrated momentum balance equation that reduces to a balance between the wave radiation stress gradient and the hydrodynamic pressure gradient (no wave-induced currents exist). In a 2-D case, the computation of the wave-induced set-up is based on the divergence of the vertically integrated momentum balance equation equaling zero.

5.1.4 Detailed Formulation

The following sections describe the expressions for the physical processes of generation, dissipation and nonlinear wave-wave interactions available in SWAN.

5.1.4.1 Input by Wind (S_{in})

Wave growth by wind is described in **Section 5.1.1.2** by equation (2). The SWAN model is driven by the wind speed at 10-m elevation, U_{10}, whereas the computations use the friction velocity U_*. For the WAM Cycle 3 formulation the transformation from U_{10} to U_* is obtained with:

$$U_*^2 = C_D U_{10}^2, \tag{7}$$

in which C_D is the drag coefficient from Wu (1982):

$$C_D(U_{10}) = \begin{cases} 1.2875\times10^{-3} & \text{for}\, U_{10} < 7.5\,\text{m/s} \\ (0.8 + 0.065\,\text{s/m}\times U_{10})\times10^{-3} & \text{for}\, U_{10} \geq 7.5\,\text{m/s} \end{cases}, \tag{8}$$

For the WAM Cycle 4 formulations, the computation of U_* is an integral part of the source term.

5.1.4.2 Linear Growth by Wind:

For the linear growth term A, the expression by Cavaleri and Malanotte-Rizzoli (1981) is used with a filter to eliminate wave growth at frequencies lower than the Pierson-Moskowitz frequency (Tolman, 1992a) (Note that in Tolman's Eq. 9 the 10^{-5} should be 10^{-3}. H. Tolman, personal communication, 1995):

$$A = \frac{1{,}5\times10^{-3}}{g^2 2\pi}[U_* \max\,[0, \cos(\theta - \theta_w)]\,]^4 H \quad ,$$

$$H = \exp(-(\sigma/\sigma^*_{PM})^{-4}) \quad \text{with} \quad \sigma^*_{PM} = \frac{0.13g}{28U_*}2\pi \quad , \tag{9}$$

in which θ_w is the wind direction, H is the filter and $\theta_{PM}{}^*$ is the peak frequency of the fully developed sea state according to Pierson and Moskowitz (1964; reformulated in terms of friction velocity).

5.1.4.3 Exponential Growth by Wind:

SWAN has two expressions for exponential growth by wind optionally available. The first is a Komen et al. (1984) expression that is a function of $\frac{U_*}{C_{ph}}$:

$$B = \max\left[0, 0.25\frac{\rho_a}{\rho_w}\left[28\frac{U_*}{C_{ph}}\cos(\theta - \theta_w) - 1\right]\right]\sigma \quad , \tag{10}$$

in which c_{ph} is the phase speed and ρ_a and ρ_w are the density of air and water, respectively. This expression is also used in WAM Cycle 3 (the WAMDI group, 1988). The second is a Janssen (1989, 1991) expression based on a quasi-linear wind-wave theory given by:

$$B = \beta\frac{\rho_a}{\rho_w}\left(\frac{U_*}{c_{ph}}\right)^2 \max[0, \cos(\theta - \theta_w)]^2\sigma \quad , \tag{11}$$

where β is the Miles "constant". In the theory of Janssen (1991), the Miles "constant" is estimated from the non-dimensional critical height λ:

$$\begin{cases} \beta = \dfrac{1.2}{\kappa^2} \lambda \ln^4 \lambda \ , & \lambda \leq 1 \\ \lambda = \dfrac{g\, z_e}{c_{ph}^2} e^r , & r = \kappa c / \left| U_* \cos(\theta - \theta_w) \right| \end{cases} , \qquad (12)$$

where κ is the Von Karman constant, equal to 0.41, and z_e is the effective surface roughness. If the non-dimensional critical height $\lambda > 1$, the Miles constant β is set equal to zero. Janssen (1991) assumes the wind profile is given by:

$$U(z) = \frac{U_*}{\kappa} \ln\left(\frac{z + z_e - z_o}{z_e} \right) \ , \qquad (13)$$

in which $U(z)$ is the wind speed at height z (10 m for SWAN) above the mean water level, z_o is the roughness length. The effective roughness length z_e depends on the roughness length z_o and the sea state through the wave induced stress, τ_w, and the total surface stress τ:

$$z_e = \frac{z_o}{\sqrt{1 - \tau_w / \tau}} \quad \text{and} \quad z_o = \hat{\alpha} \frac{U_*^2}{g} \quad . \qquad (14)$$

The second of these two equations is a Charnock-like relation in which $\hat{\alpha}$ is a constant equal to 0.01. The wave stress $\underline{\tau}_w$ vector is given by:

$$\underline{\tau}_w = \rho_w \int_0^{2\pi} \int_0^{\infty} \sigma B\, E(\sigma, \theta) \frac{\underline{k}}{k} d\sigma d\theta \quad . \qquad (15)$$

The value of U_* can be determined for a given wind speed, U_{10}, and a given wave spectrum, E(σ, θ), from the above set of equations. The iterative procedure of Mastenbroek et al. (1993) is used in SWAN. This set of expressions (Eq. 11-15) is also used in WAM Cycle 4 (Komen et al., 1994).

5.1.4.4 Dissipation of Wave Energy (S_{ds})

5.1.4.4.1 Whitecapping

The pulse-based model of Hasselmann (1974) represents the processes of whitecapping in the SWAN model. Reformulated in terms of wave number rather than frequency to be applicable in finite water depth (cf. the WAMDI group, 1988), this expression is found in **Section 5.1.1.3** under equation (3). The steepness dependent coefficient (Γ), as given by the WAMDI group (1988), has been adapted by Günther et al. (1992) based on Janssen (1991a, see Janssen, 1991b):

$$\Gamma = \Gamma_{KJ} = C_{ds}\left((1-\delta) + \delta\frac{k}{\tilde{k}}\right)\left(\frac{\tilde{s}}{\tilde{s}_{PM}}\right)^{p} \quad , \qquad (16)$$

For $\delta = 0$ the expression of Γ reduces to the expression as used by the WAMDI group (1988). The coefficients C_{ds}, δ and m are tunable coefficients, $\tilde{s}$ is the overall wave steepness (defined below), $\tilde{s}_{PM}$ is the value of $\tilde{s}$ for the Pierson-Moskowitz spectrum (1964; $\tilde{s}_{PM}$ = $(3.02 \times 10^{-3})^{\frac{1}{2}}$). This overall wave steepness $\tilde{s}$ is defined as:

$$\tilde{s} = \tilde{k}\sqrt{E_{tot}} \quad . \qquad (17)$$

The mean frequency $\tilde{\sigma}$, the mean wave number $\tilde{k}$, and the total wave energy E_{tot} are defined as (cf. the WAMDI group, 1988):

$$\tilde{\sigma} = \left(E_{tot}^{-1}\int_0^{2\pi}\int_0^{\infty}\frac{1}{\sigma}E(\sigma,\theta)\,\mathrm{d}\sigma\,\mathrm{d}\theta\right)^{-1}$$

$$, \qquad (18)$$

$$\tilde{k} = \left(E_{tot}^{-1}\int_0^{2\pi}\int_0^{\infty}\frac{1}{\sqrt{k}}E(\sigma,\theta)\,\mathrm{d}\sigma\,\mathrm{d}\theta\right)^{-2}$$

$$E_{tot} = \int_0^{2\pi}\int_0^{\infty}E(\sigma,\theta)\,\mathrm{d}\sigma\,\mathrm{d}\theta \quad . \qquad (19)$$

The values of the tunable coefficients C_{ds} and δ and exponent p in this model have been obtained by Komen et al. (1984) and Janssen (1992) by closing the energy balance of the waves in idealized wave growth conditions (both for growing and fully developed wind seas) for deep water. This implies that coefficients in the steepness dependent coefficient Γ depend on the wind-input formulation used. Since two different wind input formulations are used in the SWAN model, two sets of coefficients are used. For the wind input of Komen et al. (1984): $C_{ds} = 2.36\times10^{-5}$, $\delta = 0$ and $p = 4$. Janssen (1992) and Günther (1992) obtained (assuming $p = 4$) $C_{ds} = 4.10\times10^{-5}$ and $\delta = 0.5$ (as used in the WAM Cycle 4; Komen et al., 1994).

5.1.4.4.2 Bottom Friction

The bottom friction models that have been selected for SWAN are the empirical model of JONSWAP (Hasselmann et al., 1973), the drag law model of Collins (1972) and the eddy-viscosity model of Madsen et al. (1988). The formulations for these bottom friction models can all be expressed in the form of equation (4) found in **Section 5.1.1.3**. C_{bottom} is a bottom friction coefficient that generally depends on the bottom orbital motion represented by U_{rms}:

$$U_{rms}^2 = \int_0^{2\pi}\int_0^{\infty} \frac{\sigma^2}{\sinh^2(kd)} E(\sigma,\theta)\,\mathrm{d}\sigma\,\mathrm{d}\theta \quad . \tag{20}$$

Hasselmann et al. (1973) found from the results of the JONSWAP experiment $C_{bottom} = C_{JON} = 0.038$ m^{2}s^{-3} for swell conditions. Bouws and Komen (1983) selected a bottom friction coefficient of $C_{JON} = 0.067$ m^{2}s^{-3} for fully developed wave conditions in shallow water. Both values are available in SWAN.

The expression of Collins (1972) is based on a conventional formulation for periodic waves with the appropriate parameters adapted to suit a random wave field. The dissipation rate is calculated with the conventional bottom friction formulation of Eq. 2, in which the bottom friction coefficient is $C_{bottom} = C_f g U_{rms}$ with $C_f = 0.015$. Note that Collins (1972) contains an error in the expression due to an erroneous Jacobean transformation; see page A-16 of Tolman, 1990.

Madsen et al. (1988) derived a formulation similar to that of Hasselmann and Collins (1968), but in their model the bottom friction factor is a function of the bottom roughness height and the actual wave conditions. Their bottom friction coefficient is given by:

$$C_{bottom} = f_w \frac{g}{\sqrt{2}} U_{rms} \quad , \tag{21}$$

in which f_w is a non-dimensional friction factor estimated by using the formulation of Jonsson (1966; cf. Madsen et al., 1988):

$$\frac{1}{4\sqrt{f_w}} + \log_{10}\left[\frac{1}{4\sqrt{f_w}}\right] = m_f + \log_{10}\left[\frac{a_b}{K_N}\right] \quad , \tag{22}$$

in which $m_f = -\,0.08$ (Jonsson and Carlsen, 1976) and a_b is a representative near-bottom excursion amplitude:

$$a_b^2 = 2\int_0^{2\pi}\int_0^{\infty} \frac{1}{\sinh^2(kd)} E(\sigma,\theta)\,\mathrm{d}\sigma\,\mathrm{d}\theta \quad , \tag{23}$$

and K_N is the bottom roughness length scale. For values of a_b / K_N smaller than 1.57 the friction factor f_w is 0.30 (Jonsson, 1980).

5.1.4.4.3 Depth-induced Wave Breaking

To model the energy dissipation in random waves due to depth-induced breaking, SWAN uses the bore-based model of Battjes and Janssen (1978). The mean rate of energy dissipation per unit horizontal area due to wave breaking D_{tot} is expressed as:

$$D_{tot} = -\frac{1}{4}\alpha_{BJ} Q_b \left(\frac{\bar{\sigma}}{2\pi}\right) H_m^2 \quad , \tag{24}$$

in which $\alpha_{BJ} = 1$ in SWAN, Q_b is the fraction of breaking waves determined by:

$$\frac{1-Q_b}{\ln Q_b} = -8\frac{E_{tot}}{H_m^2} \quad , \tag{25}$$

in which H_m is the maximum wave height that can exist at the given depth and $\bar{\sigma}$ is a mean frequency defined as:

$$\bar{\sigma} = E_{tot}^{-1} \int_0^{2\pi}\int_0^{\infty} \sigma E(\sigma,\theta) \mathrm{d}\sigma \mathrm{d}\theta \quad . \tag{26}$$

Extending the expression of Eldeberky and Battjes (1995) to include the spectral directions, the dissipation for a spectral component per unit time is calculated with equation (5) in **Section 5.1.1.3**.

The maximum wave height, H_m, is determined with $H_m = \gamma d$, in which γ is the breaker parameter and d is the total water depth including the wave-induced set-up if computed by SWAN. In the literature, this breaker parameter γ is often a constant or it is expressed as a function of bottom slope or incident wave steepness (see Galvin, 1972; Battjes and Janssen, 1978; Battjes and Stive, 1985; Arcilla and Lemos, 1990; Kaminsky and Kraus, 1993; Nelson, 1987, 1994). Since SWAN is locally defined, the dependency on incident wave steepness cannot be used. Instead, the constant value or bottom-slope dependent options are used in all SWAN versions to determine the value of the breaker parameter. In SWAN Version 40.11, the option of Nelson has been removed, as the results of SWAN are better with the option of a constant value.

In the dissipation model description of Battjes and Janssen (1978) a constant breaker parameter of $\gamma = 0.8$ was used based on Miche's criterion. Battjes and Stive (1985) re-analyzed laboratory and field wave data and found values for the breaker parameter varying between 0.6 and 0.83 for varying bathymetry (plane, bar-trough and bar) with an average of 0.73. Kaminsky and Kraus (1993) found breaker parameters in the range of 0.6 to 1.59 with an average of 0.79.

5.1.4.5 Nonlinear Wave-wave Interactions (S_{nl})

5.1.4.5.1 Quadruplet Wave-wave Interactions

The quadruplet wave-wave interactions are computed with the Discrete Interaction Approximation (DIA) as proposed by Hasselmann et al. (1985). Their source code (slightly adapted by Tolman, personal communication, 1993) has been used in the SWAN model. In DIA two quadruplets of wave numbers are considered, both with frequencies:

$$\begin{aligned} \sigma_1 &= \sigma_2 = \sigma \\ \sigma_3 &= \sigma(1+\lambda) = \sigma^+ \quad , \\ \sigma_4 &= \sigma(1-\lambda) = \sigma^- \end{aligned} \tag{27}$$

where λ is a constant coefficient set equal to 0.25. To satisfy the resonance conditions for the first quadruplet, the wave number vectors with frequency σ_3 and σ_4 lie at an angle of θ_1 = - 11.5° and θ_2 = 33.6° to the two identical wave number vectors with frequencies σ_1 and σ_2. The second quadruplet is the mirror of this first quadruplet (the wave number vectors with frequency σ_3 and σ_4 lie at mirror angles of θ_3 = 11.5° and θ_4 = - 33.6°).

Within DIA, the source term S_{nl4} (σ, θ) is given by:

$$S_{nl4}(\sigma,\theta) = S^*_{nl4}(\sigma,\theta) + S^{**}_{nl4}(\sigma,\theta) \quad , \tag{28}$$

where S^*_{nl4} refers to the first quadruplet and S^{**}_{nl4} to the second quadruplet (the expressions for S^{**}_{nl4} are identical to those for S^*_{nl4} for the mirror directions) and

$$S^*_{nl4}(\sigma,\theta) = 2\delta S_{nl4}(\alpha_1\sigma,\theta) - \delta S_{nl4}(\alpha_2\sigma,\theta) - \delta S_{nl4}(\alpha_3\sigma,\theta) \quad , \tag{29}$$

where $\alpha_1 = 1$, $\alpha_2 = (1 + \lambda)$ and $\alpha_3 = (1 - \lambda)$. Each of the contributions (i = 1, 2, 3) is:

$$\begin{aligned} \delta S_{nl4}(\alpha_i\sigma,\theta) = C_{nl4}(2\pi)^2 g^{-4}\left(\frac{\sigma}{2\pi}\right)^{11} \Bigg[& E^2(\alpha_i\sigma,\theta)\left(\frac{E(\alpha_i\sigma^+,\theta)}{(1+\lambda)^4} + \frac{E(\alpha_i\sigma^-,\theta)}{(1-\lambda)^4}\right) \\ & -2\frac{E(\alpha_i\sigma,\theta)E(\alpha_i\sigma^+,\theta)E(\alpha_i\sigma^-,\theta)}{(1-\lambda^2)^4}\Bigg] \quad . \end{aligned} \tag{30}$$

The constant $C_{nl4} = 3\times10^7$. Following Hasselmann and Hasselmann (1981), the quadruplet interaction in finite water depth is taken identical to the quadruplet transfer in deep water multiplied with a scaling factor R:

$$S_{nl4,\,finitedepth} = R(k_p d)\, S_{nl4,infinitedepth} \quad , \tag{31}$$

where R is given by:

$$R(k_p d) = 1 + \frac{C_{shl}}{k_p d}\left(1 - C_{sh2} \cdot k_p d\right)\exp\left(C_{sh3} \cdot k_p d\right) \quad , \tag{32}$$

in which k_p is the peak wave number of the JONSWAP spectrum for which the original computations were carried out. The values of the coefficients are $C_{shl} = 5.5$, $C_{sh2} = 6/7$ and $C_{sh3} = -\ 1.25$. In the shallow water limit, i.e., $k_p d \rightarrow 0$, the nonlinear transfer tends towards infinity. Therefore a lower limit of $k_p d = 0.5$ is applied, resulting in a maximum value of $R(k_p d) = 4.43$. To increase the model robustness in case of arbitrarily shaped spectra, the peak wave number k_p is replaced by $k_p = 0.75\tilde{k}$ (Komen et al., 1994).

5.1.4.5.2 Triad Wave-wave Interactions

The Lumped Triad Approximation (LTA) of Eldeberky (1996), a slightly adapted version of the Discrete Triad Approximation of Eldeberky and Battjes (1995), is used in SWAN in each spectral direction:

$$S_{nl3}(\sigma,\theta) = S^-_{nl3}(\sigma,\theta) + S^+_{nl3}(\sigma,\theta) \tag{33}$$

with

$$S^+_{nl3}(\sigma,\theta) = \max\left\{0, \alpha_{EB}\, 2\pi c c_g\, J^2 \left|\sin(\beta)\right| \left\{E^2(\sigma/2,\theta) - 2E(\sigma/2,\theta)\, E(\sigma,\theta)\right\}\right\} \tag{34}$$

and $S^-_{nl3}(\sigma,\theta) = -2S^+_{nl3}(2\sigma,\theta)$ (35)

in which α_{EB} is a tunable proportionality coefficient. The biphase β is approximated with

$$\beta = -\frac{\pi}{2} + \frac{\pi}{2}\tanh\left(\frac{0.2}{Ur}\right) \tag{36}$$

with Ursell number Ur

$$Ur = \frac{g}{8\sqrt{2}\pi^2}\frac{H_s \bar{T}^2}{d^2} \tag{37}$$

with $\bar{T} = 2\pi/\bar{\sigma}$. The triad wave-wave interactions are calculated only for $10 > Ur > 0.1$. The interaction coefficient J is taken from Madsen and Sørensen (1993):

$$J = \frac{k_{\sigma/2}^2 (g\,d + 2c_{\sigma/2}^2)}{k_\sigma d(g\,d + \frac{2}{15} g\,d^3 k_\sigma^2 - \frac{2}{5}\sigma^2 d^2} \tag{38}$$

5.1.4.5.3 *Wave-induced Set-up*

In a geographic 1-D case the wave induced set-up computation is based on the vertically integrated momentum balance equation. It is a balance between the wave force (gradient of the wave radiation stress normal to the coast) and the hydrodynamic pressure gradient (note that the component parallel to the coast causes wave-induced currents but no setup):

$$F_x + g\,d\frac{\partial \overline{\eta}}{\partial x} = 0, \tag{39}$$

where d is the total water depth and η is the mean surface elevation. Both d and η include wave-induced set-up.

Computations based on the vertically integrated momentum balance equation of Dingemans et al. (1987) show that the wave-induced currents are driven by the divergence-free portion of the wave forces, whereas the set-up is mainly due to the rotation-free part of these forces. To compute the set-up, it is sufficient to consider the divergence of the momentum balance equation. If the divergence of the acceleration in the resulting equation is ignored, the result is:

$$\frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial}{\partial x}\left(g\,d\frac{\partial \zeta}{\partial x}\right) + \frac{\partial}{\partial y}\left(g\,d\frac{\partial \zeta}{\partial y}\right) = 0 \quad . \tag{40}$$

5.2 Numerical Implementation

The integration of the action balance equation has been implemented in SWAN with finite difference schemes in all dimensions including time, geographic space and spectral space. These are first described for wave propagation without source terms for generation, dissipation and wave-wave interactions. Then the implementation of these source terms is described.

Time is discretized with a simple constant time step, Δt, for the simultaneous integration of the propagation and the source terms. In WAM and WAVEWATCH the propagation time step is different from the time step for the source terms. Geographic space is discretized with a rectangular grid with constant resolutions Δx and Δy in x- and y-direction, respectively. In fact, this rectangular grid is a special case of the curvilinear grid programmed in SWAN. The spectrum in the model is discretized with a constant directional resolution $\Delta\theta$ and a constant relative frequency resolution $\Delta\sigma/\sigma$ (logarithmic frequency distribution). For reasons of economy, an option is available to compute only wave components traveling in a pre-defined directional sector ($\theta_{\min} < \theta < \theta_{\max}$; e.g., those components that travel shoreward within a limited directional sector). The discrete frequencies are defined between a fixed low-frequency cut-off and a fixed high-frequency cut-off. For these frequencies the spectral density is unconstrained. Below the low-frequency cut-off, which is typically f_{min} = 0.04 Hz for field conditions, the spectral densities are assumed to be zero. Above the high-frequency cut-off, typically 1 Hz for field conditions, a diagnostic tail f^{-m} is added. This tail is used to compute nonlinear wave-wave interactions at the high frequencies and compute integral wave parameters. The reason for using a fixed high-frequency rather than a dynamic cut-off frequency that depends on the wind speed or on the mean frequency, as in WAM and WAVEWATCH III, is that in coastal regions mixed sea states with different characteristic frequencies may occur. For instance, a local wind may generate a very young sea behind an island, totally unrelated to (but superimposed on) a simultaneously occurring swell. In such cases a dynamic cut-off frequency may be too low to properly account for the locally generated sea state. Based on physical arguments the value of m (the power of the spectral tail) should be between 4 and 5 (e.g., Phillips, 1985). In SWAN $m = 4$ if the wind input formulation of Komen et al. (1984) is used (cf. WAM Cycle 3), and $m = 5$ if the wind input formulation of Janssen (1991a) is used (WAM Cycle 4).

5.2.1 Propagation

The numerical schemes in SWAN have been chosen on the basis of robustness, accuracy and economy. Since the nature of the basic equation is such that the state in a grid point is determined by the state in the upwave grid points, the most robust scheme would be an implicit upwind scheme, in both geographic and spectral space. The term "implicit" is used to indicate that all derivatives of action density (in t, x or y) are formulated at one computational level, i_t or i_x or i_y, except for the integration dimension derivative for which the previous or upwave level is used (time in non-stationary mode and x or y in stationary mode). For such a scheme the values of the time and space steps Δt, Δx, and Δy are mutually independent. An implicit scheme would also be economical because the scheme is unconditionally stable. It permits relatively large time steps in the computations, much larger than for explicit schemes in shallow water. Using the second-generation HISWA shallow water wave model, Holthuijsen et al., 1989 showed that for coastal regions a first-order upwind difference scheme in geographic space is sufficiently accurate. This observation, together with SWAN test runs, has also shown that in spectral space a higher accuracy than that of a first-order upwind scheme is required. This can be achieved by supplementing with a second-order central approximation, which is more economic than a second-order upwind scheme. For SWAN therefore, implicit upwind schemes in both geographic and spectral space have been chosen, supplemented with a central approximation in spectral space.

The state in a grid point is determined by the state in the upwave grid points as defined by the direction of propagation. This permits a decomposition of the spectral space into four quadrants (eight octants would be an alternative). In each of the quadrants the computations may be carried out independently except for the interactions between them due to refraction and nonlinear wave-wave interactions (formulated in corresponding boundary conditions between the quadrants). The wave components in SWAN are correspondingly propagated in geographic space with an upwind scheme ("upwind" is the commonly used term in numerical analysis, but "upwave" would be more appropriate in the case of SWAN). SWAN contains three such schemes:

a. First-order (stationary and non-stationary cases) backward space-backward time: the BSBT scheme,
b. Second-order (non-stationary cases) with third-order diffusion: the S&L-scheme (Stelling and Leedertse, 1992),
c. Second-order (stationary cases) with second-order diffusion: the SORDUP scheme.

The first-order upwind BSBT scheme is not default in SWAN. It is a sequence of four forward-marching sweeps (one per quadrant). To properly account for the boundary conditions between the four quadrants, the computations are carried out iteratively at each time step. The integration in time is a simple backward finite difference, so that the discretization of the action balance equation is as follows for positive propagation speeds including the computation of the source terms but ignoring their discretization:

$$\begin{aligned}
&\left[\frac{N^{i_t,n}-N^{i_t-1}}{\Delta t}\right]_{i_x,i_y,i_\sigma,i_\theta} + \left[\frac{[c_xN]_{i_x} - [c_xN]_{i_x-1}}{\Delta x}\right]_{i_y,i_\sigma,i_\theta}^{i_t,n} + \left[\frac{[c_yN]_{i_y} - [c_yN]_{i_y-1}}{\Delta y}\right]_{i_x,i_\sigma,i_\theta}^{i_t,n} \\
&+ \left[\frac{(1-\nu)[c_\sigma N]_{i_\sigma+1} + 2\nu[c_\sigma N]_{i_\sigma} - (1+\nu)[c_\sigma N]_{i_\sigma-1}}{2\Delta\sigma}\right]_{i_x,i_y,i_\theta}^{i_t,n} + \\
&\left[\frac{(1-\eta)[c_\theta N]_{i_\theta+1} + 2\eta[c_\theta N]_{i_\theta} - (1+\eta)[c_\theta N]_{i_\theta-1}}{2\Delta\theta}\right]_{i_x,i_y,i_\sigma}^{i_t,n} = \left[\frac{S}{\sigma}\right]_{i_x,i_y,i_\sigma,i_\theta}^{i_t,n^*}
\end{aligned} \tag{41}$$

where i_t is the time-level index, i_x, i_y, i_σ and i_θ are grid counters, and Δt, Δx, Δy, $\Delta\sigma$, and $\Delta\theta$ are the increments in time, geographic space and spectral space, respectively. The iterative nature of the computation is indicated with the iteration index n (the iteration index for the source terms n^* is equal to n or n-1, depending on the source term. See below). Because of these iterations, the scheme is also approximately implicit for the source terms. For negative propagation speeds, appropriate + and - signs are required in Eq. 41.

The coefficients ν and η determine the degree to which the scheme in spectral space is upwind or central. They control the numerical diffusion in frequency and directional space, respectively. A value of $\nu = 0$ or $\eta = 0$ corresponds to central schemes which have the largest accuracy (numerical diffusion ≈ 0). Values of $\nu = 1$ or $\eta = 1$ correspond to upwind schemes which are somewhat more robust and diffusive and therefore less accurate. If large gradients of the action density in frequency space or directional space are present, numerical oscillations may arise, especially with the central difference schemes, resulting in negative action density values. In each sweep such negative values are removed from the two-dimensional spectrum by setting these values equal to zero and re-scaling the remaining positive values such that the frequency-integrated action density per spectral direction is conserved. The depth derivatives and current derivatives in the expressions of c_σ and c_θ are calculated with a first-order upwind scheme. For very strong refraction the value of c_θ is reduced in each grid point and each wave component individually with the square of the fraction of the grid spacing over which $kd < 3.0$.

For stationary conditions SWAN may be run in stationary mode. Time is removed as a variable but the integration in geographic space is still carried out iteratively. The propagation scheme is still implicit as the derivatives of action density (in x or y) at the computational level (i_x or i_y, respectively) are formulated at that level except in the integration dimension, (x or y; depending on the direction of propagation) where the upwave level is used. The values of Δx and Δy are therefore still mutually independent.

For the stationary second-order upwind scheme (Rogers et al., 2000) which is the default scheme for **stationary** computations, the two terms in Eq. 41 representing x- and y-derivatives are replaced by:

$$\left[\frac{1.5[c_xN]_{i_x} - 2[c_xN]_{i_x-1} + 0.5[c_xN]_{i_x-2}}{\Delta x}\right]_{i_y,i_\sigma,i_\theta}^{i_t,n} + \left[\frac{1.5[c_yN]_{i_y} - 2[c_yN]_{i_y-1} + 0.5[c_yN]_{i_y-2}}{\Delta y}\right]_{i_x,i_\sigma,i_\theta}^{i_t,n} \qquad (41a)$$

For the nonstationary second-order upwind scheme (Rogers et al., 2000; S and L = Stelling and Leendertse), which is the default scheme for **nonstationary** computations, the two terms in Eq. 41 representing x- and y-derivatives are replaced by:

$$\left[\frac{\frac{5}{6}[c_xN]_{i_x} - \frac{5}{4}[c_xN]_{i_x-1} + \frac{1}{2}[c_xN]_{i_x-2} - \frac{1}{12}[c_xN]_{i_x-3}}{\Delta x}\right]_{i_y,i_\sigma,i_\theta}^{i_t,n} +$$

$$\left[\frac{\frac{5}{6}[c_yN]_{i_y} - \frac{5}{4}[c_yN]_{i_y-1} + \frac{1}{2}[c_yN]_{i_y-2} - \frac{1}{12}[c_yN]_{i_y-3}}{\Delta y}\right]_{i_x,i_\sigma,i_\theta}^{i_t,n} + \qquad (41b)$$

$$\left[\frac{\frac{1}{4}[c_xN]_{i_x+1} - \frac{1}{4}[c_xN]_{i_x-1}}{\Delta x}\right]_{i_y,i_\sigma,i_\theta}^{i_t-1} + \left[\frac{\frac{1}{4}[c_yN]_{i_y+1} - \frac{1}{4}[c_yN]_{i_y-1}}{\Delta y}\right]_{i_x,i_\sigma,i_\theta}^{i_t-1}$$

To explain the above numerical solution technique in terms of matrix solutions, first ignore the decomposition in quadrants. The propagation of the waves in both geographic and spectral space would then be described with one large basic matrix that can be solved in several ways. Removing refraction, frequency shifting and nonlinear source terms from this basic matrix permits a matrix solution with a Gauss-Seidel technique (e.g., Golub and van Loan, 1986) in which the matrix is decomposed in four sections (the above four directional quadrants) that are each solved in one step (super-convergence). Restoring refraction and frequency shifting to the matrix requires the solution of a submatrix for each geographic grid point. If no currents are present and the depth is stationary, this is readily done with a Thomas algorithm (e.g., Abbott and Basco, 1989; $c_\sigma = 0$ and the submatrix is a simple tri-diagonal matrix). If currents are present or the depth is not stationary, the submatrix is a band matrix. It is solved with an iterative ILU-CGSTAB method (Vuik, 1993; Van der Vorst, 1992). Restoring refraction and frequency-shifting also introduces coefficients in each matrix section (directional quadrant) that cause dependency between the sections. The same happens when nonlinear source terms are added to the matrix. The basic matrix as a whole must be solved iteratively until some break-off criteria are met. To reduce the number of iterations in stationary mode with wind generation, SWAN begins with a reasonable first-guess of the wave field, a "quickstart" based on the second-generation source terms of Holthuijsen and De Boer, 1988, adapted for shallow water. It reduces the number of iterations typically by a factor two. In nonstationary mode, a very reasonable first-guess per time step is available from the previous time step and the number of iterations is expected to be small. If no iterations are used in nonstationary mode (as in most phase averaged wave models), the computations of propagation are still implicit and therefore still unconditionally stable.

For grid points that represent open boundaries, land boundaries and obstacles, these being the last two grids adjoining such grid points for the SORDUP scheme and the last three grids adjoining such grid points for the S&L scheme, SWAN will revert to the first-order BSBT scheme. This scheme has a larger numerical diffusion that is usually acceptable over the small distances involved.

The numerical diffusion of the S&L scheme is so small that the so-called garden-sprinkler effect (GSE) may appear if propagation over very large distances is considered. GSE is due to spectral resolution (see Booij and Holthuijsen, 1987) and can be counteracted by a diffusion term explicitly added to the numerical scheme (not default in SWAN). Its value depends on the spectral resolution and the propagation time of the waves (see the input variable [wave age] in the SCHEME command).

The diffusion applied in the propagation direction is:

$$D_{ss} = \Delta c^2 T / 12, \qquad (42)$$

where T is the wave age.

The diffusion normal to the propagation direction is:

$$D_{ss} = c^2 \Delta\theta^2 T / 12. \qquad (43)$$

From these diffusion coefficients (in terms of x and y) are calculated:

$$D_{xx} = D_{ss}\cos^2\theta + D_{nn}\sin^2\theta;$$

$$D_{yy} = D_{ss}\sin^2\theta + D_{nn}\cos^2\theta;$$
$$D_{xy} = (D_{ss} - D_{nn})\cos\theta\sin\theta. \qquad (44)$$

The diffusion terms are computed at the time level i_t - 1. The diffusion terms are computed as follows:

$$D_{xx}\left[\frac{[N]_{i_x+1} - 2[N]_{i_x} + [N]_{i_x-1}}{\Delta x^2}\right]_{i_y,i_\sigma,i_\theta}^{i_t-1}$$
$$D_{yy}\left[\frac{[N]_{i_y+1} - 2[N]_{i_y} + [N]_{i_y-1}}{\Delta y^2}\right]_{i_x,i_\sigma,i_\theta}^{i_t-1} \qquad (45)$$
$$D_{xy}\left[\frac{[N]_{i_x,i_y} - [N]_{i_x-1,i_y} - [N]_{i_x,i_y-1} + [N]_{i_x-1,i_y-1}}{\Delta x\Delta y}\right]_{i_\sigma,i_\theta}^{i_t-1}$$

This explicit finite differencing is fast (having little impact on computation time) but only conditionally stable. Through mathematical analysis (not shown) it can be shown that a likely stability condition for the one-dimensional S&L scheme with a GSE correction is $D\Delta t/(\Delta x^2) \leq 0.5$, corresponding to the two-dimensional stability criterion of Tolman (1995):

$$Q = \frac{\max(D_{xx}, D_{yy}, D_{xy})\Delta t}{\min(\Delta x\Delta y)^2} \leq 0.5 \qquad (46)$$

Eq. 46 holds true for the two-dimensional S and L scheme with a GSE correction. In experiments, instability was observed for experiments that satisfied the slightly more restrictive $Q \leq 0.48$. In short, by adding the GSE correction, the unconditionally stable advection scheme of SWAN becomes a (likely) conditionally stable advection diffusion scheme. For typical ocean applications D_{nn} dominates the diffusion and can be written as:

$$Q = \overline{C}T / \Delta x.\overline{C}\Delta t / \Delta x.\Delta\theta^2 / 12. \qquad (47)$$

The variable wave age $\overline{T}$ could be computed during SWAN computations (Booij and Holthuijsen, 1987) but requires the same order of magnitude of computer memory as integrating the action balance equation. Instead, a constant wave age $\overline{T}$ may be used as an approximation so that Eq. 47 becomes

$$Q = \overline{L} / \Delta x.\mu.\Delta\theta / 12, \qquad (48)$$

where the characteristic travel distance of the waves is $\overline{L} = \overline{C}\overline{T}$ (e.g., the dimension of the ocean basin). For oceanic applications the Courant number is typically $\mu \approx$ ½ so that $Q \leq 0.25$ for typical values of $\Delta\theta$ and $\overline{L} / \Delta x$ (the number of grid points in one direction of the grid). This implies that an S&L scheme with GSE correction is stable for typical ocean cases. For shelf sea (regional) applications the value of $\mu = O(1)$ but GSE tends to be small on these scales and the diffusion must not be used to avoid the stability problem. For small-scale (local) applications $\mu = O(10\text{-}100)$. Such cases are usually treated as stationary, and the SORDUP scheme should be used (no GSE correction is included in this scheme).

The boundary conditions in SWAN, both in geographic and spectral space, are fully absorbing for wave energy leaving the computational domain or crossing a coastline. The incoming wave energy along open geographic boundaries must be prescribed by the user. For coastal regions incoming energy is provided only along the deep-water boundary and not along the lateral geographic boundaries (i.e., the spectral densities are assumed to be zero). This implies that such erroneous lateral boundary conditions are propagated into the computational area. The affected areas are typically triangular regions with the apex at the corners between the deep-water boundary and the lateral boundaries. The areas spread towards shore at an angle of 30° to 45° for wind sea conditions on either side of the deep-water mean wave direction. The angle is less for swell conditions and is essentially equal to the one-sided width of the directional distribution of the incoming wave spectrum. Therefore, the lateral boundaries should be sufficiently far away from the area of interest to avoid the propagation of this error into the area.

5.2.1.1 Generation, Wave-wave Interactions and Dissipation

The numerical estimations of the source terms in SWAN are essentially implicit. This is achieved with explicit or implicit approximations of the source terms, which in the limit of a large number of iterations, always result in an implicit estimation. In actual computations final convergence is never achieved, and the estimations of the source terms are only approximately implicit. In the following description, the adjectives "explicit" and "implicit" refer to the approximations of the source terms within each iteration.

The linear growth term A is independent of integral wave parameters and the energy density and can therefore be readily computed. All other source terms depend on energy density and can be described as (quasi-) linear terms: $S = \phi E$, in which ϕ is a coefficient that depends on (integral) wave parameters such as E_{tot}, $\tilde{\sigma}$, $\tilde{k}$, σ, k, and action densities of other spectral components. Since these are only known at the previous iteration level n-1, the coefficient is determined at that iteration level: $\phi = \phi^{n-1}$.

For positive source terms (wind input and the triad wave-wave interactions if positive) the integration is generally more stable if an explicit formulation is used (i.e., the source term depends on E^{n-1} and not on E^n) rather than an implicit formulation (i.e., the source term also depends on E^n). The explicit formulation for these source terms in SWAN is therefore:

$$S^n \approx \phi^{n-1} E^{n-1} \tag{49}$$

The explicit approximation is also used for the formulation of the quadruplet wave-wave interactions for both the positive and negative contributions. Tolman (1992a) has shown that using an explicit formulation in combination with a limiter (see below) gives results similar to using a more expensive implicit scheme. The implicit formulation is optionally available in SWAN and is indicated as the semi-implicit scheme in WAM (The WAMDI group, 1988; Komen et al, 1994).

For negative source terms the integration is more stable if an implicit scheme is used. The strongly nonlinear, negative source term of depth-induced wave breaking at iteration level *n* is accordingly estimated with a linear approximation:

$$S^n \cong \phi^{n-1}\, E^{n-1} + \left(\frac{\partial S}{\partial E}\right)^{n-1} (E^n - E^{n-1})\,. \tag{50}$$

However, to achieve even more stable computations for this source term, the term $\phi^{n-1}E^{n-1}$ in this formulation has been replaced by $\phi^{n-1}E^n$, thus making the formulation somewhat more implicit and robust. Note that the solution in the limit is the same. Since the process of depth-induced wave breaking has been formulated such that $S = aS_{tot}$ and $E = aE_{tot}$, the derivative is $\partial S / \partial E$ analytically determined as $\partial S_{tot} / \partial E_{tot}$ (where is *a* identical in both expressions and the total energy E_{tot} and the total source S_{tot} are the integrals over all frequencies and directions of $E(\sigma, \theta)$ and $S_{ds,br}(\sigma, \theta)$, respectively). For other negative (mildly nonlinear) source terms such as whitecapping, bottom friction and negative triad wave-wave interactions, a similar accuracy of estimating S^n can be achieved with the following simpler more economical approximation in which $(\partial S / \partial E)^{n-1}$ of Eq. 13 has been replaced by $(S/E)^{n-1}$

$$S^n \cong \phi^{n-1}\, E^{n-1} + \left(\frac{S}{E}\right)^{n-1} (E^n - E^{n-1}) \quad . \tag{51}$$

With $S = \phi E$, this reduces to:

$$S^n \cong \phi^{n-1}\, E^n \quad . \tag{52}$$

The approximations for the source terms are added to the matrix elements for the propagation. To suppress the development of numerical instabilities, the maximum total change of action density per iteration at each discrete wave component is limited to a fraction of 10% of the Phillips (1957) equilibrium level. It is reformulated in terms of action density and wave number to be applicable in shallow water, as in WAM and WAVEWATCH III (Tolman, 1992a):

$$\left|\Delta N(\sigma,\theta)\right|_{\max} = \frac{0.1}{2\pi\sigma}\frac{\alpha_{PM}\pi}{k^3 c}\,, \tag{53}$$

where $\alpha_{PM} = 0.0081$ is the Phillips' "constant" of the Pierson-Moskowitz (1964) spectrum. To retain the very rapid but realistic decrease of wave energy near the shore due to depth-induced wave breaking, this limiter is not applied if the waves actually break. In SWAN, $H_{rms}/H_{\max} < 0.2$ with $H_{rms} = \sqrt{8E_{tot}}$, which implies a fraction of breakers $Q_b > 0.00001$.

The fraction of depth-induced breakers (Q_b) is determined in SWAN with

$$Q_b = 0 \qquad \text{for} \quad \beta \leq 0.2$$

$$Q_b = Q_0 - \beta^2 \frac{Q_0 - \exp((Q_0 - 1)/\beta^2)}{\beta^2 - \exp((Q_0 - 1)/\beta^2)} \qquad \text{for} \quad 0.2 < \beta < 1$$

(one step Newton-Raphson iteration) (54)

$$Q_b = 1 \qquad \text{for} \quad \beta = 1$$

where $\beta = H_{rms}/H_{\max}$. For $\beta \leq 0.5$, $Q_0 = 1$, and for $0.5 < \beta \leq 1$, $Q_0 = (2\beta - 1)^2$.

5.2.1.2 Wave-induced Set-up

In 1-D cases the wave-induced set-up is calculated with a simple trapezoidal rule.

In 2-D cases the Poisson equation of the divergence-free force field is solved using the same solver employed for wave propagation with ambient currents. The boundary conditions for the elliptical partial differential equation are:

Non-nested computations:

- At open boundaries, the equilibrium between wave force and hydrodynamic pressure gradient is normal to the model boundary;
- At the last grid points before shoreline, the equilibrium between wave force and hydrodynamic pressure gradient is normal to the model boundary;
- At deepest boundary point, the set-up is zero.

Nested computations:

- At open boundaries, the set-up is taken from the larger computation,
- At last grid points before shoreline, the equilibrium between wave force and hydrodynamic pressure gradient is normal to the model boundary.

The shoreline in SWAN moves as dictated by the wave-induced set-up. The set-up computations are available on both the rectilinear and curvilinear grids.

5.2.1.3 Curvilinear Grid

SWAN's propagation scheme for geographic space is formulated on a curvilinear geographic grid (irregular, quadrangular, and not necessarily orthogonal) rather than the rectilinear grid of SWAN Cycle 1. This modification is based on approximating the geographic distribution of the energy action density between each three neighboring grid points with a flat triangle. The gradient in each grid point at location (x_i, y_j) is then readily approximated from the upwind grid points. For the *x*-

direction this approximation is for grid point *i, j* (the grid points are ordered in *x*, *y*-space with labels *i* and *j*, respectively):

$$\frac{\partial C_x N}{\partial x} \cong \left[\frac{[c_x N]_{i,j} - [c_x N]_{i-1,j}}{\Delta\tilde{x}_1}\right] + \left[\frac{[c_x N]_{i,j} - [c_x N]_{i,j-1}}{\Delta\tilde{x}_2}\right] \tag{55}$$

where $\Delta\tilde{x}_1 = \Delta x_1 - (\Delta y_1 / \Delta y_2)\Delta x_2$, $\Delta\tilde{x}_2 = \Delta x_2 - (\Delta y_2 / \Delta y_1)\Delta x_1$. The increments are $\Delta x_1 = x_{i,j} - x_{i-1}$, $\Delta x_2 = x_{i,j} - x_{i,j-1}$, $\Delta y_1 = y_{i,j} - y_{i-1,j}$ and $\Delta y_2 = y_{i,j} - y_{i,j-1}$. The gradient in y-direction is similarly estimated.

6.0 Technical References

6.1 SWAN Software Release

Booij, N., Haagsma, IJ. G., Kieftenburg, A. T. M. M., and Holthuijsen, L. H. (2000). "SWAN Cycle II Version 40.11 Implementation Manual, Electronic Version." Delft University of Technology, the Netherlands.

Holthuijsen, L. H., Booij, N., Ris, R. C., Haagsma, IJ. G., Kieftenburg, A. T. M. M., and Kriezi, E. (2000). "SWAN Cycle III Version 40.11 User Manual, Electronic Version." Delft University of Technology, the Netherlands.

6.2 General Technical Documentation

Abbott, M. B. and Basco, D. R., (1989). Computational Fluid Dynamics. John Wiley and Sons Inc., New York, p. 425.

Abreu, M., Larraza, A., and Thornton, E., (1992). Nonlinear transformation of directional wave spectra in shallow water. *J. Geophys. Res.*, 97: 15579-15589.

Arcilla, A. S., Roelvink, J. A., O'Connor, B. A., Reniers, A. J. H. M., and Jimenez, J. A., (1994). The Delta flume '93 experiment. *In the Proc. of the Coastal Dynamics Conf.*, 488-502.

Arcilla, A. S. and Lemos, C. M., (1990). Surf-Zone Hydrodynamics. *Centro Internacional de Métodos Numéricos en Ingenieria*, Barcelona, Spain, 310 pp.

Banner, M. L. and Young, I. R., (1994). Modeling spectral dissipation in the evolution of wind waves, Part I: Assessment of existing model performance. *J. Phys. Oceanogr.*, 24(7): 1550-1571.

Battjes, J. A., (1994). Shallow water wave modeling, M. Isaacson and M. Quick, eds., In: *Proc. Waves-Phys. and Num. Modeling*, University of British Columbia, Vancouver, Canada, 1-24 pp.

Battjes, J. A. and Beji, S., (1992). Breaking waves propagating over a shoal. *In the Proc. of the 23rd Int. Conf. Coastal Eng.*, ASCE, 42-50.

Battjes, J. A. and Stive, M. J. F., (1985). Calibration and verification of a dissipation model for random breaking waves. *J. Geophys. Res.*, 90(C5): 9159-9167.

Battjes, J. A. and Van Vledder, C. Ph., (1984). Verification of Kimura's theory for wave group statistics. *In the Proc. of the 19th Int. Conf. Coastal Eng.*, ASCE, Houston, Texas, 642-648.

Battjes, J. A. and Janssen, J. P. F. M., (1978). Energy loss and set-up due to breaking of random waves. *In the Proc. of the 16th Int. Conf. Coastal Eng.*, ASCE, 569-587.

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

Bertotti, L. and Cavaleri, L., (1994). Accuracy of wind and wave evaluation in coastal regions. *In the Proc. of the 24th Int. Conf. Coastal Eng.*, ASCE, 57-67.

Booij, N., Ris, R. C., and Holthuijsen, L. H., (1999). A third-generation wave model for coastal regions, Part I: Model description and validation. *J. Geoph. Res.*, 104(C4): 7649-7666.

Booij, N., Holthuijsen, L. H., and Haagsma, IJ. G., (1998). Comparing the second-generation HISWA wave model with the third-generation SWAN wave model. *In the Proc. of the 5th Int. Workshop on Wave Hindcasting and Forecasting*, Melbourne, Florida, 215-222.

Booij, N., Holthuijsen, L. H., and Ris, R. C., (1998). Shallow water wave modeling. *In: Proc. of the Oceanology Int., The Global Ocean, Brighton, Conf.*, 3, 483-491.

Booij, N., Holthuijsen, L. H., Doorn, N., and Kieftenburg, A. T. M. M., (1997). Diffraction in a spectral wave model. *In the Proc. of the 3rd Int. Sym. on Ocean Wave Meas. and Analysis*, WAVES '97, ASCE, 243-255.

Booij, N., Holthuijsen, L. H., and Padilla-Hernandez, R., (1997). A nonstationary, parametric coastal wave model. *In: Proc. of the Conf. on Coastal Dynamics*, Plymouth, MA 99-107.

Booij, N., Holthuijsen, L. H., and Padilla-Hernandez, R., (1997). Numerical wave propagation on a curvilinear grid. *In the Proc. of the 3rd Int. Sym. on Ocean Wave Meas. and Analysis*, WAVES '97, ASCE, 286-294.

Booij, N., Holthuijsen, L. H., and Ris, R. C., (1996). The "SWAN" wave model for shallow water. *In the Proc. of the 25th Int. Conf. Coastal Eng.*, Orlando, Florida, 668-676.

Booij, N., Holthuijsen, L. H., and De Lange, P. H. M., (1992). The penetration of short-crested waves through a gap. *In the Proc. of the 23rd Int. Conf. Coastal Eng.*, Venice, New York, 1993, 1044-1052.

Booij, N. and Holthuijsen, L. H., (1987). Propagation of ocean waves in discrete spectral wave models. *J. of Comp. Phys.*, 68: 307-326.

Bouws, E. and Komen, G. J., (1983). On the balance between growth and dissipation in an extreme, depth-limited wind-sea in the southern North Sea. *J. Phys. Oceanogr.*, 13: 1653-1658.

Cavaleri, L. and Malanotte-Rizzoli, P., (1981). Wind wave prediction in shallow water: Theory and applications. *J. Geophys. Res.*, 86(C11): 10961-10973.

Chen, Y. and Guza, R. T., (1997). Modeling of breaking surface waves in shallow water. *J. Geophys. Res.*, 102(C11): 25035-25046.

Collins, J. I., (1972). Prediction of shallow water spectra, *J. Geophys. Res.*, 77(15): 2693-2707.

Dingemans, M. W., (1997). Water wave propagation over uneven bottoms, Part 1: Linear wave propagation. *Adv. Ser. Ocean Eng.*, 13, World Scientific, 471 pp.

Dingemans, M. W., Radder, A. C., and De Vriend, H. J., (1978). Computations of the driving forces of wave induced currents, *Coastal Eng.*, 11: 539-563.

Eldeberky, Y., (1996). Nonlinear transformation of wave spectra in the nearshore zone, *Ph.D. thesis*. Delft University of Technology, Department of Civil Engineering, the Netherlands.

Eldeberky, Y. and Battjes, J. A., (1996). Spectral modeling of wave breaking: Application to Boussinesq equations. *J. Geophys. Res.*, 101(C1): 1253-1264.

Eldeberky, Y. and Battjes, J. A., (1995). Parameterization of triad interactions in wave energy models. *In the Proc. of the Coastal Dynamics Conf.*, Gdansk, Poland, 140-148.

Galvin, C. J., (1972). Wave Breaking in Shallow Water, Waves on Beaches and Resulting Sediment Transport. Academic Press Inc., San Diego, California, pp. 413-455.

Goda, Y, Takeda, H., and Moriya, Y., (1967). Laboratory investigation of wave transmission over breakwaters. *Rep. Port and Harbour Res. Inst.*, 13 (from Seelig, 1979).

Golub, G. H. and Van Loan, C. F., (1986). Matrix Computations. Academic Press, London, p. 476.

Günther, H., Hasselmann, S., and Janssen, P. A. E. M. (1992). "The WAM model Cycle 4 (revised version)." Deutsch. Klim. Rechenzentrum, Technical Report No. 4, Hamburg, Germany.

Hasselmann, S., Hasselmann, K., Allender, J. H., and Barnett, T. P., (1985). Computations and parameterizations of the nonlinear energy transfer in a gravity wave spectrum, Part II: Parameterizations of the nonlinear transfer for application in wave models. *J. Phys. Oceanogr.*, 15(11): 1378-1391.

Hasselmann, S. and Hasselmann, K., (1981). A symmetrical method of computing the non-linear transfer in a gravity-wave spectrum. *Geophys. Einzelschr., Ser. A*, Geophys. Inst., Univ. of Hamburg, Hamburg, Germany, 52(8).

Hasselmann, K., Ross, D. B., Müller, P., and Sell, W., (1976). A parametric wave prediction model. *J. Phys. Oceanogr.*, 6: 200-228.

Hasselmann, K., (1974). On the spectral dissipation of ocean waves due to whitecapping. *Bound.-layer Meteor.*, 6: 1-2, 107-127.

Hasselmann, K., Barnett, T. P., Bouws, E., Carlson, H., Cartwright, D. E., Enke, K., Ewing, J. A., Gienapp, H., Hasselmann, D. E., Kruseman, P., Meerburg, A., Müller, P., Olbers, D. J., Richter, K., Sell, W., and Walden, H., (1973). Measurements of wind-wave growth and swell decay during the JOint North Sea WAve Project (JONSWAP). *Dtsch. Hydrogr. Z. Suppl.*, 12(A8): 1-95.

Hasselmann, K. and Collins, J. I., (1968). Spectral dissipation of finite-depth gravity waves due to turbulent bottom friction. *J. Mar. Res.*, 26: 1-12.

Hasselmann, K., (1960). Grundgleichungen der Seegangsvoraussage. *Schiffstechnik*, 1: 191-195.

Holthuijsen, L. H. and Booij, N., (2000). Oceanic and near-shore whitecapping effects in SWAN. *In the Proc. of the 6th Int. Workshop on Wave Hindcasting and Forecasting,* Monterey, California, In press.

Holthuijsen, L. H, Ris, R. C., Booij, N., and Cecchi, E., (2000). Swell and whitecapping, a numerical experiment. *In the Proc. of the 7th Int. Conf. Coastal Eng.*, Sydney, Australia, In press.

Holthuijsen, L. H., Ris, R. C., and Booij, N., (1998). A verification of the third-generation wave model SWAN. *In the Proc. of the 5th Int. Workshop on Wave Hindcasting and Forecasting*, Melbourne, Florida, 223-230.

Holthuijsen, L. H., Booij, N., and Haagsma, IJ. G., (1998). Comparing 1st -, 2nd - and 3rd – generation coastal wave modeling. *In the Proc. of the 26th Int. Conf. Coastal Eng.*, Copenhagen, Denmark, 140-149.

Holthuijsen, L. H., Booij, N., and Padilla-Hernandez, R., (1997). A curvilinear, third-generation coastal wave model. *In the Proc. of the Conf. Coastal Dynamics '97*, Plymouth, MA 128-136.

Holthuijsen, L. H., Booij, N., Ris, R., Andorka Gal, J. H., and De Jong, J. C. M., (1997). A verification of the third-generation wave model "SWAN" along the southern North Sea coast. *In: Proc. of the 3rd Int. Sym. on Ocean Wave Measurement and Analysis*, WAVES '97, ASCE, 49-63.

Holthuijsen, L. H., Booij, N., and Ris, R. C., (1993). A spectral wave model for the coastal zone. *In: Proc. of the 2nd Int. Sym. on Ocean Wave Meas. and Analysis*, New Orleans, LA, 630-641.

Holthuijsen, L. H., Booij, N., and Herbers, T. H. C., (1989). A prediction model for stationary, short-crested waves in shallow water with ambient currents. *Coastal Eng.*, 13: 23-54.

Holthuijsen, L. H. and De Boer, S., (1988). "Wave Forecasting for Moving and Stationary Targets." Computer Modeling in Ocean Engineering. B.Y. Schrefler and O.C. Zienkiewicz, eds., Balkema Publishing Co., Rotterdam, the Netherlands, pp. 231-234.

Holthuijsen, L. H., (1980). Methods of wave prediction, Part I and II (Methoden voor golfvoorspelling, deel I en II, in Dutch), Technical Advisory Commission Against Inundation (Technische Adviescommissie voor de Waterkeringen, in Dutch), Den Haag, the Netherlands.

Janssen, P. A. E. M., (1991a). Quasi-linear theory of wind-wave generation applied to wave forecasting. *J. Phys. Oceanogr.*, 21: 1631-1642.

Janssen, P. A. E. M., (1991b). Consequences of the effect of surface gravity waves on the mean air flow, paper presented at the *Breaking Waves. Int. Union of Theor. and Appl. Mech. (IUTAM*), Sydney, Australia, 193-198.

Janssen, P. A. E. M., (1989). Wave induced stress and the drag of air flow over sea waves. *J. Phys. Oceanogr.*, 19: 745-754.

Jonsson, I. G., (1993). Wave current interactions, in The Sea, *Ocean Eng. Sci. Ser.*, Vol. 9, Part A, B. Le Mehaute and D.M. Hanes, eds., John Wiley, New York, 65-70 pp.

Jonsson, I. G., (1980). A new approach to rough turbulent boundary layers. *Ocean Eng.*, 7: 109-152.

Jonsson, I. G. and Carlsen, N. A., (1976). Experimental and theoretical investigations in an oscillatory turbulent boundary layer. *J. Hydraulic Res.*, 14: 45-60.

Jonsson, I. G. (1966). Wave boundary layers and friction factors. *In the Proc. of the 10th Int. Conf. Coastal Eng.*, ASCE, New York, 127-148.

Kahma, K. K. and Calkoen, C. J., (1992). Reconciling discrepancies in the observed growth of wind-generated waves. *J. Phys. Oceanogr.*, 22: 1389-1405.

Kaminsky, G. M. and Kraus, N. C., (1993). Evaluation of depth-limited wave breaking criteria. *In the Proc. of the 2nd Int. Symposium on Ocean Wave Meas. and Analysis*, ASCE, New Orleans, Louisiana, 180-193.

Komen, G. J., Cavaleri, L., Donelan, M., Hasselmann, K., Hasselmann, S., and Janssen, P. A. E. M., (1994). Dynamics and Modeling of Ocean Waves. Cambridge University Press, New York, p. 532.

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

Kuik, A. J., Van Vledder, C. Ph., and Holthuijsen, L. H., (1988). A method for the routine analysis of pitch and-roll buoy wave data. *J. Phys. Oceanogr.*, 18: 1020-1034.

Luo, W. and Monbaliu, J., (1994). Effects of the bottom friction formulation on the energy balance for gravity waves in shallow water. *J. Geophys. Res.*, 99(C9): 18501-18511.

Madsen, O. S., Poon, Y.–K., and Graber, H. C., (1988). Spectral wave attenuation by bottom friction: Theory. *In the Proc.. of the 21st Int. Conf. Coastal Eng.*, ASCE, New York, 492-504.

Madsen, P. A. and Sørensen, O. R., (1993). Bound waves and triad interactions in shallow water. *Ocean Eng.,* 20(4): 359-388.

Madsen, P. A. and Sørensen, O. R., (1992). A new form of the Boussinesq equations with improved linear dispersion characteristics, Part 2: A slowly varying bathymetry. *Coastal Eng.*, 18: 183-205.

Mase, H. and Kirby, J. T., (1992). Hybrid frequency-domain KdV equation for random wave transformation. *In the Proc. of the 23rd Int. Conf. Coastal Eng.*, ASCE, New York, 474-487.

Mastenbroek, C., Burgers, G., and Janssen, P. A. E. M., (1993). The dynamical coupling of a wave model in a storm surge model through the atmospheric boundary layer. *J. Phys. Oceanogr.*, 23: 1856-1866.

Mei, C. C., (1983). The Applied Dynamics of Ocean Surface Waves. John Wiley and Sons Inc, New York, p. 740.

Miles, J. W., (1981). Hamiltonian formulations for surface waves. *Appl. Sci. Res.*, 37: 103-110.

Nelson, R. C., (1994). Depth limited wave heights in very flat regions. *Coastal Eng.*, 23: 43-59.

Nelson, R. C., (1987). Design wave heights on very mild slopes: An experimental study. *Civil. Eng. Trans.*, 29: 157-161.

Padilla-Hernandez, R., Osuna, P., Monbaliu, J., and Holthuijsen, L., (1998). Intercomparing third-generation wave model nesting. *In the Proc. of the 5th Int. Workshop on Wave Hindcasting and Forecasting*, Melbourne, Florida, 102-112.

Phillips, O. M., (1985). Spectral and statistical properties of the equilibrium range in wind-generated gravity waves. *J. Fluid Mech.*, 156: 505-531.

Phillips, O. M., (1957). On the generation of waves by turbulent wind. *J. Fluid Mech.*, 2: 417-445.

Pierson, W. J. and Moskowitz, L., (1964). A proposed spectral form for fully developed wind seas based on the similarity theory of S. A. Kitaigorodskii. *J. Geophys. Res.*, 69(24): 5181-5190.

Putnam, J. A. and Johnson, J. W., (1949). The dissipation of wave energy by bottom friction. *Trans. Am. Geoph. Union.*, 30: 67-74.

Ris, R. C., Booij, N., and Holthuijsen, L. H., (1999). A third-generation wave model for coastal regions, Part II: Verification. *J. Geoph. Res.*, 104(C4): 7667-7681.

Ris, R. C. and Holthuijsen, L. H., (1997). Modeling of current induced wave blocking in a spectral wave model. *In the Proc. of the 8th Int. Biennial Conf. on Physics of Estuaries and Coastal Seas.* J. Dronkers and M.B.A.M. Scheffers (eds.), The Hague, Netherlands, pp. 139-144.

Ris, R. C., Holthuijsen, L. H., and Booij, N., (1994). A spectral model for waves in the near shore zone. *In the Proc. of the 24th Int. Conf. Coastal Eng.*, Kobe, Japan, 68-78.

Rogers, W. E., Kaihatu, J. M., Petit, H. A. H., Booij, N., and Holthuijsen, L. H., (2000). Arbitrary-Scale Propagation in a Third Generation Wind Wave Model, manuscript.

Seelig, W. N., (1979). Effects of breakwaters on waves: laboratory tests of wave transmission by overtopping. *In: Proc. of the Conf. Coastal Structures*, 79(2): 941-961.

Shemdin, P., Hasselmann, K., Hsiao, S. V., and Herterich, K., (1978). Non-linear and linear bottom interaction effects in shallow water, in: Turbulent Fluxes through the Sea Surface, *Wave Dynamics and Prediction, NATO Conf. Ser.*, V(1), 347-372pp.

Snyder, R. L., Dobson, F. W., Elliott, J. A., and Long, R. B., (1981). Array measurement of atmospheric pressure fluctuations above surface gravity waves. *J. Fluid Mech.*, 102: 1-59.

Stelling, G. S. and Leendertse, J. J., (1992). Approximation of convective processes by cyclic AOI methods. *In: Proc. of the 2nd Int. Conf. on Estuarine and Coastal Modeling*, ASCE, Tampa, Florida, 771-782.

Thornton, E. B. and Guza, R. T., (1983). Transformation of wave height distribution. *J. Geophys. Res.*, 88(C10): 5925-5938.

Tolman, H. L., (1995). "On the selection of propagation schemes for a spectral wind-wave model." *NWS/NCEP Office Note 411*, 30 pp. + figures.

Tolman, H. J., (1992a). Effects of numerics on the physics in a third-generation wind-wave model. *J.Phys. Oceanogr.*, 22(10): 1095-1111.

Tolman, H. L., (1992b). An evaluation of expressions for the wave energy dissipation due to bottom friction in the presence of currents. *Coastal Eng.*, 16: 165-179.

Tolman, H. L., (1990). Wind wave propagation in tidal seas, *Ph.D. thesis*. Delft University of Technology, Department of Civil Engineering, the Netherlands.

Van der Vorst, H. A., (1992). Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for solution of non-symmetric linear systems. *SIAM J. Sci. Statistical Computing*, 13: 631-644.

Vincent, C. L., Smith, J. M., and Davis, J., (1994). Parameterization of wave breaking in models, Vol. II. *In: Proc. of the Int. Symp.: Waves - Physical and Numerical Modeling*, M. Isaacson and M. Quick, eds., Univ. of British Columbia, Vancouver, Canada, 753-762 pp.

Vuik, C., (1993). Solution of the discretized incompressible Navier-Stokes equations with the GMRES method. *Int. J. for Num. Meth. in Fluids*, 16: 507-523.

WAMDI Group, (1988). The WAM model - a third generation, ocean wave prediction model. *J. Phys. Oceanogr.*, 18: 1775-1810.

Weber, S. L., (1991a). Bottom friction for wind sea and swell in extreme depth-limited situations. *J. Phys. Oceanogr.*, 21: 149-172.

Weber, S. L., (1991b). Eddy-viscosity and drag-law models for random ocean wave dissipation. *J. Fluid Mech.*, 232: 73-98.

Weber, S. L., (1989). Surface gravity waves and turbulent bottom friction, *Ph.D. thesis*. University of Utrecht, The Netherlands.

Whitham, G. B., (1974). Linear and Nonlinear Waves. John Wiley and Sons Inc., New York, p. 636.

Wu, J., (1982). Wind-stress coefficients over sea surface from breeze to hurricane. *J. Geophys. Res.*, 87(C12): 9704-9706.

Young, I. R. and Van Vledder, G. Ph., (1993). A review of the central role of nonlinear interactions in wind-wave evolution. *Phil. Trans. R. Soc. London, Ser. A.*, 342: 505-524.

Young, I. R. and Banner, M. L., (1992). Numerical experiments on the evolution of fetch limited waves, paper presented at *Int. Union of Theor. and Appl. Mech. (IUTAM)*, Sydney, Australia, 267-275.

Appendix A Command syntax

It is important distinguish between the description of the commands in this manual and the actual commands in the command file. The descriptions of the commands are called command schemes. Each such command scheme includes a diagram and a description explaining the structure of the command and the meaning of the keyword(s) and data in the command. The actual SWAN user commands must be given in one file containing all commands. This file is called the command file. It must be presented to SWAN in ASCII format. The proper sequence of the commands is given in **Section 4.12.1**.

Command Keywords

Each command instructs SWAN to carry out a certain action, which SWAN executes before it reads the next command. A command must always begin with a keyword (which is also the name of the command) that indicates the primary function of that command; see list in **Section 4.12.1**. A simple command may appear in its command scheme as:

```
KEYWORD         data
```

A command may contain more than one keyword (which refines the instructions to SWAN), e.g.,

```
KEY1WORD    KEY2WORD    data
```

where KEY2WORD is the second key word.

Spelling of keywords

In every command scheme, keywords appear as words in capital letters with the first part underlined (KEY1WORD and KEY2WORD in the above example). When typing the command in the command file, the user must copy the underlined part. SWAN reads only the underlined part and SWAN is case insensitive except for one instance (character strings), see below. When typing the keyword in the command file, any extension of the underlined part is at the user's discretion as long as the extension is limited to letters (upper- and lower case) or digits (but not underlined), as well as the characters - and _. So in the command outlined above one may write: KEY or KEYW or KEY-word or KEYHOLE etc. at the users discretion.

Example:

```
KEY1WORD    KEY2WORD    data                        (command scheme):

Key1     KeY2    data                               (command file)
```

In the command file, a keyword is closed by a blank or by = or : . Square brackets or quotes do not enclose it. A keyword followed by a comma (,) is interpreted as a keyword followed by an empty data field (see below).

Required and optional keywords

All keywords in a command are required except when an option is available. Optional keywords are indicated in the command scheme with the following signs enclosing the keywords concerned:

```
       | KEY1WORD .......data.............. |
      <                                      >
       | KEY2WORD .......data.............. |
```

For the above example it may appear as:

```
               | KEY2WORD data   |
KEY1WORD      <                   >
               | KEY3WORD data   |
```

In case the user does not indicate an option in a command, SWAN chooses the alternative indicated with an arrow (->) appearing in the command scheme (the default option). In the above example it may appear as:

```
               |    KEY2WORD data |
KEY1WORD      <                    >
               | -> KEY3WORD data |
```

where `KEY3WORD` is the default option.

Repetitions of keywords and/or other data

The use of keywords is sometimes repetitive, such as in a sequence of data and keywords containing many locations in x-y space. In this case, the command scheme indicates this repetitive nature by placing the keywords (and data) concerned between angle brackets < >. In the above example it may appear as:

```
                           | KEY2WORD |
KEY1WORD          < data <             > data >
                           | KEY3WORD |
```

The user must give such a sequence in the actual command of the command file. It ends with either the end of the line, a keyword other than one mentioned in the repetition group, or the character / or ; . If more than one line is required for a command, the user may continue on the next line as described below ('end of line'). The repetition may consist of one instance (in fact, no repetition at all).

Data

Most commands contain data, either character data or numerical data. Character data (strings) are represented in the command schemes by names, enclosed in quotes (' '). Numerical data are represented by names enclosed in square brackets ([ ]). As a rule an error message will result if numerical data is given where character data should be.

Spelling of character data

Character data are represented as character strings (a sequence of characters and blanks) between quotes in the command scheme and in the command file. SWAN interprets an end of the line as an end quote. Therefore, a character data field can never extend over more than one line. In a command scheme the character string is always a name that is placed between quotes. In the command file such a name can be entered in two ways:

1. Replace the name by another character string at the user's discretion (this is the only occurrence where SWAN is case sensitive; e.g., for text to appear in a plot).

   Example:

   ```
   KEY1WORD KEY2WORD 'City' data                (command scheme)

   KEY1 KEY2 'Amsterdam-01a' data               (command file)
   ```

2. Copy the name of the variable (without the quotes) followed by an = sign and a name at the user's discretion (between quotes). SWAN interprets the copied name in the command file as a keyword, as if it were fully underlined in the command scheme with all the characteristics of a keyword, such as ending a sequence of optional data (see below). As with other keywords, the name of the variable is case-insensitive.

   Example:

   ```
   KEY1WORD KEY2WORD 'City' data                (command scheme)

   KEY1WORD KEY2WORD City='Amsterdam-01' data   (command file)
   ```

An error message will result if numerical data is given where character data should be given.

Spelling of numerical data

Numerical data are simple numbers, e.g., 15 or -7 (integer data), or 13.7 or 0.8E-4 (real data). Whether or not an integer or a real number should be given by the user is indicated in the description of the command scheme. A decimal point is not permitted in an integer number. On the other hand, SWAN accepts an integer number where a real number should be given. In a command scheme the number is always indicated with a name (which is placed between square brackets). In the command file, such a name can be entered in two ways:

1. Replace the name by a number (not between square brackets).

   Example:

   ```
   KEY1WORD KEY2WORD [nnn]                    (command scheme)

   KEY1 KEY2 307                              (command file)
   ```

2. Copy the name of the variable (without the square brackets) followed by an = sign and the number (not between square brackets). SWAN interprets the copied name in the command file as a keyword (as if it were fully underlined in the command scheme) with all the characteristics of a keyword such as ending a sequence of optional data (see below). As with other keywords, the name of the variable is case-insensitive.

   Example:

   ```
   KEY1WORD KEY2WORD [nnn]                    (command scheme)

   KEY1WORD KEY2WORD nnn=307                  (command file)
   ```

Required data and optional data

All data, given by the user, must be given in the command file in the same order as they appear in the command scheme. Blanks or commas separate them. Required data (indicated in the description of each individual command) must be given explicitly as character strings or numbers.

Optional data are indicated in the text of each individual command or by parenthesis around the data concerned. For example,

```
KEY1WORD KEY2WORD 'name' ([nnn] [mmm]) [zzz]
```

Some optional data are indicated in the same way as optional keywords are indicated:

```
 | .......data..............      |
<                                  >
 | .......data..............      |
```

Optional data may be omitted by giving blanks between commas, and SWAN will substitute reasonable default values. If all data after a required datum is optional (till the next keyword or the next end-of-line), then commas may be omitted as well. Optional data indicated like optional keywords are to be treated as optional keywords.

Command file and comments

All text after one $ or between two $ signs on one line in the command file is ignored by SWAN as comment text. Such comments may be important to the user, e.g., to clarify the meaning of the commands used. In fact, this option has been used to create the edit file SWAN.EDT (see Appendix C). Any text appearing after two $ signs is not interpreted as a comment but as data to be processed (possibly interrupted again by $ or two $ signs).

End of line or continuation

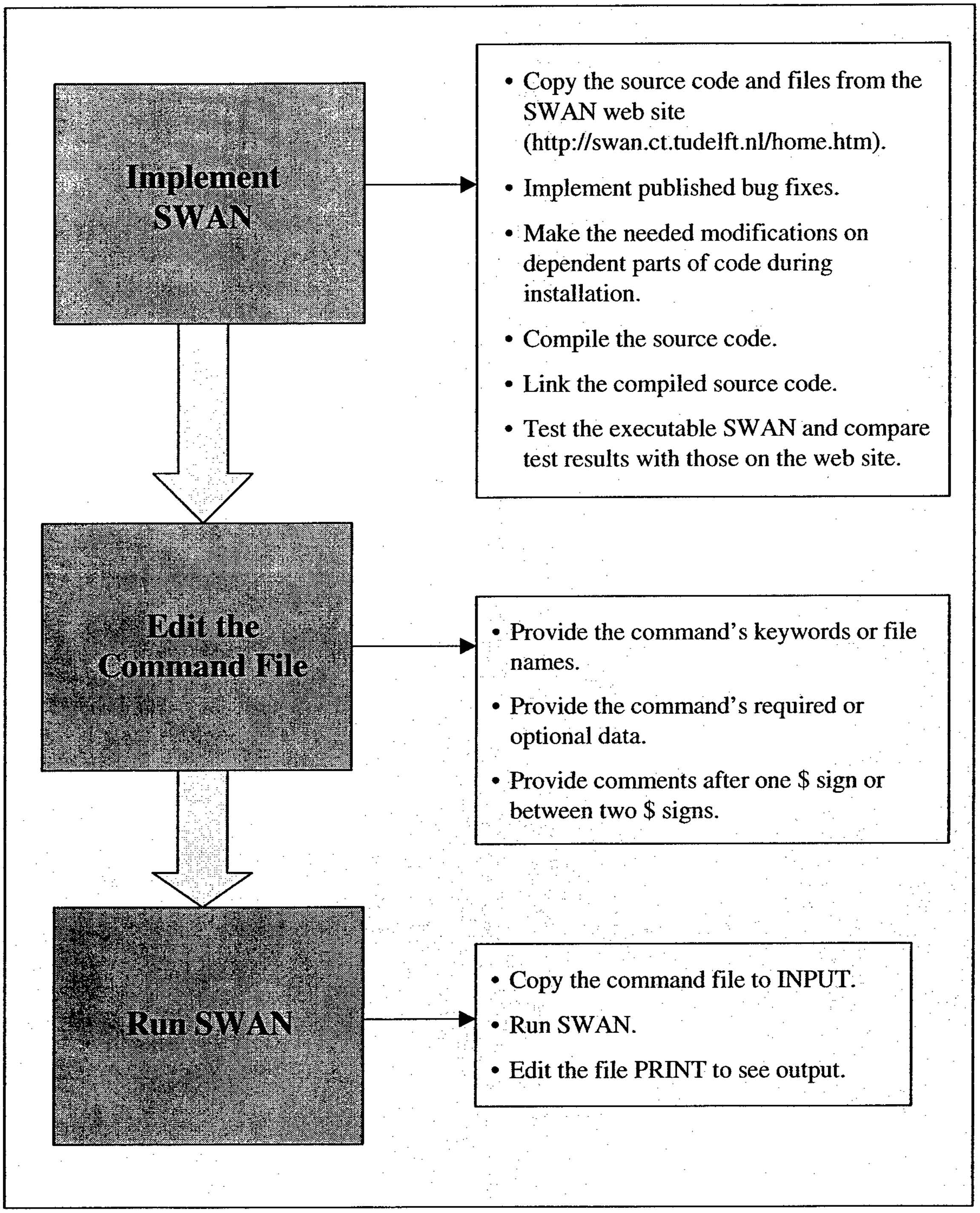
A command in the command file may be continued on the next line if the previous line terminates with a continuation mark

& or _ (underscore)

Appendix B Run instructions

To run SWAN, e.g., under DOS, UNIX or MS-WINDOWS, talk to the person who implemented SWAN on your computer system. SWAN is to be implemented by following the instructions in **Section 4.0** or by using the implementation manual found on the SWAN web site. A summary of the run instructions is illustrated in the following figure.

Concept of Execution



- Copy the source code and files from the SWAN web site (http://swan.ct.tudelft.nl/home.htm).
- Implement published bug fixes.
- Make the needed modifications on dependent parts of code during installation.
- Compile the source code.
- Link the compiled source code.
- Test the executable SWAN and compare test results with those on the web site.

- Provide the command's keywords or file names.
- Provide the command's required or optional data.
- Provide comments after one $ sign or between two $ signs.

- Copy the command file to INPUT.
- Run SWAN.
- Edit the file PRINT to see output.

Figure B-1. Flow diagram summarizing the steps for execution of SWAN Version 40.11.

Appendix C File SWAN.EDT

```
$   PROJect  'name'  'nr'
$            'title1'
$            'title2'
$            'title3'
$
$   MODE    STATionary/NONSTationary     TWODimensional/ONEDimensional
$
$   COORDinates /  -> CARTesian         \
$               \  SPHErical   CCM/QC   /
$
$   SET  [level] [nor] [depmin] [maxmes] [maxerr]      &
$        [grav] [rho] [inrhog] [hsrerr]                &
$        CARTesian/NAUTical  [pwtail] [froudmax]       &
$        [printf] [prtest]
$
$   CGRID  / REGular [xpc] [ypc] [alpc] [xlenc] [ylenc] [mxc] [myc]  \
$          \ CURVilinear [mxc] [myc] ( EXCeption [xexc] [yexc] )     /     &
$          / CIRcle                   \
$          \ SECtor [dir1] [dir2] [mdc] /                                  &
$            [flow] [fhig] [msc]
$
$   TEST [itest] [itrace]                              &
$        POINTS  IJ < [i] [j] > / XY < [x] [y] >       &
$        (PAR 'fname')  (S1D 'fname')  (S2D 'fname')
$
$   INPgrid   BOTtom/WLEVel/CURrent/VX/VY/FRiction/WInd/WX/WY         &
$      / REGular [xpinp] [ypinp] [alpinp] [mxinp] [myinp] [dxinp] [dyinp] \
$      \ CURVilinear [stagrx] [stagry] [mxinp] [myinp]                    /  &
$      (EXCeption [excval] )                                                 &
$      (NONSTATionary [tbeginp] [deltinp] Sec/MIn/HR/DAy [tendinp])
$
$   READ   BOTtom/WLevel/CURrent/FRiction/WInd/COORdinates                 &
$        [fac] 'fname1'/SERIES 'fname2' [idla] [nhedf] ([nhedt]) (nhedvec]) &
$         / FREe                    \
$        <  FORmat  'form'/[idfm]    >
$         \ UNFormatted             /
$
$   WIND  [vel] [dir]
$
$   BOUNdpar1 SHAPespec  JON [gamma] / PM / GAUSS [sigfr] / BIN        &
$        PEAK/MEAN    DSPR POWER/DEGRees
$
$   BOUNdpar2  / SIDE  N/NW/W/SW/S/SE/E/NE   CCW/CLOCKWise \
$              \ SEGMent  XY < [x] [y] > / IJ < [i] [j] >  /        &
$       / CONSTANT PAR [hs] [per] [dir] [dd]             \
$       \ VARIABLE PAR < [len] [hs] [per] [dir] [dd] >   /
$
$   BOUNdspec  / SIDE  N/NW/W/SW/S/SE/E/NE   CCW/CLOCKWise \
$              \ SEGMent  XY < [x] [y] > / IJ < [i] [j] >  /        &
$       / CONSTANT FILE 'fname' [seq]           \
$       \ VARIABLE FILE < [len] 'fname' [seq] >  /
$
$   BOUNdnest1  NEst 'fname'  CLOSed/OPEN
$
$   BOUNdnest2  WAMNest 'fname' / UNFormatted CRAY/WKstat \
$                               \ FREe                    /      &
$         [xgc] [ygc]
```

```
$
$   BOUNdnest3  WW3  'fname'  CLOSed/OPEN  ([xgc] [ygc])
$
$             /  DEFault
$   INITial  <   ZERO
$             \  PAR [hs] [per] [dir] [dd]
$              \ HOTStart 'fname'
$
$   GEN1  [cf10] [cf20] [cf30] [cf40] [edmlpm] [cdrag] [umin] [cfpm]
$
$   GEN2  [cf10] [cf20] [cf30] [cf40] [cf50] [cf60]     &
$         [edmlpm] [cdrag] [umin] [cfpm]
$
$   GEN3  / KOMen [cds2] [stpm]     \
$         \ JANSsen  [cds1] [delta] /                              &
$         (QUADrupl [iquad] [lambda] [cnl4] [csh1] [csh2] [csh3])  &
$         (AGROW [a])
$
$   BREaking  CONstant [alpha] [gamma]
$
$              / JONswap [cfjon]
$   FRICtion  <  COLLins [cfw]
$              \ MADsen [kn]
$
$   TRIad   [trfac] [cutfr]
$
$   OBSTacle   (TRANSm [trcoef] / DAM [hgt] [alpha] [beta])             &
$              (REFL  [reflc])    LINE  < [xp] [yp] >
$
$   SETUP  [supcor]
$
$   OFF   BREaking / WCAPping / REFrac / FSHift / QUADrupl / WINDGrowth
$
$   PROP  ??  T / SORDup \  NONST  / STELling [waveage] Sec/MIn/HR/DAy \
$               \ BSBT   /         \ BSBT                              /
$
$   NUMeric   ( ACCUR [drel] [dhabs] [dtabs] [npnts]               &
$                / -> STATionary [mxitst]  \
$                \ NONSTationary [mxitns]  /      [limiter] )      &
$             (DIRimpl [cdd] [cdlim]                        )      &
$             (SIGIMpl [css] [prec] [eps1] [eps2] [outp] [niter])  &
$             (SETUP rpec] [eps2] [outp] [niter])
$
$   FRAme   'sname'  [xpfr] [ypfr] [alpfr] [xlenfr] [ylenfr]       &
$           [mxfr] [myfr] [scale]
$
$   GROUP   'sname'  SUBGrid [ix1] [ix2] [iy1] [iy2]
$
$   NGRid   'sname'  [xpn] [ypn] [alpn] [xlenn] [ylenn] [mxn] [myn]
$
$   CURve   'sname'  [xp1] [yp1]   < [int]  [xp]  [yp] >
$
$   POINTs  'sname'    < [xp]  [yp] > / FILE 'fname'
$
$   RAY     'rname'  [xp1] [yp1] [xq1] [yq1]        &
$          <  [int]  [xp]  [yp]  [xq]  [yq]  >
$
$   ISOline 'sname'  'rname'  DEPth/BOTtom  [dep]
$
$   LINe     CONTinuous / DAShed [pat] [len]           &
$            (COLor [ipen])  < [xp]  [yp] >
```

```
$
$   SITEs    < 'pname'  [xp]  [yp]  [size]  REGion / TOWN >
$
$   QUANTity  DSPR/HSign/DIR/PDIR/TDIR/TM01/RTM01/RTP/TM02/FSPR/PER/RPER/
$             DEPth/VEL/FRCoeff/WIND/DISSip/QB/TRAnsp/FORce/UBOT/URMS/WLEN/
$             STEEpness/HSWEll/DHSign/DRTM01/LEAK/XP/YP/DIST                &
$          'short' 'long' [lexp] [hexp] [excv]                               &
$          ([power])  ([ref])   ([fswell])                                   &
$          (PROBLEMcoord/FRAME)
$
$   BLOck   'sname'  HEADer / NOHEADer  'fname' (LAYout [idla])              &
$        < DSPR/HSign/DIR/PDIR/TDIR/TM01/RTM01/RTP/TM02/FSPR/PER/RPER/
$          DEPth/VEL/FRCoeff/WIND/DISSip/QB/TRAnsp/FORce/UBOT/URMS/WLEN/
$          STEEpness/HSWEll/DHSign/DRTM01/LEAK                 [unit] >      &
$          (OUTput [tbegblk] [deltblk] Sec/MIn/HR/DAy)
$
$   TABle   'sname'  HEADer / NOHEADer / INDexed 'fname'                     &
$        < DSPR/HSign/DIR/PDIR/TDIR/TM01/RTM01/RTP/TM02/FSPR/PER/RPER/
$          DEPth/VEL/FRCoeff/WIND/DISSip/QB/TRAnsp/FORce/UBOT/URMS/WLEN/
$          STEEpness/HSWEll/DHSign/DRTM01/LEAK/XP/YP/DIST           >        &
$          (OUTput [tbegtbl] [delttbl] Sec/MIn/HR/DAy)
$
$   SPECout 'sname'  SPEC1D/SPEC2D  ABs/REl   'fname'                        &
$          (OUTput [tbegspc] [deltspc] Sec/MIn/HR/DAy)
$
$   NESTout 'sname'  'fname'                                                 &
$          (OUTput [tbegnst] [deltnst] Sec/MIn/HR/DAy)
$
$   PLOtgeogr  'sname'  (FIle  'fname')   'title'   (COORD [marg])           &
$          ISO  DSPR/HSign/TM01/RTM01/RTP/TM02/FSPR/PER/RPER/DEPTH/
$               FRCoeff/DISSip/QB/UBOT/URMS/WLEN/STEEpness/HSWEll/
$               DHSign/DRTM01/LEAK                    [step]  [min]  [max]   &
$          VEC  VEL/TRAnsp/FORce/WIND/DIR/PDIR/TDIR     [scale] [dist]       &
$          (CMESH) [ipen])                                                   &
$          (SITes [ipen])  (LINes [ipen])  (LOCation ('sname') [ipen])       &
$          (OUTput [tbegnst] [deltnst] Sec/MIn/HR/DAy)
$
$   PLOtstar   'sname'  (FIle  'fname')   'title'   (COORD [marg])           &
$          ISO  DSPR/HSign/TM01/RTM01/RTP/TM02/FSPR/PER/RPER/DEPTH/
$               FRCoeff/DISSip/QB/UBOT/URMS/WLEN/STEEpness/HSWEll/
$               DHSign/DRTM01/LEAK                    [step]  [min]  [max]   &
$          STAR   [scale] [dist] [bundle]                                    &
$          (CMESH) [ipen])                                                   &
$          (SITes [ipen])  (LINes [ipen])  (LOCation ('sname') [ipen])       &
$          (OUTput [tbegnst] [deltnst] Sec/MIn/HR/DAy)
$
$   PLOtpp  'sname'  FIle 'fname' 'title'                                    &
$           PROBlems FROUDe / PCONV   [symsiz]                               &
$          (SITes [ipen])  (LINes [ipen])  (LOCation ('sname') [ipen])       &
$          (OUTput [tbegnst] [deltnst] Sec/MIn/HR/DAy)
$
$   PLOtspec 'sname'  (FIle  'fname')   'title'                              &
$          SPECtrum  NORMalized / NONORM   ( < [ch] > )                      &
$          FReq  NORm / [fmax]  [fmid]      ABS/REL                          &
$          (OUTput [tbegnst] [deltnst] Sec/MIn/HR/DAy)
$
$   COMPUTE
$   COMPUTE  STATionary [time]
$   COMPUTE  NONSTat [tbegc] [deltc] Sec/MIn/HR/DAy [tendc])
$
$   HOTFile  'fname'
```

```
$
$     POOL
$
$     STOP
```

Appendix D Spectrum files, input and output.

This appendix describes the format of the files for spectral input (command BOUNDARY) and output (commands SPEC and NEST) by SWAN. The presence of the keyword SWAN and a version number recognize the files on the first line of the file. This description is valid for version number 1.

These files contain the following information:

- coordinates of locations.
- frequencies.
- directions (if used for 2-D).
- time (if time-dependent).
- spectral energy or variance densities (and aver. dir. and dir. spread if 1-D).

<u>Example of a 1-D nonstationary spherical coordinates file:</u>

```
SWAN 1                                        Swan standard spectral file, version
$ Data produced by SWAN version 40.11
$ Project:'projname'              ;           run number:'runnum'
TIME                                          time-dependent data
   1                                          time coding option
LONLAT                                        locations in spherical coordinates
   2                                          number of locations
            1.00       1.00
            1.20       1.00
RFREQ                                         relative frequencies in Hz
      25                                      number of frequencies
      0.0418
      0.0477
      0.0545
      0.0622
      0.0710
      0.0810
      0.0924
      0.1055
      0.1204
      0.1375
      0.1569
      0.1791
      0.2045
      0.2334
      0.2664
      0.3040
      0.3470
```

```
        0.3961
        0.4522
        0.5161
        0.5891
        0.6724
        0.7675
        0.8761
        1.0000
QUANT
   3                                       number of quantities in table
VaDens                                     variance densities in m^2/Hz
m^2/Hz                                     unit
  -0.9900E+02                              exception value
CDIR                                       average Cartesian direction in degr
degr                                       unit
  -0.9990E+03                              exception value
DSPRDEGR                                   directional spreading
degr                                       unit
  -0.9000E+01                              exception value
19680606.030000                            date and time

LOCATION     1
     0.3772E-03        190.1          6.3
     0.1039E-02        190.2          6.5
     0.2281E-02        190.3          6.7
     0.3812E-02        190.3          6.7
     0.4255E-02        190.3          6.6
     0.2867E-02        190.1          6.3
     0.1177E-02        189.6          5.8
     0.3892E-03        192.0         15.2
     0.8007E-03        244.5         22.9
     0.6016E-02        251.4         11.5
     0.1990E-01        251.0         11.0
     0.3698E-01        249.9         10.9
     0.3874E-01        248.1         12.1
     0.2704E-01        246.6         13.0
     0.1672E-01        247.0         13.5
     0.1066E-01        247.7         13.7
     0.5939E-02        247.3         14.0
     0.3247E-02        246.5         14.6
     0.1697E-02        245.9         14.9
     0.8803E-03        245.6         15.1
     0.4541E-03        245.5         15.3
     0.2339E-03        245.4         15.5
     0.1197E-03        245.5         15.6
   0.6129E-04      245.5         15.7
   0.3062E-04      245.3         15.9

LOCATION    2
         0.7129E-02       67.2     25.3
         0.3503E-01       67.5     21.7
         0.1299E+00       68.2     19.7
         0.5623E+00       69.7     18.0
         0.1521E+01       71.4     18.0
         0.3289E+01       74.0     18.8
         0.4983E+01       77.2     20.3
         0.4747E+01       79.9     22.0
         0.2322E+01       79.4     30.7
         0.1899E+01      341.1     56.2
         0.1900E+01      314.6     39.4
         0.6038E+01      324.3     31.9
         0.8575E+01      326.1     31.0
         0.4155E+01      325.1     30.5
         0.1109E+01      322.8     32.9
         0.7494E+00      323.1     33.3
         0.4937E+00      323.1     33.3
         0.2953E+00      323.3     33.7
         0.1661E+00      323.6     34.0
         0.9788E-01      323.7     33.8
         0.5766E-01      323.8     33.6
         0.3397E-01      324.0     33.5
         0.2001E-01      324.1     33.4
         0.1179E-01      324.2     33.3
         0.6944E-02      324.2     33.2
```

Example of a 2-D stationary Cartesian coordinates file :

```
SWAN 1                                  Swan standard spectral file, version
$ Data produced by SWAN version 40.11
$ Project: 'projname'                   run number: 'runnum'
LOCATIONS                               locations in x-y-space
   2                                    number of locations
            0.00                0.00
        22222.22                0.00
RFREQ                                   relative frequencies in Hz
     25                                 number of frequencies
     0.0418
     0.0477
     0.0545
     0.0622
```

```
          0.0710
          0.0810
          0.0924
          0.1055
          0.1204
          0.1375
          0.1569
          0.1791
          0.2045
          0.2334
          0.2664
          0.3040
          0.3470
          0.3961
          0.4522
          0.5161
          0.5891
          0.6724
          0.7675
          0.8761
          1.0000
CDIR                                   spectral Cartesian directions in degrees
          24                           number of directions
          7.5000
          22.5000
          37.5000
          52.5000
          67.5000
          82.5000
          97.5000
          112.5000
          127.5000
          142.5000
          157.5000
          172.5000
          187.5000
          202.5000
          217.5000
          232.5000
          247.5000
          262.5000
          277.5000
          292.5000
          307.5000
          322.5000
```

```
     337.5000
     352.5000
QUANT
   1                                number of quantities in table
VaDens                              variance densities in m2/Hz/degr
m2/Hz/degr                          unit
  -0.9900E+02                       exception value


FACTOR
      0.422574E-11
     0    0    0    0    0    0    0    0    0    0    0    0     1     0     0
 0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    44     3     0
 0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0   817    60     1
 0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0  8018   574     9
 0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0 39230  2532    38
 0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0 92174  4477    68
 0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0 99010  1946    29
 0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0 47054   131     2
 0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0 13228     0     0
 0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0 39417     0     0
 0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0 61269     0     0
 0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0 29738     0     0
 0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0  2161     0     0
 0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0     2     0     0
 0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0     0     0     0
 0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0     0     0     0
 0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0     0     0     0
 0    0    0    0    0    0    0    0    0
```

```
    0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
 0     0     0     0     0     0     0     0     0
    0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
 0     0     0     0     0     0     0     0     0
    0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
 0     0     0     0     0     0     0     0     0
    0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
 0     0     0     0     0     0     0     0     0
    0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
 0     0     0     0     0     0     0     0     0
    0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
 0     0     0     0     0     0     0     0     0
    0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
 0     0     0     0     0     0     0     0     0
    0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
 0     0     0     0     0     0     0     0     0

FACTOR
    0.675611E-06
    51   242   574   956  1288  1482  1481  1286   957   579   244    51     0     0     0
 0     0     0     0     0     0     0     0     0
   129   610  1443  2402  3238  3725  3724  3234  2406  1454   613   128     0     0     0
 0     0     0     0     0     0     0     0     0
   273  1287  3054  5084  6846  7872  7869  6837  5091  3076  1295   271     0     0     0
 0     0     0     0     0     0     0     0     0
   665  3152  7463 12402 16712 19229 19221 16690 12419  7518  3172   662     0     0     0
 0     0     0     0     0     0     0     0     0

  1302  6159 14608 24275 32688 37618 37603 32644 24309 14716  6198  1296     0     0     0
 0     0     0     0     0     0     0     0     0
  2328 10989 26020 43341 58358 67109 67080 58281 43401 26213 11058  2317     0     0     0
 0     0     0     0     0     0     0     0     0
  3365 15922 37712 62733 84492 97150 97110 84380 62820 37991 16021  3349     0     0     0
 0     0     0     0     0     0     0     0     1
  3426 16230 38440 63939 86109 99010 98969 85995 64027 38724 16331  3410     0     0     0
 0     0     0     0     0     0     0     0     1
  2027  9612 22730 37790 50909 58529 58505 50841 37843 22898  9672  2018     0     0     0
 0     0     0     0     0     0     0     0     1
   672  3178  7538 12535 16892 19440 19432 16870 12552  7594  3198   669     0     0     0
 0     0     0     0     0     0     0     0     0
   101   479  1135  1890  2542  2924  2923  2539  1892  1144   482   101     0     0     0
 0     0     0     0     0     0     0     0     0
     2    11    26    43    57    66    66    57    43    26    11     2     0     0     0
 0     0     0     0     0     0     0     0     0
     0     0     0     1     1     1     1     1     1     0     0     0     0     0     0
```

```
0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0 0 0    0
0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0 0 0    0
0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0 0 0    0
0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0 0 0    0
0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0 0 0    0
0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0 0 0    0
0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0 0 0    0
0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0 0 0    0
0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0 0 0    0
0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0 0 0    0
0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0 0 0    0
0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0 0 0    0
0    0    0    0    0    0    0    0    0
```

Note that the true variance or energy densities are obtained by multiplying each number with the factor given under the keyword FACTOR.

Formal description of the 1-D and 2-D spectral file

This description refers to either **write or read** energy/variance density spectra **to or from** the file.

The description of the file to write or read source terms is identical to this description except that the quantities differ.

FREE FORMAT (FORTRAN convention) is the format used by SWAN or the user when offering the file to SWAN, except that all keywords and names of quantities (see below) should start on the first position of the line on which they appear (see Appendix A for the syntax of keywords). This format implies that SWAN ignores all information on each line after the required input for SWAN. This can be used to enter user's information at the discretion of the user.

1st line: the keyword "SWAN" followed by the version number

2nd line: If nonstationary computations: the keyword "TIME".

If stationary computations: not present.

3rd line: If nonstationary computations: a number which is the time coding option. It is recommended to use 1 (ISO notation).

If stationary computations: not present.

4th line: The description of the locations:

If using Cartesian coordinates the keyword "LOCATIONS".

If using spherical coordinates the keyword "LONLAT".

Number of locations.

If using Cartesian coordinates:

For each location give an x- and y-coordinate (in m, Cartesian (problem) coordinates).

If the file is used for input to SWAN (but not generated by SWAN) and the user so desires, the names of locations can be written behind the two coordinates. These names are ignored by SWAN when reading the file (see remark on format above).

If using spherical coordinates:

For each location give longitude and latitude.

If the file is used for input to SWAN (but not generated by SWAN) and the user so desires, the names of locations may be entered behind the two coordinates. These names are ignored by SWAN when reading the file.

Next: The frequency data (for 1-D and 2-D spectra):

The keyword "AFREQ" or "RFREQ" (to distinguish between abs. and rel. freq.).

Number of frequencies.

A column with frequencies (each on a new line) always in Hz.

Next: The direction data (only for 2-D spectra in/output):

The keyword "NDIR" or "CDIR" (to distinguish between Nautical and Cartesian direction).

Number of directions.

A column with directions (each on a new line) always in degrees.

Then The group describing the quantities in the tables of this file (see the above examples):

The keyword "QUANT".

Number of quantities.

For each quantity:

Name of the quantity.

Unit of the quantity.

Exception value of the quantity, i.e., the value that is written instead of a computed value if that is undefined (not computed, such as mean wave direction if the wave energy is zero)

XXX: Note for 1-D spectra:

The number of quantities is always 3, and the quantities are always energy (or variance) density, average direction (CDIR for Cartesian direction and NDIR for Nautical direction) and directional spread (DSPR in terms of DEGREES). (SWAN writes the keyword "DSPRDEGR". SWAN reads the keywords "DSPRD" or "DEGR" in case of option DEGREES or the keywords "DSPRP" or "POWER" in case of option POWER in command BOUNDPAR1). The quantities appear in the order in which they appear in this description.

Note for 2-D spectra:

The number of quantities is always 1. The quantity is always energy or variance density ("EnDens" is the (short) name of true energy densities, "VaDens" for variance densities).

Then: The group with the tables of the quantities:

VVV Date and time (not present for stationary computations).

For each location:

If 2-D spectrum:

- The keyword "FACTOR". FACTOR is replaced by the keyword ZERO if the spectrum is identical 0. FACTOR is replaced by NODATA if the spectrum is undefined (not computed, as on land and no numbers follow).
- The factor to multiply the values in the following table.
- Scaled energy/variance densities (truncated by SWAN to integer values for compact writing. SWAN accepts these values as reals when reading this file. Other programs e.g., for postprocessing, should also accept these values as reals. The values should be multiplied by the factor to get the proper values of the densities.

Else, if 1-D spectrum:

- The keyword LOCATION followed by the index of the location (on the same line). This is replaced by the keyword NODATA if the spectrum is undefined (not computed e.g. on land. No numbers follow).
- Else, a table containing three columns. The three quantities per frequency: energy (or variance) density, average direction (CDIR for Cartesian direction and NDIR for Nautical direction) and directional spread (DSPR in terms of DEGREES (writing or reading the file) or POWER (only reading the file), see XXX). The quantities appear in the order in seen in this description.

For nonstationary computations repeat from VVV.

Appendix E Acronyms

ASCE	American Society of Civil Engineering
BSBT	Backward Space-Backward Time Scheme
CCM	Central Conformal Mercator
DIA	Discrete Interaction Approximation
DTA	Discrete Triad Approximation
GSE	Garden Sprinkler Effect
HPGL	HP Graphick Language
LTA	Lumped Triad Approximation
OPPL	Ocean Pack Plot Code
SORDUP	Second Order Upwind Scheme
S&L	Stelling and Leendertse's Second Order with Third-Order Diffusion Scheme
SWAN	Simulating Waves Nearshore