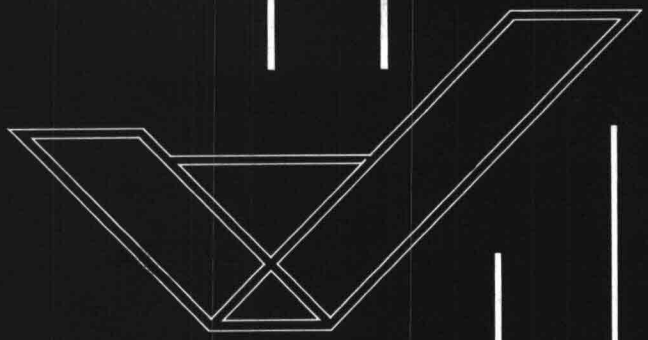


The shallow water wave hindcast model HISWA Part III: user manual

N. Booij
L.H. Holthuijsen
T.H.C. Herbers

Rep. No. 8-85

Delft University of Technology
Department of Civil Engineering
Group of Fluid Mechanics

The shallow water wave hindcast model HISWA Part III: user manual

N. Booij
L.H. Holthuijsen
T.H.C. Herbers

Rep. No. 8-85

PROJECT REPORT	**Delft University of Technology** **Department of Civil Engineering** **Group of Fluid Mechanics**
Project title	GEOMOR wave model (HISWA)
Project description	Development of a two-dimensional model to hindcast spectral wave parameters in an estuary with tidal flats on the basis of bottomtopography, current and wind data
Customer	Rijkswaterstaat Deltadienst, Afdeling Kustonderzoek van Alkemadelaan 400 2597 AT THE HAGUE, the Netherlands
represented by	J. v. Marle
Project leader	dr.ir. L.H. Holthuijsen
work carried out by	dr.ir. N. Booij dr.ir. L.H. Holthuijsen ir. T.H.C. Herbers G. Marangoni
Conclusion	This part III contains the user manual of the HISWA model
Status of report	confidential, final report, part III
City/date:	Delft, February 20, 1985

Part III. HISWA user manual

date of printing: January 4, 1985

Contents

1. PHYSICAL AND NUMERICAL BACKGROUND

1.1. Introduction

The model HISWA is a numerical model to obtain realistic estimates of wave parameters in coastal areas, lakes and estuaries for given stationary wind-, bottom-, and current conditions. The basis of the model is a parameterized version of the energy balance of the waves (or action balance in the presence of currents).

The notion of energy *) balance is to equate for each spectral wave component the rate of change of its energy density to the rate of energy transfer due to such physical processes as wind growth, bottom disspation etc.. In many conventional spectral wave models each such wave component is followed across the area of interest. During this journey the effects of wind, bottom etc. are determined and accumulated until th wave component arrives at the location of interest. This process is repeated for all spectral components (all frequencies and all directions) so that eventually the full two-dimensional spectrum is determined at the location of interest.

The technique which is used in HISWA is similar to the above described technique but it differs from it in that per spectral direction the spectrum is parameterized in frequency domain with one (variable) frequency and a (variable) directional energy density (hence the characteristic "parameteric in frequency" of the HISWA model). The wave component is thus characterized by its direction only; its energy density and its frequency (equal to the mean frequency of the waves from that direction) are computed in HISWA by integrating the local effects of wind, bottom and currents while propagating the waves over the area of interest. These computations are carried out for each direction separately and virtually independently from each other (hence the characteristic "directionally decoupled" of the HISWA model). For this HISWA uses two equations for each spectral direction, both obtained by integrating the spectral energy balance of the waves over the frequencies :

- the energy balance equation for the energy density of all wave components from the direction under consideration
- an evolution equation to determine the mean frequency of all wave components from the same direction.

---

*) In this manual the term "energy" stands for the more appropriate term "variance", it stands for "action" whenever a mean current is present.

The physical phenomena which are accounted for in these equations are :

a) refraction

b) diffraction

c) wind growth

d) bottom dissipation

e) surf dissipation

f) current dissipation

These phenomena are addressed separately below.

1.2 Refraction

Refraction which is commonly defined for one spectral wave component as the change in propagation direction due to variations along the wave crest of phase speed, is usuallly computed with the so-called ray-technique.From an arbitrarily chosen location (usually on the up-wave boundary of the area of interest) the path of the wave components under consideration is computed from the initial direction and frequency on the basis of local depth and current information. The wave direction is thus a dependent variable as the computations progress along the ray. During the journey along the ray such a model can take into account the local effects of wave growth and dissipation. This is essentiallly a Lagrangian approach : the changes are considered while travelling with the component.

In HISWA the approach is basically Eulerian : the changes in the wave component are considered at one fixed location and at fixed spectral directions. This different approach requires that the wave energy density and frequency are considered as a function of direction at the fixed location. Refraction is then accounted for by transferring energy across the directions. The mathematical formulation for refraction that is used in HISWA can therefore be considered as an Eulerian formulation of the conventional refraction theory. It is applicable to both bottom- and current refraction.

The above Eulerian formulation is used in HISWA in the energy balance equation and in the evolution equation for the determination of the mean frequency. This implies that in contrast to conventional refraction computations, these refraction computations are carried out for a wave frequency which varies due to wave growth and decay. This frequency is constant when growth and decay are absent.

1.3. Diffraction

Diffraction is commonly defined for one spectral wave component as the change in propagation properties due to variations along the wave crest in wave energy. Its effect is to redistribute wave energy over the area of interest, away from areas of high wave energy towards areas of low wave energy, and to change the wave direction

correspondingly. The computation of diffraction in arbitrary geophysical conditions is rather complex and requires considerable computing effort. To avoid this the theoretical correct approach is not used in HISWA. Instead a diffraction-like behaviour of the waves is modelled in the sense that the same qualitative behaviour of spatial redistribution and changes in wave direction is obtained. This is achieved by transferring wave energy across the spectral directions (at the fixed location under consideration). The rate of transfer depend on the local gradient along the wave crest of the energy density of the wave component considered. The formulation is such that currents affect this diffraction-like behaviour.

The above diffraction simulation is used in HISWA in the energy balance and in the evolution equation for the determination of the mean frequency. This implies that, in contrast to conventional diffraction computations, the diffraction-like computations are carried out for a wave frequency which varies due to wave growth and decay. The frequency is constant if wave growth and decay are absent.

1.4. Wind growth

The growth of the wave energy is considered for each spectral direction separately and independently from other spectral directions. The growth is taken equal to the growth for the considered direction in an idealized situation (unlimited ocean with homogeneous and stationary wind) as published in an extensive study of advanced spectral wave models (Allender, 1981), under the assumption that in that situation the directional energy distribution has a universal shape.

The evolution of the mean frequency is also considered separately and independently for each direction. It is directly related to the growth of the wave energy in the considered direction. This evolution of the mean frequency is based on an assumed universal relationship between the wave energy and the mean frequency (in dimensionless form). When the wave field does not conform to this universal relationship (e.g. by rapid changes in the bottom or current pattern or through user-imposed boundary conditions), the evolution of the mean frequency is adapted to force the wave field towards this relationship.

The effect of currents is included by using the apparent wind speed and direction in the growth and evolution expressions rather than the real wind speed and direction.

1.5 Bottom dissipation

In shallow water the wave energy in each spectral direction is dissipated in HISWA by bottom friction. This dissipation is determined with a fairly conventional nonlinear bottom friction model including the effect of a mean current. It is based on a conventional formulation for periodic waves (quadratic friction law) with the appropriate parameters adapted to suit a random wave field as considered in HISWA.

The mean wave frequency is only indirectly affected by bottom friction in HISWA, virtually by an assumed

relationship with the wave energy dissipation due to bottom friction.

1.6. Surf dissipation

In extremely shallow water the waves break in a surf zone. The corresponding energy dissipation is determined in HISWA with a bore-model for the waves which are higher than some treshold value. Only the total rate of energy dissipation is thus determined (i.e. integrated over all directions). This total rate is distributed over the directions proportional to the wave energy in a direction.

The mean wave frequency is affected only indirectly by surf dissipation in HISWA, virtually by an assumed relationship with the wave energy dissipation due to surf breaking.

The effect of a mean current on surf breaking is not taken into account.

1.7. Current dissipation

In a strong adverse current some wave energy is carried away by the current. This is energy which in a full spectral model (all directions and frequencies included) is carried by wave components that cannot travel against the current (propagation speed less than current speed). This energy is removed from the wave field in the HISWA model through high frequency dissipation in an assumed standard frequency spectrum.

1.8. The numerical procedure

The balance equations used in HISWA are partial differential equations of the first order (apart from the diffraction-like terms), with two horizontal coordinates (x and y) and wave direction (θ) as independent variables. The state in a point in (x,y,θ)-space is determined by what happens at the upwave side of this point.
Therefore a numerical procedure can be employed in which the wave state is determined point by point; this method is known as the explicit method.

The computational region is a rectangular grid. One of the axe (called the x-axis) is chosen in the downwave direction. The computation starts at the upwave boundary x=0 and proceeds in positive x-direction. After the states in all points on a line perpendicular to the x-axis have been determined, the computation proceeds with the next line in the grid (see figure 1.1)

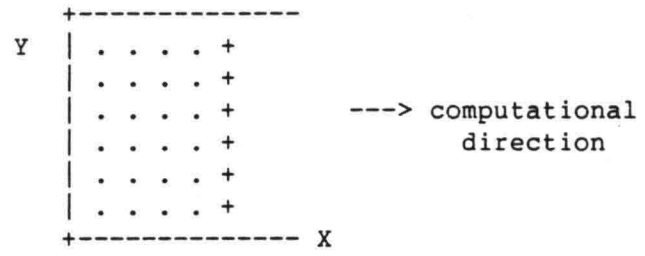


figure 1.1. computational grid
. points determined previously
+ points being computed

The procedure has certain consequences. First the computation is only stable under the condition that the ratio of the forward step (i.e. the step in the computational direction) and the lateral step (step size in the direction normal to the computational direction) as well as the ratio of the forward step and the directional step are lower than a certain limit. The stability conditions are:

$$DX/DY < Pn * Cx / Cy , \qquad (1)$$

In cases without current this is equivalent with:

$$DX/DY < Pn * Cotg(\theta) . \qquad (2)$$

The second stability condition reads:

$$DX/D\theta < Pd * Cx / C_{\theta} , \qquad (3)$$

which is roughly equivalent with:

$$DX/D\theta < Pd * Depth/Slope ,$$

where DX is the forward step, DY the mesh size in normal direction and $D\theta$ the directional step size; Slope is the bottom slope. Pn and Pd are dimensionless constants, both equal to 0.7. A consequence of the first condition is that only waves can be computed which deviate from the computational direction by less than a certain angle, which is always smaller than 90 degrees. A consequence of the second condition is that DX must be at most of the order of the mesh size in the bottom grid, and usually smaller.

The partial differential equation can only be solved if a number of boundary conditions are given. As a general rule, on every boundary the waves must be prescribed which enter the computational region.

1.9. Limitations of the model

In this section the limitations of the model are discussed. These limitations are the consequences of the assumptions of the mathematical model and the method of computation.

The model is stationary. It can be used if the travel time of the waves over the computational region is small compared with the time scale of the currents, e.g. the tidal period, or in case of a storm surge, with the characteristic period of the surge.

The wind in HISWA is both stationary and uniform over the region. Therefore the travel time of the waves must be small compared with the time scale of the wind, and the dimensions of the computational region must be small compared with the length scale of the wind.

Diffraction is approximated only crudely in HISWA, so it cannot be a major effect in a problem treated by HISWA. Therefore the limiting condition is determined by the refraction part of the model. This condition is that the length of the waves must be small compared with the length scale of depth and current.

The forward stepping procedure allows only waves propagating in forward direction to be computed. Reflecting or backscattering waves are not accounted for.

The depth must be given to the model before computation. So wave setup cannot be taken into account directly. In principle however an iterative procedure can be employed in which currents and waves are computed alternatively. Depths and currents are computed (not by HISWA, but by a 2-dimensional flow model) taking into account the radiation stresses accompanying the waves, and waves are computed on the basis of the depths and currents. To enable this procedure to be carried out, HISWA computes the radiation stress gradients (see the output commands in chapter 4).

2. HOW TO USE HISWA

2.1. Introduction

The purpose of this chapter is to give some practical advice to a user, who is preparing his first HISWA run. An important question addressed is how to choose the computational grid and the bottom grid.

Futhermore suggestions are given that should help the user the choose among the many options available in HISWA, for instance in source terms and boundary conditions. A general suggestion is: Start simple. HISWA helps in this because standard options are usually the most simple options.

The possibility of 'nesting' models is also discussed. The idea of nesting is to have a coarse model for a larger region and a finer model for a smaller region. First a run with a coarse model is done, and then one with the finer model; the fine model obtains its boundary conditions from the coarse model.

The last subject of this chapter is the output and its interpretation.

2.2. Dimensions and coordinate systems

Most wave parameters that play a role in the HISWA model, bear some dimension. HISWA expects all quantities that are input by the user, to be expressed by means of the S.I. System of units: m, kg, s and composites of these. Consequently the waves height and water depth are in m, wave period in s etc.

Energy densities are represented in HISWA by the variance density, of which the unit is m2. Consequently the dissipation and energy leak are expressed in m2/s, and energy transport in m3/s. The radiation stress gradients are in m2/s2.

All locations, such as location and orientation of bottom grid and computational grid, locations of output points etc. are given with respect to a common Cartesian coordinate system, chosen by the user. This common system is designated as the user coordinate system in the manual. Horizontal coordinates are always given in meters.

Direction of wind, incident waves etc. are also given with respect to the user coordinate system according to the Cartesian convention, i.e. measured counterclockwise from the positive x-axis of this system (in degrees).

2.3. Choice of grids

2.3.1. Computational grid in real x-y-space

The computational grid must be chosen carefully. Some advice is given here regarding size and orientation of this grid. First, the x-axis of this grid must be more or less parallel to the main wave direction in the region. The command

defining the computational grid (command GRID) has a standard option to make the x-axis parallel to the mean direction of the incident waves.

The side of the grid x=0, where the incident waves enter the grid, is usually chosen in water with such a depth that refraction effects have not yet influenced the wave field. This requirement is not applicable if the data concerning the incident wave field are provided by a model which takes into account refraction, for instance HISWA itself.

Of course the computational grid must be large enough to encompass the area where the user wants to know the wave parameters. The width of the grid (in y-direction) must be larger than what would be necessary for this, because along each lateral side of the grid regions exist where the wave field is disturbed in (the model) due to the fact that the lateral boundary condition is usually not well known (see figure 2.1).

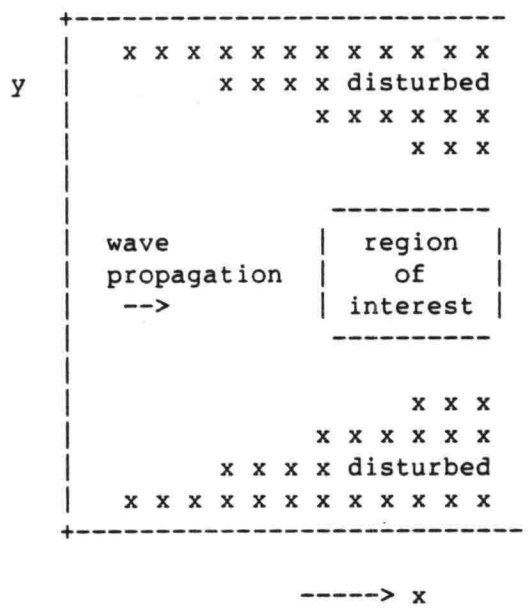


Figure 2.1. Disturbed regions in the computational grid

As a rule of thumb one may assume that the disturbed regions are of triangular shape with a top angle of roughly 20 degrees.

2.3.2. Computational grid in θ-space

The spectral wave directions computed by HISWA can deviate from the computational direction only to a certain extent, but always less than 90 degrees. Usually a good choice is to make the directional sector available to the computations equal to 120 degrees. The directional initial step size is suggested to be about 10 degrees of this sector when using an initial directional distribution with directional spreadingof 30 degrees or more.

The spatial mesh sizes are obviously dependent on the spatial variations of the wave field. A first choice might be of the mesh size of the bottom grid.

The spatial mesh sizes are checked by the program in view of stability; if θ0 is half the sector width, DX and DY must

obey according to inequality (2):

$$DX/DY <= PD * Cotg(\Theta 0)$$

Even then there is no guarantee against instability. If instability occurs, DX should be choosen smaller.

2.3.3. Bottom grid

The bottom grid needs not be identical with the computational grid. It is best to make the bottom grid larger than the computational grid, so large that it completely covers the latter for every expected situation.

2.3.4. Boundary conditions

On three of the four sides of the computational region the user must prescribe a boundary condition. We distinguish the incident wave boundary or startline, x=0, and the lateral boundaries, y=0, and y=Y.

The incident boundary condition is prescribed using the command INC (See chapter 4). Often it is sufficient to prescribe significant wave height and direction of the incident field. The user can leave out the command INC if no waves enter the computational region over that boundary. More complicated possibilities do exist; these may be useful if the spectrum of the incident field is computed by a another program, or by HISWA itself. This occurs if one uses nested models.

The lateral boundaries are different in the sense that both incoming and outgoing waves may be present. The user has no control over the outgoing waves. He can prescribe the incoming waves; for details see command BOUNDARY. As in the case of the command INC, if the command BOUNDARY is not given by the user, it is assumed that no waves enter over the lateral boundaries.

2.4. Physical processes

2.4.1. Wave energy

HISWA simulates a number of effects that add or withdraw wave energy to or from the wave field. The effects modelled are: wind growth, bottom friction and surf breaking.

Wind growth and friction are inactive unless the user explicitly instructs the program to make these terms active. Surf breaking is active at the outset, but the user can modify the parameters. Although the user can modify each coefficient in each of the source terms, there is usually in each term only one coefficient which is modified in practice; see below.

In the first runs however however one does not modify any parameter, but determines whether or not a certain effect is relevant to the result. If this cannot be decided by means of a simple hand computation, one can perform a HISWA

computation without and with the effect under investigation, in the latter case using the standard values chosen in HISWA.

After it has been established that a certain effect is important, it may be worthwhile to modify coefficients. In the case of wind growth (see command WIND) one might at first try to vary the wind velocity. In the bottom friction (see command FRIC) the best coefficient to vary is the friction coefficient [CFW] and, if a current is present, [CFC].

Switching off the surf breaking term is usually unwise, since this leads to unacceptable wave heights near beaches. In the surfbreaking term the coefficients [GAMS] and [GAMD] can be varied.

2.4.2. Wave frequency

Each of the energy dissipation or generation effects may also influence the average frequency. The same approach that was suggested above, regarding the effect on the energy density, also applies here.

The coefficients in the expression for influence of wind on frequency the coefficient to be varied primarily, is [B2], in the case of bottom friction it is [B3], and in surf breaking: [B4].

2.5. How HISWA operates

HISWA consists of three programs that exchange information. The first is the reading part; (HISWA/PREP, see figure 2.3) it reads and processes the user's commands. It also produces the verification plots by preparing data for the contouring program G.P.C.P. (file PCDEP). The second is the computational part (HISWA/COMPU); it carries out the wave computation. It receives its instructions from the reading part via the file INSTR. The computational part produces computational results which are written on the file REKRES; often this file is so large that it is put on the magnetic tape.

The computational results are read from the file REKRES and processed by the third program of HISWA, the output part HISWA/OUTPUT. This part receives it instructions (the so-called output instructions) from the reading part via the file INSTU. The output part can send data to the printer, and to the file PCWAV which serves as input for the contour line program G.P.C.P. The output part can send data to other files also, for further processing by other programs. For instance, if one wants to make a nested model, the output part should be instructed to write spectra to a file. These spectra are read as boundary data by the reading part in the run which carries out the computation in the nested model.

The communication between the parts of HISWA is shown in figure 2.2.

After a run the computation results are not destroyed. Therefore it is possible to have some output made, to study these and if necessary, to ask the program for some

additional output. Such a run is carried out by the version HISWA-11 whereas the full run, including computation, is done by the version HISWA-10.

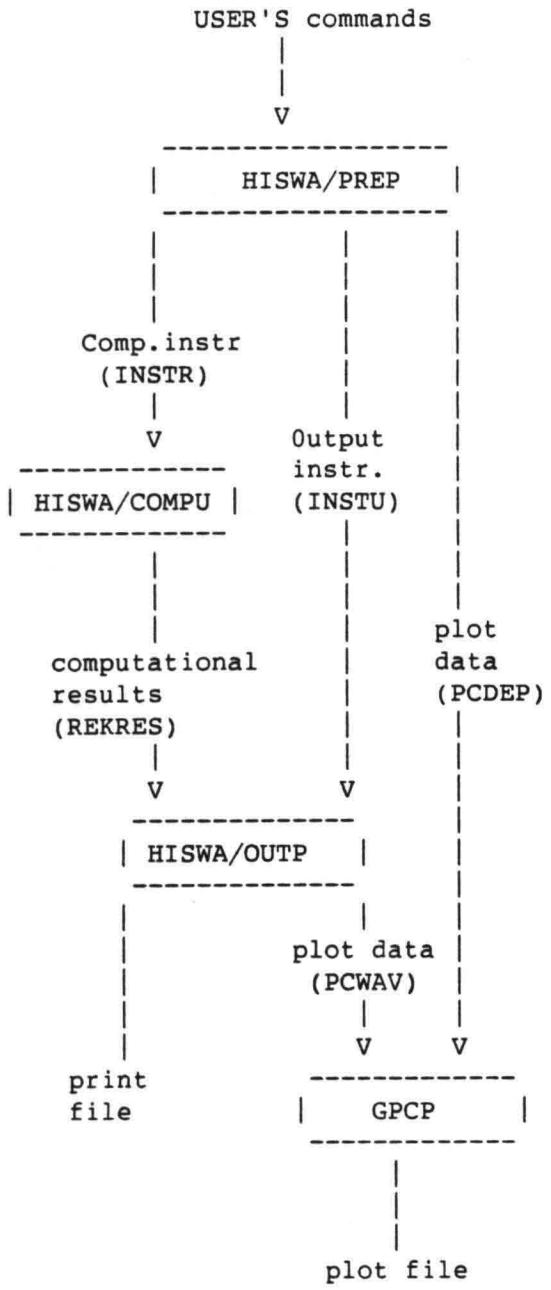


Figure 2.2. Flow chart of the system HISWA-10.

In the version HISWA 11 the computational module HISWA/COMPU is not active. Lines denote transfer of data; file names are between parentheses.

2.6. Output facilities

HISWA is very flexible regarding output. The user can be informed about many different wave parameters and about quantities related with these parameters. The output will be prepared for subregions defined by the user. The general rule is that no output is produced by HISWA, unless the user asks for it.

The output instructions are distinguished into two categories:
- definitions of output regions, or other sets of output points. It was indicated earlier that the orientation of the computational region can vary according to the incident wave direction. The output regions are fixed, and HISWA takes care of the interpolation from the computational region to the output region.
- output requests, asking that certain wave parameters are printed, plotted, or written to secondary storage. Various forms of print and plot are avaiable.

The number of output regions, and that of output requests are free.

The wave parameters which are printed and/or plotted are either scalars or vectors. Scalar quantities are for instance: significant wave height, average period, directional spread and dissipation. Vectorial quantities are energy transport and radiadion stress gradient. Depth and current velocity can also be shown in the output.

In the visual presentation the scalar quantities are shown by means of iso contour lines, and the vectorial quantities by arrows in grid points of the output region. In the prints the values of the scalar quantity or the values of the vector components at the same grid points are printed.

2.7. Interpretation of the output

In the interpretation of the output it is essential to bear in mind the schematizations employed in the HISWA model. In most applications the most severe schematization is in the fact that the spectrum is integrated over frequencies. Although HISWA assumes standard shape of the spectrum it is not possible to draw any conclusions from the computational results regarding the real shape of the spectrum. Comparison with measurements can only be done in terms of total energy density and/or average frequency. Note however, that the average frequency (averaged over action) is not equal to the peak frequency. As a rule the significant wave period or the peak period are 10 to 20% longer than the average period. This difference plays a role both in the input quantities, such as the period of the incident wave field, and in the interpretation of the output.

Another important approximation is in the diffraction effects. Since this effect is simulated only rather poorly, the wave field immediately behind obstacles cannot be expected to be accurate.

3. RUN-PROCEDURES

Two run-procedures for the UNIVAC-system exist, viz. HISWA10 and HISWA11. HISWA10 is intended for a run with computation, HISWA11 for a run with only output, based on computational results obtained in a previous run.

The call of HISWA10 reads:

```
@PRO  DDTHGEOMOR*HISWA-ECL.S10, DEPF, bottom file,
      OPC, run number, PLOT, plot number, BPT, printfile
```

followed by the HISWA commands (see chapter 4)

The list of parameters in PRO is incomplete; the complete list follows. The meaning of the parameters is:

DEPF	file containing bottom levels and possibly current velocities,
OPC	run number, used to distinguish results of different jobs,
PLOT	plot number, used to distinguish plots,
BPT	output print file, is relevant only in foreground jobs; first give @CAT for this file.
NEST	file containing incident wave data, or data for the full boundary of the model.
FBR	file containing data for the right boundary condition.
FBL	file containing data for the left boundary condition.
F20	output file with reference number 20; can be used to transfer data from HISWA to another program.
F21	output file with reference number 21.
F22	output file with reference number 22.
VERSION	Version 1 is the production version, version 2 the test version; 1 is standard, so in case a test run is made, add: VERSION,2
WERKF	workfile, for SAVING input data.
WERKF2	secondary workfile.

To generate a batch job one can replace the procedure call by @PRO,BG ... and otherwise the same data as above.

To make a run with HISWA11, give

```
@PRO  DDTHGEOMOR*HISWA-ECL.S11, DEPF, bottom file,
      OPC, run number, PLOT, plot number, BPT, printfile
```

The parameters are identical with those of HISWA10.
In order to obtain the good computational results, make sure that the parameter OPC is the same as in the run of HISWA10 in which these results were computed.

4. DESCRIPTION OF COMMANDS

4.1. List of available commands

The following commands are available to users of HISWA:

General commands

PROJECT	Title of the problem to be computed
POOL	assign space for dynamic data pool
TEST	requests the output of intermediate results for testing purposes.
STOP	End of user's input. Starts a computation.
RESTORE	Brings data of old run back into memory.

Commands for model description

SET	sets values of certain general parameters
GRID	position and size of computational grid
BOTTOM	position and size of bottom grid
READ	read bottom and/or current velocity values
INC	defines incident wave field
BOUND	boundary conditions at lateral sides of the computational grid
WIND	wind influence on wave field
BREAK	breaking of waves
FRIC	bottom friction
DIF	diffraction
NUM	numerical parameters

Output commands

FRAME	defines an output frame
CURVE	defines an output curve
NGRID	defines a nested grid
POINTS	defines a set of individual output points
BLOCK	requests a block print
TABLE	requests printing of a table
PAPER	data concerning plotting paper
PLOT	requests plotting of a figure
SPEC	spectral output
NEST	spectral output for nesting models
SHOW	requests a verification plot

4.2. Format of the input

The input for HISWA is organized in the form of commands. Each command is designated by a keyword consisting of letters and (sometimes) digits, but always beginning with a letter. After this keyword usually other data appear, such as real or integer numbers, or character strings. Character strings must always be enclosed in quotes; keywords are not in quotes. Strings and keywords have an entirely different meaning to the program; a string is a variable, a keyword has a fixed meaning instructing the program to perform certain actions.

It is not always necessary to actually write down all the data required by the program. In many cases the program will assume reasonable values for variables that do not appear in the command. The command description will mention whether an initial value or a default value is assigned to a variable. An initial value is assigned by the program at the very start of the job, whereas a default value is assigned at the moment that the command is executed. Often an initial value and a default value are different.

Commands are described in the following form:

```
                |     ST 'NAME'  [X]  [Y]  |
KEYWORD     <   |     -                    >
---             | ->  NU [RA]  [RB]        |
                |     -
```

The following rules apply for the command description:

- Keywords are not enclosed by square brackets or quotes, the letters that are underlined must be copied literally; other letters or digits may be added, as well as the characters - and _. So in the command outlined above one may write: KEY or KEYW or KEY-WORD or KEYHOLE etc.
  The first keyword in the command scheme is also the command name.
- A name between square brackets is to be replaced by a (real or integer) number; a name between quotes is to be replaced by a string, also enclosed in quotes. In the command description one can find what the program does if a variable is not assigned a value by the user. The description also should make clear whehter a real or integer number is expected.
  Note: a decimal point is not permitted in an integer number.
  The data must be given in the same order as they appear in the description. If one wants to assign the value 6 to the variable [RA], one writes 6 or 6., or RA=6.
- If one line of input is not large enough to hold the data for one command, the command can be continued on the next line, if the last one has a continuation mark as its last item. The following continuation marks can be used: & or _ (the underscore, not the minus sign). In the command descriptions the & also signifies a continuation mark.
- A group of data between parentheses ( ) is optional; the command description tells what happens if the group of data does not appear int the user's command.
- A group of data within angle brackets < > can be given repeatedly. In the user's input the end of the repetition

is indicated by the end of the line, by the appearance of a keyword, or by one of the following characters: / or]. The group of data must be given at least once, unless it is also surrounded by parentheses.

- Alternative options in the command are written between braces of the following form:

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

  Each alternative is characterized by a keyword (in the above example: ST and NU). If an arrow appears before one of the alternatives (->), this alternative is chosen if none of these keywords appears.
- Data are separated by blanks and/or commas. A keyword is closed by a blank or one of the following characters: = or :. An empty data field in a series of data fields is recognized only if it is surrounded by commas. Also the program will assume that a data field is empty is it finds a keyword where a data item is expected.
- If the user wants to write a set of identical data fields, he can use the repetition factor, e.g.

```
    6 * 'I.O.U.', 3*0.,
```

  Note: The repetition factor cannot be used in front of a coefficient that may appear as the first on a new line of input, nor can it be used to give a number of empty data fields.
- To clarify the meaning of the input, one can insert comment. Comment must be enclosed in $ signs. If there is only one $ sign on a line it is assumed that the end of the line is the end of the comment; the next line is considered again as ordinary input.

4.3. General commands

PROJECT 'TITLE1'

'TITLE2'

'TITLE3'

A description of the run is given in the 3 lines 'TITLE1', 'TITLE2' and 'TITLE3'. Each of these is max. 56 characters long. The lines will be reproduced in the output by the program. Initially all three lines are empty.

POOL [NPOOL]

HISWA stores most numbers in a dynamic data pool, which must be large enough to hold all this information. The command POOL is used to give the pool a proper size. [NPOOL] is the size of the pool in blocks of ... numbers. The initial value of [NPOOL] is 12, which is sufficient for smaller problems.

Important: if the command POOL is used, it must be the first command in the list of commands given by the user.

STOP (NORUN) (SAVE [WF])
---- --- ---

Marks the end of the users input.
After this command the program will start to write the model description and other computational instructions to file INSTR (computational instructions), unless the keyword NORUN is present.
Furthermore it will write the output parameters and output requests to file INSTU (output instructions).
If the keyword SAVE is present, the model description, output data and output requests are written to the file with reference number [WF]. This file is called the Workfile, it can be used if one wants to carry out a series of computations without being forced to repeat the model description etc. every time. The SAVED data can be restored in the next run by the command RESTORE. Default: [WF]=12

RESTORE [WF]

This command RESTORES the data that were SAVED by the command STOP in a previous run. Init: [WF]=12
If this command is used, it should be the first in that run.

The reason is that it will destroy all information entered by previous commands in the same run, and will replace it by information from the run in which the SAVE was done.
One can do in the same run a RESTORE and a SAVE on the same file. In that case the information from the previous run is lost. If one wants to retain the information from the previous run, the value for [WF] in command RESTORE must differ from the one in the command SAVE.

After RESTORE one has to enter only the commands needed to change the model, or the output.

TEST [ITEST] [ITRACE] [IX] [IY]

If the program produces unexpected results, this command can be used to instruct the program to produce intermediate results during the computation process.

Parameters:

[ITEST] instructs HISWA to produce test output; the larger the value of [ITEST], the more output is produced. For values under 100 the amount is usually reasonable, for values above 200 it can be huge.
Init: [ITEST]=0, so no test output is made;
Default: [ITEST]=30, test output is made that can be interpreted by the user. For higher values of [ITEST] output is made, that can only be interpreted by those who have the program source listing at their disposal.

[ITRACE] instructs HISWA to produce a message if subroutines are entered.
[ITRACE]=1: only the first entry is signalled;
[ITRACE]=2: every entry is signalled;
Init: [ITRACE]=0 (no messages).

[IX], [IY] instruct the program to produce detailed information about the computational process for one point of the computational grid. the numbers indicate the point in computational grid coordinates, in terms of number of meshes. Default: none, no output of this kind is produced.

4.4. Commands for model description

SET [LEVEL] [MAXERR] [GRAV] [NEGMES]

Assigns values to various general parameters.

[LEVEL] is the (constant) water level in the region (in m). The depths used in the wave propagation computation are equal to the sum of the water level defined by this command, and the bottom level read by the command READ. Init: [LEVEL]=0.

[MAXERR] indicates at which error level a computation may be started. The error level is coded as follows: 1: Warnings, 2: (possibly repairable) errors, 3: Severe errors. Init: [MAXERR]=1

[GRAV] is the gravitational acceleration (in m/s2). Init: [GRAV]=9.81

[NEGMES] maximum number of error messages during the computation. During the computational process messages are printed, if for instance the action density or the frequency in a point becomes negative. Such negative values are a sign that instabilities might have occurred. It is assumed that after a lot of such messages have occurred, the results have become useless, so that the computation is terminated. Init: 200

GRID [XLEN] [YLEN] [SECTOR] [MX] [MY] [MD] &

FIXED [XPC] [YPC] [ALPC]
< >
ROT [XPR] [YPR] [XCR] [YCR]

Defines the position and size of the computational grid. The X-axis of this grid is the computational direction, the Y-axis is normal to this direction. The X-axis should be roughly in the mean wave propagation direction. The orientation of the grid can either be FIXED, or ROTATING. In the latter case the program chooses the X-axis to be coincident with the average direction of the incoming waves given by the command INC; this is allowed only for parametric incident waves (see command INC PARA ...).

Meaning of the parameters:
[XLEN] length of the grid in X-direction (in m).
[YLEN] length of the grid in Y-direction (in m).
[SECTOR] directional interval of propagation directions for which the wave energy density will be determined (in degrees). This sector must be smaller than 180 degrees. The X-direction shall be in the middle of the sector.

[MX] number of meshes in X-direction. In view of numerical stability the program will check whether [XLEN]/[MX] is smaller than [YLEN]/[MY] divided by tan(0.5*[SECTOR]). If not, an error message is printed.
[MY] number of meshes in Y-direction.
[MD] number of subdivisions of the directional interval, so [SECTOR]/[MD] is the spectral directional resolution.
[XPC] Fixed grid: position of the origin of the computational grid in terms of user coordinates (X-coordinate, in m).
[YPC] ,, (Y-coordinate, in m).
[ALPC] Fixed grid: orientation of the computational grid with respect to the X-axis of the user coordinate system (angle in degrees, measured counterclockwise).
[XPR] Rotating grid: position of the rotation point in user coordinates (in m).
[YPR] ,,
[XCR] Rotating grid: position of the rotation point in computational grid coordinates (in m).
[YCR] ,,

BOTTOM [XUB] [YUB] [ALUB] [MX] [MY] &

[DX] [DY] [DEPX]

Defines the position and size of the bottom grid. If a current field is present the current has to be given at the same points as the bottom.

Parameters:
[XUB] position of the origin of the bottom grid in user coordinates (in m).
[YUB] ,,
[ALUB] orientation of the bottom grid with respect to the positive X-axis of the user coordinate system (angle in degrees, measured counterclockwise).
[MX] number of meshes in X-direction of the bottom grid.
[MY] number of meshes in Y-direction of the bottom grid.
[DX] mesh size in the bottom grid (in m).
[DY] mesh size in the bottom grid (in m); default: equal to [DX].
[DEPX] depth outside the bottom grid (in m); if the computational region contains points outside the bottom grid, in these points the depth will be taken equal to [DEPX]. Default: [DEPX]=1000.

The bottom levels themselves are read by the command READ (see next command). These are taken positively downward, measured from a datum chosen by the user. A certain water level (read by SET LEVEL) constant over the region can be added. If the water level is not horizontal, one should input the actual depth, i.e. the bottom level measured from the water surface, instead of the depth relative to the chosen datum.

```
          |  BOTTOM                  |
          |  ---                     |
  READ  <                | -> SEP |    >     &
  ----    |              |    -   |    |
          |  CURRENT  <             >  |
          |  ---         |  COMB  |
                         |  -     |


        [FAC]  [IDFM]  [IDLA]  'FORMAT'  (PRINT)
                                          ----
```

With this command the bottom configuration or the current field is read from the file DEPF (see chapter 3) by the program.

BOTTOM means bottom levels have to be read from the file DEPF (see chapter 3). the level is defined positive in downward direction.

CURRENT means current velocities are read; both the X- and Y-components are expected to be available on the input file DEPF.

SEP/COMB indicates whether the velocity components appear in the file separately or combined, i.e. (case SEP) first all X-components and then all Y-components, or (case COMB) X- and Y-components for each point together.

[FAC] the depth or velocity values are multiplied by the factor [FAC]. For instance if the depths are given in dm, one should make [FAC]=0.1 . Default: [FAC]=1.

[IDFM] The input of the value is to be formatted. The format used is defined by [IDFM].
In formatted input each number must appear in a certain field on the input line, preferably the decimal point is always used in formatted input.
Continuation marks are not allowed. Simply continue on next line if a line is not long enough to hold all the numbers.
=1: (Default) Format according to BODKAR convention
Format (10X, 12F5).
If IDFM=1, the numbers must be preceded by 6 lines of text which are reproduced in the output.
=2: Format is specified by the user; see 'FORMAT' below.
=5: Format (16F5), i.e. an input line consists of 16 fields
of 5 places each.
=6: Format (12F6), 12 fields of 6 places each.
=8: Format (10F8), 10 fields of 8 places each.

[IDLA] prescribes the order in which the values are read (default: [IDLA]=1) =1: values appear line by line (BODKAR convention)

```
(1,MYB+1), (2, MYB+1), ...   (MXB+1,MYB+1)
(1,MYB),   (2,MYB),    ...   (MXB+1,MYB)
.....
(1,1),     (2,1),       ...   (MXB+1,1)
```

a new line in the array should start on a new line of input.
=2: order of values as with [IDLA]=1, but now a new

line in the array must not necessarily start on a new line of input.
=3: values appear line by line:

```
(1,1),     (2,1),      ...   (MXB+1,1)
(1,2),     (2,2),      ...   (MXB+1,2)
.....
(1,MYB+1), (2, MYB+1), ...   (MXB+1,MYB+1)
```

a new line in the array should start on a new line of input.
=4: order of values as with [IDLA]=3, but now a new line in the array must not necessarily start on a new line of input.
=5: values appear column by column:

```
(1,1),     (1,2),      ...   (1,MYB+1)
(2,1),     (2,2),      ...   (2,MYB+1)
.....
(MXB+1,1), (MXB+1,2),  ...   (MXB+1,MYB+1)
```

a new column in the array should start on a new line of input.
=6: order of values as with [IDLA]=5, but now a new column in the array must not necessarily start on a new line of input.

'FORMAT' a user-specified format, i.e. '((10X, 12F5.0))'

PRINT if this keyword appears the program will print the depth or velocity values read. It will do this according to [IDLA]=4.

Remark: HISWA uses the level of the bottom with respect to a common datum. Therefore a positive depth will usually mean a negative bottom level. If necessary the sign of the value can be corrected by assigning a negative value to the number [FAC].

If the file DEPF does not contain enough values, HISWA will print an error message.

One can obtain a contour plot of the bottom levels by the following command:

SHOW BOTTOM

For details see command SHOW.

```
          | PARAM  [HSIG]  [PER]  [DIR]  [DSPR]
          | ---
   INC  <
   ---    |
          |           | FILE  [NF]
          |           | ----
          | SPECTR  <
          | ----      | <  [E]  [FREQ]  >
```

This command defines the incident wave field. This command in not compulsory; if the command is absent it is assumed that there are no incident waves.

The incident waves can be given either in PARAMETRIC or in SPECTRAL form.

In the PARAMETRIC case the user prescribes: [HSIG], significat wave height, [PER], the mean wave period, [DIR], the mean direction of the waves, and [DSPR], which is a

parameter for the directional spread, defined as follows. The program assumes that the incident waves are distributed over the directions according to the function (Cos(θ/2))**[DSPR]. The default is: [DSPR]=2.

In the SPECTRAL case the spectrum is prescribed either by giving a set of values for energy density and mean period, as function of the direction θ, in which case the same spectrum is assumed for the whole boundary x=0; or by reading the values from a file, in which case the spectrum must be given for each point of this boundary.

The numbers [E] and [FREQ] must be given for each spectral direction, starting with the lowest value of θ, so [MD]+1 occurrences of this pair must be present. The command GRID must be given prior to INC in this case. [E] is the spectral energy density for that direction, and [FREQ] is the mean frequency for the same direction.

With the option FILE [NF] the spectral values are read from a file; the file reference number [NF] must be given. For each point on the boundary x=0, starting with the point y=0 the whole spectrum must be present on the file.

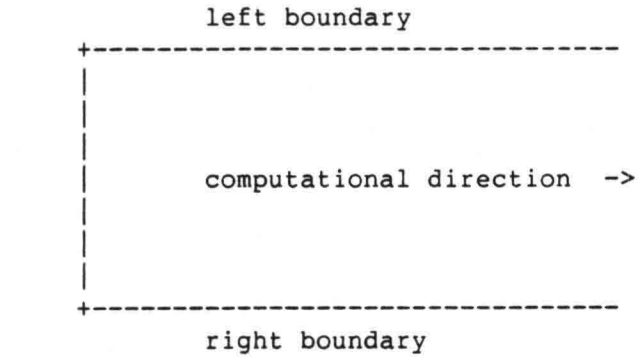
```
              |  NEST
              |  --
   BOUNDARY   <
   ---        |  LEFT   |     |  INC
              |  --     |     |  ---
              |           >   <
              |  RIGHT  |     |  FILE  [NF]
              |  --     |     |  ----
```

This command has two options; first (option NEST) it can be used to inform the model that all boundary conditions (on the startline and the lateral sides of the computational grid) are taken from a larger model; in the computation of this larger model output must have been made by means of the commands NGRID and NEST.
With the option LEFT or RIGHT it is used to determine the boundary conditions at one of the lateral sides of the computational region, i.e. the wave that are incident over that particular boundary. The command has no effect on the waves leaving the computational region; these are always absorbed by the boundary. If the command BOU is absent it is assumed that no waves enter the region over the lateral boundaries.

The RIGHT boundary is the boundary y=0, the LEFT boundary is y=[MY]*[DY].

INC: the wave spectrum at the boundary (only the incoming waves) is the same as that of the wave field entering through x=0, which is defined by the command INC. Applicable only in the case of INC PARAMETRIC.

FILE [NF]: the spectra are read from a file. For each point of the boundary the whole spectrum must be present on the file, so also values for wave directions pointing out of the region must be available on the file, although these do not have any effect on the computation.



WIND [VEL] [DIR] [A] [B] [C] [D] [A2] [B2]

Upon this command a source term due to wind influence is added to the energy balance equation. Only the values for [VEL], the wind velocity (at an elevation of 10 m, unit: m/s) and [DIR] the direction in which the wind blows, with respect to the user coordinate system (in degrees), are required. For the other parameters reasonable values are assumed by the program. These values are empirical. See part I of this report.

BREAKING [GAMS] [GAMD] [ALFA] &

(FREQ [A4] [B4])

With this command a source term due to breaking can be specified. For each coefficient in the formula for the source term a default value is assumed by the program. Initially the breaking term is already present, in contrast with the wind and bottom friction terms, which must explicitly switched on using the commands WIND and FRICTION. Initially no influence of the breaking process on the frequency is assumed. The frequency influence can be switched on by: BRE FREQ .

Parameters:

[GAMS] controls shallow water breaking (depth breaking): ratio of the maximum wave height and the local depth; init: 0.8

[GAMD] controls deep water breaking (steepness breaking): the maximum wave height is equal to [GAMD]/K, where K is the local wavenumber; init: 0.88

[ALFA] factor by which the dissipation is multiplied; init: 1.0

If the keyword FREQ is not present, the program assumes that the breaking process does not influence the frequency of the waves. If FREQ is given, the change of frequency is related to the change of wave energy, by means of a formula described in part I of this report.

FRICTION [CFW] [CFC] (FREQ [A3] [B3])

Upon this command a source term due to bottom friction is added. See part I of this report. For each coefficient a default value is assumed by the program; usually it suffices to give the command: FRIC or: FRIC FREQ .

Meaning of parameters:

[CFW] Coefficient for bottom friction due to waves; init: 0.01

[CFC] Coefficient for bottom friction due to the current; init: 0.005

FREQ If this keyword is present, it is assumed that the wave frequency is influenced by the bottom friction.

[B3] It is assumed that the energy density depends on the wavenumber to the power -[B3] in the range of the higher frequencies, and it is assumed that the bottom dissipation decreases the energy density primarily in the range of the lower frequencies. Init: 3.

DIFFRACTION [ALFA]

Upon this command diffraction terms are added to the model. It is stressed that diffraction is approximated rather roughly.

A likely value of [ALFA] cannot be given, since it is restricted by the forward step size. A value of [ALFA] which is too large, may cause numerical instabilities.

NUMERIC < | UPSTREAM | -- | -> [NCOR] [CDN] [CDD] [STABN] [STABD]

This command specifies the parameters of the numerical computation.

There are two options, first the option UPSTREAM; this is a simple, relatively stable, but inaccurate scheme (the so-called upstream or upwind scheme).

The second option is the predictor-corrector scheme, or iterative Crank-Nicholson scheme. This option is default. A number of parameters can be chosen, viz.:

[NCOR] In each numerical step a number of iterations is performed. The initial value is 2. A larger value usually does not improve accuracy nor stability; a smaller value may harm the stability.

[CDN] Coefficient for the numerical diffusion in the direction normal to the computational direction. The initial value is 0. A larger value introduces numerical damping; the maximum is 1, at which value the diffusion in the numerical scheme is equivalent to that of an 'upwind' scheme.

[CDD] Coefficient for the numerical diffusion in θ-direction. The initial value is 0. A larger value introduces numerical damping; the maximum is 1, at which value the diffusion in the numerical scheme is equivalent to that of an 'upwind' scheme.

[STABN] Stability coefficient, equal to the coefficient Pn in inequality (1). Initial value: 0.7 .

[STABD] Stability coefficient, equal to the coefficient Pd in inequality (3). Initial value: 0.7 .

4.5. Output requests

There are different kinds of output commands:

1. commands defining sets of points for which one will want output.
   The following types of point sets exist: FRAME, a rectangular array of points, CURV, a set of points along a curve, POINT, a set of isolated points.
2. names and lines that can be plotted in figures.
   Names of towns or regions that must appear in the plots are defined by the command PLACE, and lines (for instance giving the contour of a certain object) are defines by the command LINES.
3. commands requesting certain output actions.
   Types of output available in HISWA are: BLOCK, a wave variable is printed for a rectangular array of points; TABLE, for each point of a set of points a number of variables is printed; PLOT, a figure is plotted for a rectangular region, containing iso-lines of a scalar quantity and/or a vector plot for a vector-type quantity.

In addition the command SHOW provides the possibility to verify the location of output frames, curves and points. The command SHOW also has options to verify the depths and currents velocities. The verification plots are made before the computation starts.

```
FRAME  'SNAME'  [XLEN]  [YLEN]  [XP]  [YP]   &
---
           [ALP]  [MX]  [MY]  [SCALE]
```

An output region with rectangular shape is defined. The name of the point set is 'SNAME'.
Its location is defined in terms of user coordinates; [XP] and [YP] are the coordinates of the origin of the frame (in m.), and [ALP] is the angle in degrees of the x-axis of the frame with respect to the x-axis of the user coordinate system.
[XLEN] and [YLEN] are the dimensions of the frame (in m.); these values are required. [MX] and [MY] are the number of meshes along each of the sides of the frame. Default: [MX]=20, [MY]=20. Restriction: if the frame is used in a PLOT or SHOW command, [MY] must be smaller than 74.
[SCALE] is the scale to which the region must be plotted (if plotting is requested), in cm on paper per m in reality. The default is such that a picture with a width (in Y-direction) of 18 cm results.

Each output point set is characterized by its name, denoted in this manual as SNAME. If one gives two definitions for the same set name, the first definition is lost.

Two output frames are generated automatically by the program, viz. the frame 'BOTTGRID', which is identical with the bottom grid, and 'COMPGRID', which is identical with the computational grid.

CURVE 'SNAME'

< [XP1] [YP1] < [INT] [XP] [YP] > >

This command defines a set of output points along a curve. Actually this curve is a broken line, defined by meand of its corner points.
The command can be used to define more than one curve.

'SNAME' name of the point set, max. 8 characters.
[XP1] [YP1] user coordinates of the first point of a curve (in m.).
[INT] number of sub-intervals (integer) between two consecutive between two corners of the curve; the program will generate [INT]-1 intermediate output points.
[XP] [YP] user coordinates of a corner point of the curve.

RAY 'RNAME' &

[XP1] [YP1] [XQ1] [YQ1] &

< [INT] [XP] [YP] [XQ] [YQ] >

Together with the command DEP (see next command) this command serves to define a set of output curves that follow a certain depth contour.

The command RAY defines a set of straight rays along which the program will attempt to find a point with a certain depth. Such a point will be added to the set of output points. Each of these rays is characterized by its end points ([XP], [YP) and ([XQ], [YQ]). Between two rays defined by the user the program will generate [INT]-1 intermediate rays.

'RNAME' name of the set of rays. The command DEP will refer to this ray set name.
[XP1] [YP1] [XQ1] [YQ1] user coordinates of the end points of the first ray.
[INT] number of subdivisions; the program will generate [INT]-1 intermediate rays.
[XP] [YP] [XQ] [YQ] user coordinates of the end points of a ray.

DEPTH 'SNAME' 'RNAME' < [DEP] >

Together with the command RAYS (see previous command) this command serves to define a set of output curves that follow

a certain depth contour.

'SNAME' name of the output curve, or the set of output curves.
'RNAME' name of the set of rays; the program will search points with the depth [DEP] along each ray in this set.
[DEP] the depth on one of the output curves. More than one values may be given. Each value leads to an output curve.

NGRID [XLEN] [YLEN] [SECTOR] [MX] [MY] [MD] &

[XPC] [YPC] [ALPC]

Defines the position and size of a nested computational grid. This command is used together with the command NEST to produce boundary conditions for a nested model. The command NGRID defines the position and size of the computational grid used in this nested model.

The meaning of the parameters in the command is entirly corresponding to those in the command GRID. However, the option of a ROTATING grid is not possible in this case.

The output points are situated on the boundaries x=0, y=0 and y=[YLEN] of this new computational grid. It is possible to make a TABLE for a point set of the type NGRID, but not a BLOCK or PLOT.

POINTS 'SNAME' < [XP] [YP] >

This command defines a set of individual output points.

'SNAME' name of the point set
[XP] [YP] user coordinates of one output point.

PLACE

< 'PNAME' [XP] [YP] [SIZE] [SIT] >

Defines name of towns, regions etc. that can be plotted in the figures produced by the program. The command PLACE can be entered more than once. Each time new names are added to the set of names.
'PNAME' is the name of a town or region within the problem area. It can be max. 16 characters long.
[XP], [YP] are the coordinates of the point of reference (in m) where the name must be plotted.

[SIZE] is the size of the characters on the plot (in cm). Default: 0.28 cm.

[SIT] an integer number indicating how the name must be placed on the plot.
1 (default): the first letter is placed at the point of reference (if the space allows). This is intended for names of towns etc.
2: the name is plotted with the point of reference in the middle (if the space allows). This is intended for names of regions, islands etc.
If the point of reference is near one of the edges of the frame, the name will be shifted as far as necessary.

LINE

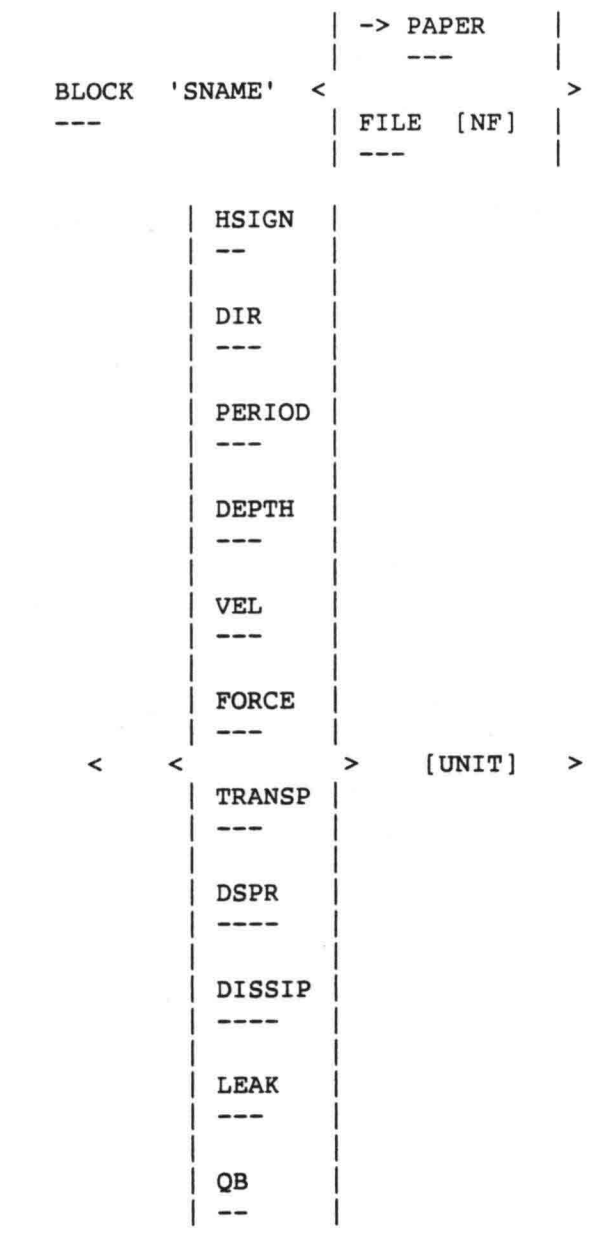
< [LINTYP] < [XP] [YP] > >

Defines one or more lines that can be plotted in the figures produced by the program. The lines consist of straight rods, of which the end points must be given in the command. The command LINES can be entered more than once. Each time new lines are added to the set of lines.
The command is provided to facilitate orientation on the plots. One can indicate coastlines, contours of certain landmarks etc. For each line the following data must be entered:

[LINTYP] indicates the type of line;
0: Heavy continuous line,
1: Thin continuous line,
2: Continuous line with cross-hatches,
4: Thin broken line.

[XP], [YP] coordinates of a corner point of the line. The number of corner points is free.

For each new line (except the first) the number [LINTYP] must appear on a new input line.



This command instructs the output program of HISWA to produce a block print or a number of block prints. The block can either be made on paper, or on a file for further processing. In the latter case [NF] is the file reference number.

The block print is made for the set of output points denoted by the point set name 'SNAME'. This point set must be of the type FRAME.

In the block print on paper only integer numbers are printed. The number printed is the value of the variable divided by [UNIT]. Choose [UNIT] small enough so that a sufficient number of significant digits remains. For each integer 5 places are available. Default: if [UNIT] is not given by the user, the program will choose a value such that the largest number occurring in the block can be printed.
The above does not apply in the case of output to a file. In that case the numbers are written to the file in real format. Now the default for [UNIT] is 1.

The block print can be made for several different variables:

HSIGN the significant wave height,

PERIOD the mean wave period

DIR the mean wave direction (in degrees, measured counterclockwise from the x-axis of the output frame); this is the direction normal to the wave crests; note that, if currents are present, this direction is different from the energy transport direction.

DSPR the directional spread of the waves (in degrees),

DEPTH the depth,

VEL the current velocity, both the x- and the y-component with respect to the frame coordinate system are printed.

FORCE the radiation stress gradient, which is equal to the resulting force exerted by the waves per unit surface; both the x- and the y-component with respect to the frame coordinate system are printed.

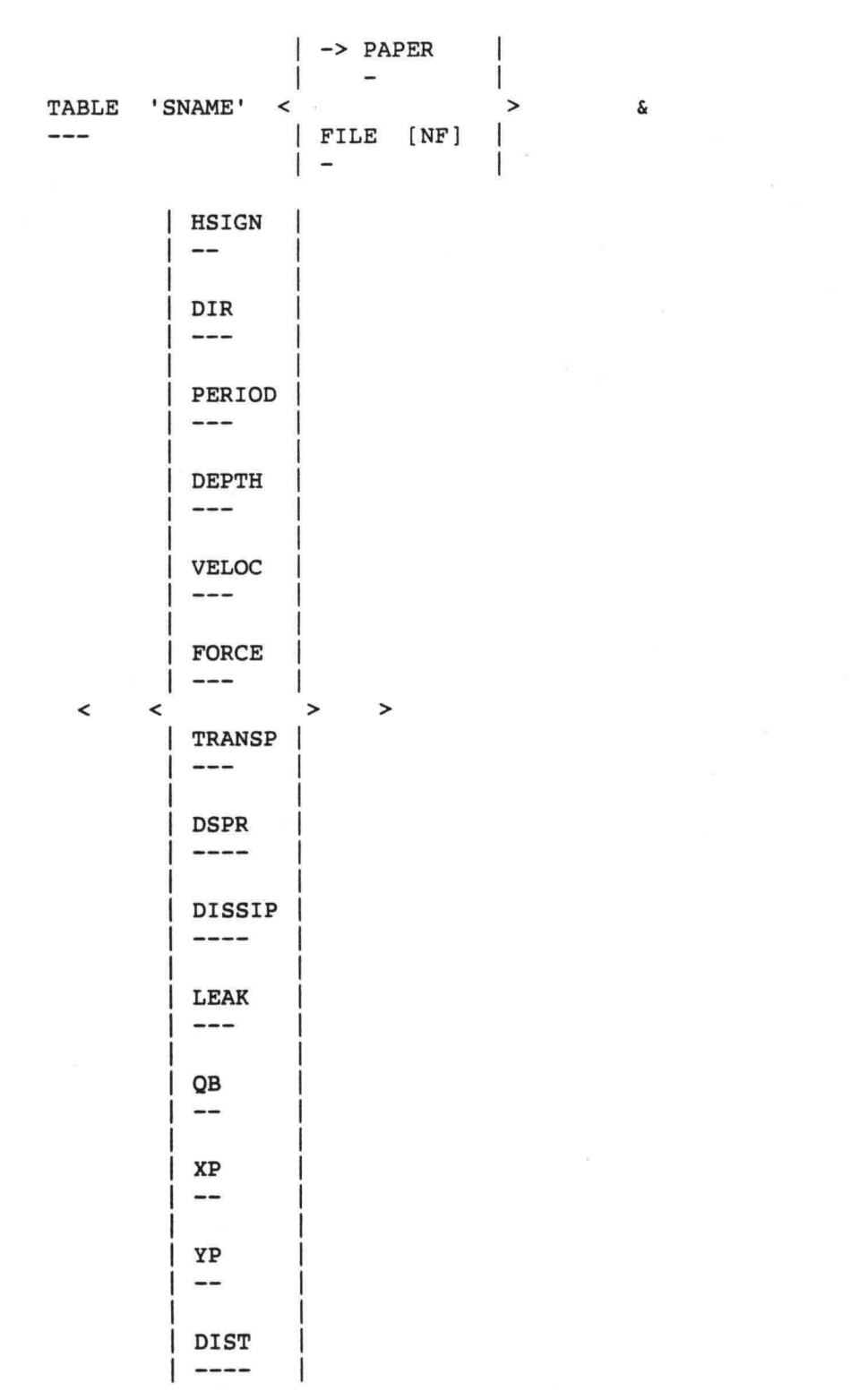
TRANSP the energy transport vector, both the x- and the y-component with respect to the frame coordinate system are printed.

TRY the energy transport in y-direction,

DISSIP the energy dissipation,

LEAK the leakage of energy over the sector boundaries,

QB the fraction of breaking waves (a parameter in the surf breaking formula).



The output module of HISWA is instructed to produce a table. The table can either be made on paper, or on a file for further processing. In the latter case [NF] is the file reference number.

The table is made for the set of output points denoted by the point set name 'SNAME'. This point set can be of any type. For each output point all different variables are written in a row.

The table can be made for several different variables; most of these are already described with the command BLOCK. The

vectorial quantities VELOCITY, FORCE and TRANSPORT are now given with respect to the user coordinate system.
Some variables are added, which indicate the output locations, and which might be needed for further processing, viz.:

XP problem x-coordinate of the output point
YP problem y-coordinate of the output point
DIST distance along an output curve (only useful if the point set is of type CURVE). For the first point the distance is 0.

PAPER [FAC] [WIDTH]

This command provides information about the plotting paper to be used.

[FAC] is an enlargement factor for the figures made by the program. In the preparation phase one can choose [FAC] smaller than 1 in order to save plot and computer cost. Init: [FAC]=1.
[WIDTH] is the paper width to be used for the figures.

PLOT 'SNAME' 'TITLE' &

HSIGN
PERIOD

DEPTH

(ISO < > [STEP] [MIN] [MAX]) &

DISSIP
LEAK

QB
--

VEL

(VEC < TRANSP > [SCALE]) &

| --- |
FORCE

(PLACES) (LINES)
--- ---

The output module of HISWA is instructed to produce a figure containing a contour line plot and/or a vector plot.

The plot is made for the set of output points denoted by the point set name 'SNAME'. This point set must be of the type FRAME. Default: SNAME='BOTTGRID'.

The 'TITLE' is plotted with the figure. If no title is given by the user, the program will generate a title.

For one scalar variable a contour plot is made, if the part ISO ... is present. The variable is characterized by a keyword (HSIGN, PERIOD, DEPTH etc.). In the description of the command BLOCK it is described which keyword indicates which physical variable.
For the contour plot one can prescribe: [STEP], the difference in value for two neighbouring contours, [MIN], the minimum value for which a contour is made, and [MAX], the maximum value for which a contour is drawn.
If [STEP] is not specified, the program will choose [MIN] in the neighbourhood of the smallest value occurring in the grid, and [MAX] in the neighbourhood of the largest value. Then [STEP] will be made roughly equal to ([MAX]-[MIN])/10.

A vector plot can be made for one of the following variables: current velocity (keyword VEL), energy transport (TRANSP), or the radiation stress gradient (FORCE).
The user can prescribe [SCALE], this is the scale to which the vectors are plotted. [SCALE] is in cm (on paper) per unit of the physical variable (for instance m/s in the case VEL). If [SCALE] is not specified, the program will assign it a value such that the arrows in the plot will not be larger than the mesh size of the output frame. The scale chosen by the program will be printed in the paper output.

If the keyword PLA is present, names of places (defined in the command PLA) will be plotted in the figure. If the keyword LIN is present, lines (defined in the command LIN) will be plotted in the figure.

```
SHOW   'SNAME'   'TITLE'        &
---

       | LOCATIONS  [SYMSIZ]                 |
       | ---                                 |
       |                                     |
     < | BOTTOM   [STEP]   [MIN]   MAX]      |  >    &
       | ---                                 |
       |                                     |
       | CURRENT   [SCALE]                   |
       | ---                                 |


       (PLACES)  (LINES)
        ---       ---
```

This command produces figures before the computation is carried out. These figures serve to facilitate the verification of input data.

The plot is made for the set of output points denoted by the point set name 'SNAME'. This point set must be of the type FRAME. Default: SNAME='BOTTGRID'.

The 'TITLE' is plotted with the figure. If no title is given by the user, the program will generate a title.

SHOW LOCATIONS : the location of output frames, curves and points is shown in the figure. [SYMSIZ] is the size of the symbols on the plot, in cm. Default: SYMSIZ=0.28 .

SHOW BOTTOM : iso-lines of the bottom level are shown; for [STEP], [MIN] and [MAX] see command PLOT.

SHOW CURRENT : a vector plot of the current velocity is made; for [SCALE] see command PLOT.

PLACES, LINES: see command PLOT.

SPEC 'SNAME' [NF]

Spectra are written onto the file with reference number [NF]. These spectra can be used as input to other programs. [NF] must preferably be between 20 and 22; see chapter 3.

NEST 'SNAME' [NF]

Spectra are written onto the file with reference number [NF]. These spectra can be used as input with the command BOUNDARY NEST. This facility is useful in the case of nesting of models, or if one wants to restart a job. The point set with name SNAME must be of the type NGRID (see command NGRID). [NF] must preferably be between 20 and 22; see chapter 3.

5. ERROR MESSAGES

Error messages issued by HISWA can be distinguished into two categories: first the messages printed by the input program HISWA/PREP, and second the messages printed during the computation by the computation program HISWA/COMPU.

The messages issued by HISWA/PREP are meant to be self-explanatory. Messages by HISWA/COMPU are printed if the results of the computation do not conform to certain conditions. The program tests for the following conditions:
- whether action density is positive for each spectral direction;
- whether the average frequency is positive for each spectral direction;
- whether the stability condition for DΘ is fulfilled;
- whether the stability condition for DY is fulfilled.

Due to numerical inaccuracies or instability negative action densities or frequencies may occur. A large number of such occurrences is an indication for a deterioration of the quality of the results.

If the stability condition for DΘ is not fulfilled (see inequality (3), the value of the slope is decreased locally, such that inequality (3) is fulfilled again. This is necessary for stability, but obviously it will deteriorate the accuracy of the results, if it happens frequently.
Likewise, if the stability condition for DY is violated, the component of the propagation velocity in y-direction is reduced. Since the computational grid is already checked for inequality (2), the message 'C-y corrected' can only occur in computations with current.