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

The shallow water wave hindcast model HISWA Part IV: system documentation

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

Rep. No. 9-85

CT-III lab Vloeistofmech
E. Lute K004

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 IV contains the system documentation of the HISWA model
Status of report	confidential, final report, part IV
City/date:	Delft, February 20, 1985

Part IV. HISWA system documentation

date of printing: January 11, 1985

TABLE OF CONTENTS

1. INTRODUCTION

1.1 General characteristics of the model

In this document the system documentation of a numerical shallow water waves hindcast model named HISWA is presented. This model is expected to provide realistic estimates of the wave conditions in coastal areas. It is a directionally decoupled parametric model containing bottom refraction, wave growth, dissipation due to wave breaking (surf zone) and bottom friction as well as a simple representation of diffraction effects. Further the effects of currents on refraction, wind generation, bottom friction and wave breaking in deep water is included.

For the mathematical formulation of this model reference is made to Holthuijsen and Booij (1983) and to part I of this report.

Two balance equations in the parameters A0 (wave action integrated over frequency) and W0 (wave frequency averaged over action), containing gradients in three dimensions x, y and θ (wave direction), are solved :

$$\frac{@}{@X}(CX0.A0)+\frac{@}{@Y}(CY0.A0)+\frac{@}{@\theta}(C\theta 0.A0+Cdif(CY0.\frac{@}{@X}A0 -CX0.\frac{@}{@Y}A0))=\frac{S0}{Wr}-\frac{A0}{W0}.\frac{d}{dT}W0 \qquad (1)$$

$$\frac{@}{@X}(CX0.W0.A0)+\frac{@}{@Y}(CY0.W0.A0)+\frac{@}{@\theta}(C\theta 0.W0.A0 +Cdif(CY0.\frac{@}{@X}(W0.A0)-CX0.\frac{@}{@Y}(W0.A0)))=\frac{W0.S0}{Wr} \qquad (2)$$

in eq. (1) and (2) :

CX0, CY0, Cθ0 are the components in X, Y resp. θ direction of wave action transport velocity

Wr is the relative frequency

S0 is the source term including wind generation, bottom friction and wave breaking (surf zone and deep water)

Cdif is a diffraction coefficient

(expressions for these terms are given in chapter 3)

A numerical grid is defined in three dimensions x, y and θ (fig. 1). The direction of wave propagation θ is defined as the angle between the wave number vector and the positive x-axis, measured counter-clockwise from the direction of the positive x-axis.

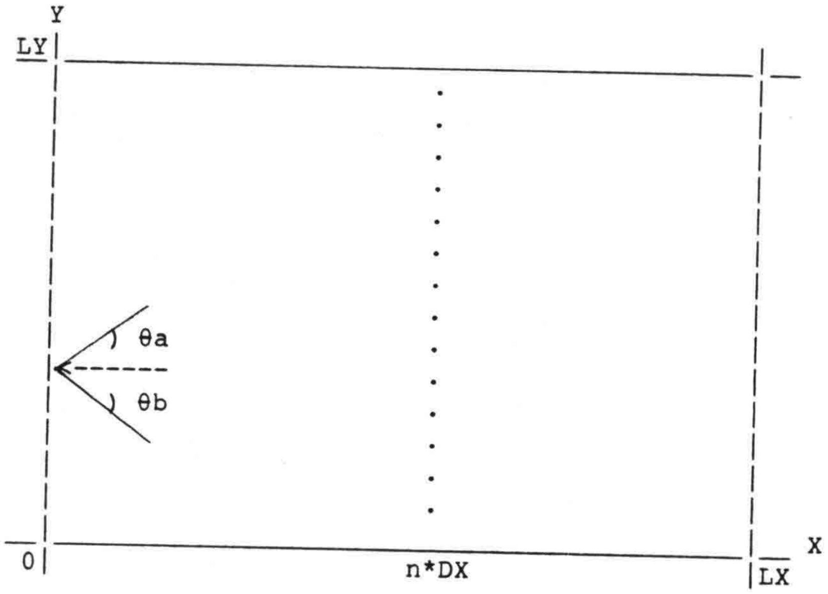

fig. 1 the computational region

The computation progresses in the positive x-direction and propagation of wave energy is limited to a sector defined by θa and θb around the direction of the x-axis (both |θa| and |θb| < 90 degrees). The computations are carried out line by line with an explicit predictor-corrector scheme. The number of corrector steps is free but two steps are sufficient to obtain a stable scheme. Lines are defined parallel to the y and θ axis.

Beside the computational grid described above at least one other grid is used in the model HISWA, the bottom grid containing the bottom topography and current field (in x-y plane).
The location of both grids and of other points are described by means of a common system of coordinates, known as the user coordinates.

1.2. Computer programs

The model HISWA consists of three computer programs :

PREP input preparation and control part
COMPU computational part
OUTP output of results

In the following chapters these three programs will be described. The programs HISWA/PREP and HISWA/OUTP are adjusted versions of the corresponding programs PREP and UITV of the refraction/diffraction model CREDIZ of Rijkswaterstaat (CREDIZ01 system documentation, 1984).

1.3. Documentation

Descriptions of the computer programs and of their subroutines are given in chapters 2 through 5.
In each chapter the structure of one of the programs is explained. The relations between the subroutines are shown in block diagrams. An example of a block diagram is given below.

```
          A
          |
          |___ B
          |    |
          |    |___ C
          |
          |___ D
               |
               |...
```

fig. 2 block diagram

In the program or subrutine A, subroutines B and D are called. Further subroutine C is called in subroutine B. The symbol shown under subroutine D means that the subroutines called by D are found in a separate diagram further on in the text. Some of the subroutines are called very often, notably STRACE, COPYCH and MSGERR. These are not mentioned in the structure diagrams.

The structure of a single program or subroutine is presented in structural diagrams. For convenience the conventional construction in the left part of figure 3 is sometimes replaced by the one on the right or in the lower part of the figure.

```
|      if .........         |     | if ... then          |
| T                       F |     |   :------------------|
|--------------:------------|     |   | .....            |
| ......       | .......    |     |   :------------------|
|              |            |     | else                 |
|              |            |     |   :------------------|
|              |            |     |   | .....            |

or      ------------------------------------------
        If .....
        Then ...
             ...
        Else ...
             ...
        ------------------------------------------
```

fig. 3 representation of a conditional statement

Descriptions of the various subroutines are given in chapters 2 to 5. The sequence in which the subroutines are discussed, corresponds with the place in the program COMPU at which the subroutine is called for the first time. Parameter lists of the subroutines are described in which input- and output parameters are denoted by (I) resp. (O).
In chapter 6 the storage of variables and arrays in common blocks and files is described. A flexible handling of computer storage, necessary for the considerable number of arrays in the programs is obtained through the application of a Dynamic Data Pool.
In this report reference is made to the CREDIZ01 system documentation (1984) for a detailed description, in so far as subroutines and other facilities of the model CREDIZ are implemented in the present model.

2. PREPARATION PROGRAM HISWA/PREP

2.1. Structure of the program

```
HISWA/PREP
    |
    |___VERSIE               generation date is printed
    |
    |___OPENF                open all necessary files
    |
    |___STARTW               initialisation of common variables
    |     |
    |     |___INPOOL         initialization of the dynamic data pool
    |     |     |
    |     |     |___REQDA          expansion of the dynamic data pool
    |     |
    |     |___ADPOOL         reduction or expansion of a pool array
    |           |
    |           |___REQDA          expansion of the dynamic data pool
    |
    |___REDEP                reads bottom and current data
    |     |
    |     |___ADPOOL
    |           |
    |           |___REQDA
    |
    |___PROUT               processes output commands
    |     |
    |     |...              see further down
    |
    |___PLOTX               produces verification plots
    |     |
    |     |...              see further down
    |
    |___BRTRAF              computes transformation coefficients
    |
    |___PRCON               performs checks on input
    |     |
    |     |___INUPT
    |           |
    |           |___INBOD   determines whether point is in bottom grid
    |           |
    |           |___INREK   determines whether point is in comp. grid
    |
    |___WRCOM             common blocks are written to file INSTR
    |     |
    |     |...            see further down
    |
    |___WRBC              interpolates depth and current to computational
          |               grid and writes result to file INSTR
          |
          |___INPDC       determines depth and current in one point
```

diagram of subroutines called by PROUT

```
PROUT
  |
  |___NEWUR     initializes a new output request record
  |     |
  |     |___ADPOOL
  |           |
  |           |___REQDA
  |
  |___ADDUR     adds one number to output request record
  |     |
  |     |___ADPOOL
  |           |
  |           |___REQDA
  |
  |___ODLOC     searches for an output point set
  |
  |___ODAPT     updates output data
  |     |
  |     |___ODSHIF     shifts output data
  |           |
  |           |___ADPOOL
  |                 |
  |                 |___REQDA
  |
  |___ODDELR    deletes an output point set
  |     |
  |     |___ODDELP    deletes an output data record
  |
  |___ODEXP     adds data to output data record
        |
        |___ODSHIF
              |
              |___ADPOOL
                    |
                    |___REQDA
```

diagram of subroutines called by PLOTX

```
PLOTX
  |
  |___PLOSIT    produces location plot
  |     |
  |     |___PLNAME        plots a name
  |     |
  |     |___PENTO         moves pen to a point
  |     |
  |     |___PLYN          plots a line
  |           |
  |           |___SNYPT1   computes crossing point
  |           |
  |           |___SNYPT2   computes crossing point
  |
  |___PLODEP    plots bottom level
  |     |
  |     |___INPDC     computes depth in a point
  |
  |___PLSTRM    plots current velocities
  |     |
  |     |___INPDC     computes velocity in a point
  |
  |___PNAMES    plots geographical names
  |     |
  |     |___PLNAME
  |
  |___PCOAST    plots (coast)lines
        |
        |___PENTO         moves pen to a point
        |
        |___PLYN          plots a line
              |
              |___SNYPT1   computes crossing point
              |
              |___SNYPT2   computes crossing point
```

diagram of subroutines called by WRCOM and WRCOMX

```
WRCOM
  |
  |___WRPOOL    reads or writes one pool array
  |     |
  |     |___ADPOOL
  |           |
  |           |___REQDA
  |
  |___WRDUMP    prints the contents of an array
```

The scheme for the subroutine WRCOMX is identical.

2.2. Main program

The main program of HISWA/PREP processes most of the user's commands, except the output requests. Interpretation of the main program is easiest with help of the description of the commands in the User's Manual (part III of this report).

2.3. Subroutine VERSIE

A message is printed giving the date of generation of the present version of the model HISWA.

2.4. Subroutine WRBC

Function :
In this subroutine depths D(Y) and current velocity components UX(Y), UY(Y) are determined at a line in the computational grid. The result is written to file INST.

Method :
For every grid point, bottom grid coordinates are computed and depth and current velocity components are determined through bilinear interpolation.

Structure :

```
    *WRBC*
    :--------------------------------------------------------:
    | For every x do                                         |
    |    :---------------------------------------------------|
    |    | For every y do                                    |
    |    |    :----------------------------------------------|
    |    |    | determine bottom grid coordinates of point   |
    |    |    |----------------------------------------------|
    |    |    | CALL INPDC determine depth and current       |
    |    |    |                                    in point  |
    |    |    |----------------------------------------------|
    |    |    | if current is on then                        |
    |    |    |    :-----------------------------------------|
    |    |    |    | determine current relative to           |
    |    |    |    |                    computational grid   |
    |    |---------------------------------------------------|
    |    | write depths to file with no. NREF                |
    |    |---------------------------------------------------|
    |    | If current is on then                             |
    |    |    :----------------------------------------------|
    |    |    | Write Ux and Uy values to file NREF          |
    :--------------------------------------------------------:
```

The call of this subroutine is :
    CALL WRBC(NREF)

parameter :
 NREF (I) file reference number of file to which depths and current components are written.

2.5. Subroutine INPDC

Function :
Depth and current velocity components are computed in a point given in bottom grid coordinates (IB,JB).

Method :
A bilinear interpolation is carried out with the surrounding points in the bottom grid. If point (IB,JB) is located outside the bottom grid then a constant depth and no current is assumed.

Structure :

```
   *INPDC*
   :-------------------------------------------------------:
   | if point is located in bottom grid then               |
   |    :--------------------------------------------------|
   |    | compute depth                                    |
   |    |--------------------------------------------------|
   |    | if depth is positive and current is on then      |
   |    |    :---------------------------------------------|
   |    |    | compute current components                  |
   |    |    :---------------------------------------------|
   |    | else                                             |
   |    |    :---------------------------------------------|
   |    |    | current is 0.                               |
   |    :--------------------------------------------------|
   | else                                                  |
   |    :--------------------------------------------------|
   |    | depth is constant value outside bottom grid      |
   |    |                               and current is 0.  |
   :-------------------------------------------------------:
```

The call of this subroutine is :

```
     CALL INPDC
```

3. COMPUTATION PROGRAM HISWA/COMPU

3.1. Structure of the program

The computational part COMPU of the model HISWA carries out the computation proper. It gets its instructions from HISWA/PREP and it transmits its results to HISWA/OUTP.

A block diagram showing the relations between the various subroutines of COMPU is given in fig. 3.1. Separate diagrams for the subroutines WRCOM, WRCOMX, WAVPA and TERMD are included in fig. 3.2 through 3.5.

```
  COMPU
    |
    |___OPENF              open all necessary files
    |
    |___INPOOL             initialization of the dynamic data pool
    |     |
    |     |___REQDA        expansion of the dynamic data pool
    |
    |___WRCOM              reads common blocks from file INSTR
    |     |
    |     |...             see section 2.1
    |
    |___WRCOMX             writes common blocks to file REKRES
    |     |
    |     |...             see section 2.1
    |
    |___ADPOOL             reduction or expansion of a pool array
    |     |
    |     |___REQDA        expansion of the dynamic data pool
    |
    |___DIVKOP             a title is written above the output
    |
    |___STARTB             computation of wave parameters at the
    |     |                boundary x=0
    |     |
    |     |___DISTR        computation of directional distribution
    |     |
    |     |___WAVPA        computation of wave parameters at a line
    |     |     |          in the computational grid
    |     |     |...
    |     |
    |     |___WRIRE        results of computations are written to a file
    |
    |___NUMSC              computation of wave parameters at a new line
          |
          |___PREDT        wave action and frequency are computed throug
          |                linear extrapolation from the former lines
          |
          |___DISPA        computation of parameters in source terms
          |     |
          |     |___FRABRE the fraction of breaking waves is computed
          |
          |___TERMD        the terms of the two balance equations
          |     |          are computed in a grid point
          |     |...
          |
          |___SUMDE        evaluation of balance equations yields the
          |                wave action and -frequency in a grid point
          |
          |___WAVPA        computation of wave parameters at a line
          |     |          in the computational grid
          |     |...
          |
          |___WRIRE        results of computations are written to a file
```

fig. 3.1. relations between subroutines in COMPU

```
WAVPA
  |
  |___ITWN              wave number and relative frequency are
  |                     computed in a grid point
  |
  |___VWPRO             propagation velocity components are
                        computed in a grid point
```

fig. 3.2. diagram of subroutines called by WAVPA

```
TERMD
  |
  |___TRSY              divergence of transport in y-direction
  |
  |___TRST              divergence of transport in θ-direction
  |
  |___DIFT              diffraction terms
  |
  |___SWIND             wind generation source terms
  |
  |___SBOT              bottom dissipation source terms
  |
  |___SSURF             surf breaking source terms
  |
  |___SDBR              deep water breaking source terms
```

fig. 3.3. diagram of subroutines called by TERMDE

3.2. Main program

Input is read from a file named INSTR and output is written to a file named REKRES (section 6.3). The diagram of the main program is shown below.

```
* HISWA/COMPU *
:-------------------------------------------------------:
| CALL OPENF open all necessary files                   |
|-------------------------------------------------------|
| read dimension of pool and testparameter              |
|                                        from file INSTR |
|-------------------------------------------------------|
| CALL INPOOL initialize the pool and fill              |
|                                  it with empty arrays |
|-------------------------------------------------------|
| CALL WRCOM read common blocks and pool                |
|                                arrays from file INSTR |
|-------------------------------------------------------|
| CALL WRCOMX write common blocks and pool              |
|                                arrays to file REKRES  |
|-------------------------------------------------------|
| CALL ADPOOL enlarge the dimensions of pool arrays     |
|-------------------------------------------------------|
| CALL DIVKOP write a title                             |
|-------------------------------------------------------|
| CALL STARTB compute wave parameters on line           |
|            x=0 and write the results to file REKRES   |
|-------------------------------------------------------|
| for every line do                                     |
|     :-------------------------------------------------|
|     | CALL NUMSC compute wave parameters on this      |
|     |        line and write the results to file REKRES |
:-------------------------------------------------------:
```

INPUT :
In the program COMPU instructions, definitions of grids, coefficients etc. (formulated in PREP) and arrays containing bottom topography and current field are read from a file.
PROCEDURE :
First the values of the frequency integrated action A0 and mean action frequency W0 are determined at the boundary x=0. Further depths, currents, wave numbers and wave propagation velocity components are computed at this boundary. For every new line (x=n*dx) A0 and W0 are obtained through the application of the numerical scheme described in section 1.1.
OUTPUT :
For every line the following results are written to the file REKRES that will be read by the program HISWA/OUTP :

wave action, absolute frequency, relative frequency, group velocity, wave number and components of wave action transport velocity (in every grid point in the y-θ plane)

leakage of energy through the θ boundary (for every value of y)

dissipation of energy due to bottom friction, surf- and deep water breaking (for every value of y)

the fraction of breaking waves (for every value of y)

3.3. Subroutine STARTB

Function :
In the subroutine STARTB the wave conditions at the boundary

x=0 are determined. Further the directional distribution of waves in an ideal wind field with given wind direction is determined.

Method :
If necessary the wave action A0(y,θ) and average frequency W0(y,θ) are read from the file NEST.
Depths and currents are determined at the boundary x=0.
Wave number, relative frequency, group velocity and components of propagation velocities are computed in every grid point in the Y-θ plane.
Next these parameters are written to the file REKRES.

Structure :

```
   *STARTB*
   :---------------------------------------------------:
   | determine wind direction relative to the           |
   |                             computational grid     |
   |----------------------------------------------------|
   | CALL DISTR compute the directional                 |
   |                           distribution of waves    |
   |----------------------------------------------------|
   | If IINC=2 then                                     |
   |    :-----------------------------------------------|
   |    | read wave action and frequency from the file  |
   |    |                                       LFNEST  |
   |----------------------------------------------------|
   | IF NBOUR=1 then                                    |
   |    :-----------------------------------------------|
   |    | retain right boundary condition in arrays     |
   |    | IABR and IFBR                                 |
   |----------------------------------------------------|
   | IF NBOUL=1 then                                    |
   |    :-----------------------------------------------|
   |    | retain left boundary condition in arrays      |
   |    | IABL and IFBL                                 |
   |----------------------------------------------------|
   | CALL WAVPA determine wave parameters               |
   |                               at boundary x=0      |
   |----------------------------------------------------|
   | CALL WRIRE write wave parameters to file REKRES    |
   :---------------------------------------------------:
```

The call of this subroutine is :
    CALL STARTB

3.4. Subroutine DISTR

Function :
In subroutine DISTR the directional distribution of wave action is computed.

Method :
For the directional distribution a cos**COEF distribution is chosen. The parameter COEF is supplied by the user.

$$B(\theta)=c.\cos^{COEF}(\theta-DIR).A(Y) \quad \text{for } |\theta-DIR|<90 \text{ deg.}$$

$$=0 \quad \text{for } |\theta-DIR|\geq 90 \text{ deg.}$$

where DIR is the mean wave or wind direction.

The call of this subroutine is :

```
     CALL DISTR(DIR,COEF,NORM)
```

Parameters :

DIR (I)	mean wave or wind direction
COEF (I)	parameter of directional distribution
NORM (I)	= 1 the normalization coeff. c is computed = 0 c is assumed to be known

3.5. Subroutine WAVPA

Function :
In this subroutine wave numbers K0, relative frequency Wr, group velocity CG0 and propagation velocity components CX0, CY0 and Cθ0 are computed at a line IX in the computational grid.

Method :
In order to evaluate these parameters first the depths D and current velocity components UX, UY at line IX are determined.

Structure :

```
   *WAVPA*
   :-----------------------------------------------------:
   | if predictor is passed or line is boundary          |
   |                                            x=0 then |
   |   :-------------------------------------------------|
   |   | if predictor is passed then                     |
   |   |   :---------------------------------------------|
   |   |   | move arrays containing depths and currents  |
   |   |   | at line IX to arrays with old values        |
   |   |-------------------------------------------------|
   |   | read depth and current values from file INSTR   |
   |-----------------------------------------------------|
   | for every y do                                      |
   |   :-------------------------------------------------|
   |   | for every θ do                                  |
   |   |   :---------------------------------------------|
   |   |   | CALL ITWN calculate wave number and         |
   |   |   |                          relative frequency |
   |   |   |---------------------------------------------|
   |   |   | CALL VWPRO calculate group velocity and     |
   |   |   |         components of propagation velocity  |
   :-----------------------------------------------------:
```

The call of this subroutine is :

```
     CALL WAVPA(IX)
```

parameter :

IX (I) line in computational grid at which wave parameters are computed (X=(IX-1)DX)

3.6. Subroutine ITWN

Function :
In ITWN the wave number K0 and relative frequency Wr in a grid point (IX,IY,IT) is determined.

Method :
The current component in the direction of wave propagation is determined.

$$U = Ux.\cos\theta + Uy.\sin\theta \qquad (1)$$

If U > 0 then the wave number K0 is computed through a Newton-Raphson iteration process, applied to eq. (2).

$$F \equiv W0-K0(UX.\cos\theta+UY.\sin\theta)-(g.K0.\tanh(K0.D))^{1/2} = 0 \qquad (2)$$

This procedure requires an estimate of the wave number K0 as a start value. Here the value of K0 on line IX-1 is used as an estimate for K0. At the boundary x=0 the following approximation of K0 is applied.

$$\frac{g.K0}{W0^{2}} = \frac{1}{(\tanh(W0^{2}.D/g))^{1/2}} \qquad (3)$$

The relative frequency Wr is calculated with eq. (4).

$$Wr=(g.K0.\tanh(K0.D))^{1/2} \qquad (4)$$

The group velocity CG0 is computed:

$$CG0 = Wr\left(\frac{1}{2.K0} + \frac{D}{\sinh(2.K0.D)} \right) \qquad (5)$$

If U < 0 first the frequency Wc is determined which is the highest frequency capable to transport wave energy against the current U. For this purpose eq. (6) is solved through a Newton-Raphson iteration process.

$$G \equiv U + Wc'.\left(\frac{1}{2.K1} + \frac{D}{\sinh(2.K1.D)} \right) =0 \qquad (6)$$

with

$$Wc' = (g.K1.\tanh(K1.D))^{1/2} \qquad (7)$$

Wc is given by

$$Wc = K1.U + Wc' \qquad (8)$$

If W0 < Wc (eq. (2) has a solution) then K0 and Wr are computed with the procedure described above. If W0 > Wc (eq. (2) has no solution) then K1 and Wr' are used as estimates of K0 resp. Wr.

A source term Sdbr is introduced to dissipate wave components with frequency > Wc (section 3.19).
For points outside the bottom grid K0 is computed with eq. (9).

$$K0=W0^2/g \qquad (9)$$

If a negative depth is encountered then K0 and Wr are given the values -1.0 resp. 0.0.

Structure :

```
  *ITWN*
  :-----------------------------------------------------:
  | if point is located outside bottom grid then        |
  |   :-------------------------------------------------|
  |   | compute wave number and relative frequency      |
  |   :-------------------------------------------------|
  | else if depth is negative then                      |
  |   :-------------------------------------------------|
  |   | wave number is -1.,relative frequency is 0.     |
  |   :-------------------------------------------------|
  | else                                                |
  |   :-------------------------------------------------|
  |   | give an estimate for the wave number Ke         |
  |   |-------------------------------------------------|
  |   | compute current component in direction of       |
  |   |                           wave propagation  U   |
  |   |-------------------------------------------------|
  |   | if U < 0 then                                   |
  |   |   :---------------------------------------------|
  |   |   | compute estimate for wave number Kl         |
  |   |   |---------------------------------------------|
  |   |   | compute function G                          |
  |   |   |---------------------------------------------|
  |   |   | for i= 1 to 50 while G > accuracy do        |
  |   |   |   :-----------------------------------------|
  |   |   |   | compute derivative of G                 |
  |   |   |   |-----------------------------------------|
  |   |   |   | compute wave number Kl                  |
  |   |   |   |-----------------------------------------|
  |   |   |   | compute function G                      |
  |   |   |---------------------------------------------|
  |   |   | compute critical frequency                  |
  |   |   :---------------------------------------------|
  |   | else                                            |
  |   |   :---------------------------------------------|
  |   |   | critical frequency is -1.                   |
  |   |-------------------------------------------------|
  |   | if critical frequency < 0 or > wave frequency   |
  |   |   :---------------------------------------------|
  |   |   | compute function F                          |
  |   |   |---------------------------------------------|
  |   |   | for i = 1 to 50 while F > accuracy do       |
  |   |   |   :-----------------------------------------|
  |   |   |   | compute derivative of F                 |
  |   |   |   |-----------------------------------------|
  |   |   |   | compute wave number Ke                  |
  |   |   |   |-----------------------------------------|
  |   |   |   | recompute function F                    |
  |   |   |   :-----------------------------------------|
  |   |   | else                                        |
  |   |   |   :-----------------------------------------|
  |   |   |   | wave number is Kl                       |
  |   |   |---------------------------------------------|
  |   |   | compute group velocity                      |
  :-----------------------------------------------------:
```

The call of this subroutine is :

```
     CALL ITWN(IX,IY,IT)
```

Parameters :

```
     IX (I)
             )
     IY (I)    coordinates of point in computational grid
             )
     IT (I)
```

3.7. Subroutine VWPRO

Function :
In subroutine VWPRO the components of wave action transport velocity CX0, CY0 and Cθ0 are determined in a point (IX,IY,IT) in the computational grid.

Method :
The relations for the parameters mentioned above used in this model are

$$CX0 = CG0.\cos\theta + UX \qquad (1)$$

$$CY0 = CG0.\sin\theta + UY \qquad (2)$$

$$C\theta 0 = -\frac{Wr}{\sinh(2.K0.D)}\left(-\sin\theta\frac{\partial D}{\partial X}+\cos\theta\frac{\partial D}{\partial Y}\right)-\cos\theta\left(-\sin\theta\frac{\partial UX}{\partial X}+\cos\theta\frac{\partial UX}{\partial Y}\right)$$

$$-\sin\theta\left(-\sin\theta\frac{\partial UY}{\partial X}+\cos\theta\frac{\partial UY}{\partial Y}\right) \qquad (3)$$

The terms containing current velocity components UX and UY in eq. (1) through (3) are omitted if no current is present. The term Cθ0 is evaluated at intermediate lines between IX and IX+1. Derivatives of depth and current are determined through a central difference scheme (after the predictor step).
If negative depths are encountered then all velocity components are given the value 0..

Structure :

```
   *VWPRO*
   :--------------------------------------------------:
   | if depth is negative then                        |
   |    :---------------------------------------------|
   |    | give velocity components and derivatives    |
   |    |                              the value 0.   |
   |    :---------------------------------------------|
   | else                                             |
   |    :---------------------------------------------|
   |    | if predictor step is passed then            |
   |    |    :----------------------------------------|
   |    |    | compute depth derivatives and current  |
   |    |    |                            derivatives |
   |    |---------------------------------------------|
   |    | compute components of wave action transport |
   |    |                                    velocity |
   |    | (Cθ0 only if line ≠ boundary x=0)           |
   :--------------------------------------------------:
```

The call of this subroutine is :

```
     CALL VWPRO(IX,IY,IT)
```

Parameters :

```
     IX (I)
             )
     IY (I)    coordinates of point in computational grid
             )
     IT (I)
```

3.8. Subroutine NUMSC

Function :
In the subroutine NUMSC wave parameters are computed at a new line IX+1 in the computational grid.

Method :
The following numerical scheme is applied :
Estimates for the wave action A0 and -frequency W0 at line IX+1 are obtained through a linear extrapolation from the lines IX-1 and IX (predictor step). With these estimates the other wave parameters at line IX+1 can be determined. Linear interpolation between the lines IX and IX+1 yields the wave parameters at line IX+1/2, necessary for the corrector step. The corrector step (which can be repeated several times) consists of an explicit differential scheme,applied to the two balance equations described in section 1.1.
The amount of energy lost through dissipation (FD) and leakage of energy through the boundaries θa and θb (FL) is kept.
Results of the computations are written to the file REKRES.

Structure :

```
   *NUMSC*
   :-----------------------------------------------------:
   | CALL PREDT predictor estimates for wave action      |
   |                        and -frequency on line IX+1  |
   |-----------------------------------------------------|
   | move contents of arrays with new values of wave     |
   | number,relative and critical frequency and propaga- |
   | tion velocity components to arrays with old values  |
   |-----------------------------------------------------|
   | CALL WAVPA compute wave parameters at line IX+1     |
   |-----------------------------------------------------|
   | determine depths and currents                       |
   |                     intermediate lines IX and IX+1  |
   |-----------------------------------------------------|
   | for every corrector step do                         |
   |   :-------------------------------------------------|
   |   | determine wave parameters intermediate lines    |
   |   |                                     IX and IX+1 |
   |   |-------------------------------------------------|
   |   | CALL DISPA compute parameters in dissipation    |
   |   |                                           terms |
   |   |-------------------------------------------------|
   |   | for every y do                                  |
   |   |   :---------------------------------------------|
   |   |   | if last corrector step is in progress then  |
   |   |   |   :-----------------------------------------|
   |   |   |   | initialize leakage and dissipation      |
   |   |   |   |                         in point x,y    |
   |   |   |---------------------------------------------|
   |   |   | if depth is positive then                   |
   |   |   |   :-----------------------------------------|
   |   |   |   | for every θ do                          |
   |   |   |   |   :-------------------------------------|
   |   |   |   |   | CALL TERMD compute terms of the     |
   |   |   |   |   |             two balance equations   |
   |   |   |   |   |-------------------------------------|
   |   |   |   |   | CALL SUMDE determine wave action    |
   |   |   |   |   |                     and frequency   |
   |   |   |   |-----------------------------------------|
   |   |   |   | if last corrector step in progress then|
   |   |   |   |   :-------------------------------------|
   |   |   |   |   | compute leakage and dissipation in  |
   |   |   |   |   |                         point x,y   |
   |   |-------------------------------------------------|
   |   | CALL WAVPA compute wave parameters on line IX+1|
   |-----------------------------------------------------|
   | CALL WRIRE write results of line IX+1 to REKRES     |
   :-----------------------------------------------------:
```

The call of this subroutine is :
    CALL NUMSC(IX)

Parameter :
    IX (I) wave parameters are determined at line IX+1
            in the computational grid (X=IX.DX)

3.9. Subroutine PREDT

Function :
Estimates for the wave action A0 and absolute frequency W0 at line IX+1 (predictor step) are determined in this subroutine.

Method :
The predictor is a simple extrapolation procedure. A0 and W0 at line IX+1 are determined as follows :

$$A0^{IX+1} = 2.A0^{IX} - A0^{IX-1} \qquad (1)$$

$$W0^{IX+1} = 2.W0^{IX} - W0^{IX-1} \qquad (2)$$

If a negative depth is encountered then A0 and W0 are given the value 0..
If the estimate of the wave height in a grid point (x,y), obtained thus, exceeds the local maximum wave height (gamma*d) then the predictor estimates of A0 in this point are reduced accordingly.

Structure :

```
*PREDT*
:-----------------------------------------------------:
| if line is boundary X=0 then                        |
|   :-------------------------------------------------|
|   | wave action and -frequency on new line are      |
|   |              given the values on the old line   |
|   :-------------------------------------------------|
| else                                                |
|   :-------------------------------------------------|
|   | for every y do                                  |
|   |   :---------------------------------------------|
|   |   | if depth is positive then                   |
|   |   |   :-----------------------------------------|
|   |   |   | for every θ do                          |
|   |   |   |   :-------------------------------------|
|   |   |   |   | move wave action and -frequency     |
|   |   |   |   | to arrays with old values and       |
|   |   |   |   | compute new values                  |
|   |   |   :-----------------------------------------|
|   |   | else                                        |
|   |   |   :-----------------------------------------|
|   |   |   | for every θ do                          |
|   |   |   |   :-------------------------------------|
|   |   |   |   | move wave action and -frequency     |
|   |   |   |   | to arrays with old values and       |
|   |   |   |   | give new values the value 0.        |
|   |-------------------------------------------------|
|   | If NBOUR=2 then                                 |
|   |   :---------------------------------------------|
|   |   | Read boundary data into pool arrays         |
|   |   |   IABR and IFBR                             |
|   |-------------------------------------------------|
|   | If NBOUL=2 then                                 |
|   |   :---------------------------------------------|
|   |   | Read boundary data into pool arrays         |
|   |   |   IABL and IFBL                             |
|-----------------------------------------------------|
| if surf breaking is on then                         |
|   :-------------------------------------------------|
|   | for every y do                                  |
|   |   :---------------------------------------------|
|   |   | compute wave energy in grid point x,y       |
|   |   |---------------------------------------------|
|   |   | if wave energy exceeds local maximum        |
|   |   |                           wave energy then  |
|   |   |   :-----------------------------------------|
|   |   |   | for every θ do                          |
|   |   |   |   :-------------------------------------|
|   |   |   |   | reduce wave action in grid point    |
|   |   |   |   |                              x,y,θ  |
:-----------------------------------------------------:
```

The call of this subroutine is :
    CALL PREDT(IX)

Parameter :
    IX (I) wave parameters are determined at line IX+1
            in the computational grid (X=IX.DX)

3.10. Subroutine DISPA

Function :
In this subroutine parameters at line IX+1/2 are determined, necessary for the evaluation of the dissipation terms in the two balance equations.

Method :
The following parameters are determined :

orbital velocity at the bottom Ubot
wave energy density Et
local maximum wave height Hm
the fraction of breaking waves Qb

For these parameters the following relations are used :

$$Ubot = \left(D\theta . \sum_{\theta} \frac{Wr.A0^{3}}{\sinh^{2} (K0.D)} \right)^{1/2} \qquad (1)$$

$$Et = D\theta . \sum_{\theta} Wr.A0 \qquad (2)$$

$$Hm = Gamd.\overline{K0}^{-1} .\tanh(Gams.\overline{K0}.D/Gamd) \; , \; \overline{K0} = \frac{1}{N\theta} . \sum_{\theta} K0 \qquad (3)$$

(the coefficients Gams and Gamd are given by the user)

$$\frac{1-Qb}{\ln Qb} = -8.\frac{Et}{Hm^{2}} \quad \text{(evaluated in FRABRE)} \qquad (4)$$

Structure :

```
*DISPA*
if bottom dissipation is on then
    for every y do
        compute orbital velocity at the bottom
if surf breaking is on then
    for every y do
        compute wave energy density
        compute local maximum wave height
        CALL FRABRE compute fraction of breaking waves
```

The call of this subroutine is :

```
    CALL DISPA
```

3.11. Subroutine FRABRE

Function :
In this subroutine the fraction of breaking waves in a point x,y in the computational grid (Qb) is computed.

Method :
The fraction of breaking waves is given by the implicit relation :

$$F \equiv 1-Qb+8.\frac{Et}{Hm^{2}}.\ln Qb = 0 \qquad (1)$$

The following approximation is applied (Dingemans, 1983) :

$$b = (8Et/Hm^{2})^{1/2} \qquad (2)$$

$$Q0 = (2b - 1)^{2} \qquad 0.5 < b < 1 \qquad (3)$$

$$= 0 \qquad b < 0.5$$

$$Qb = 0 \qquad b < 0.3 \qquad (4)$$

$$= Q0 - b^{2} . \frac{Q0 - \exp((Q0-1)/b^{2})}{b^{2} - \exp((Q0-1)/b^{2})} \qquad 0.3 < b < 0.9$$

$$= Q0 \qquad 0.9 < b < 1.0$$

The parameters Et and Hm (in eq. 1) are determined in subroutine DISPA.

The call of this subroutine is :

```
CALL FRABRE(IY)
```

Parameter :

```
IY (I)  y-coordinate of point in which the fraction
        of breaking waves Qb is computed
```

3.12. Subroutine TERMD

Function :
In this subroutine the terms of the two balans equations are evaluated in the point IX+1/2,IY,IT. The source terms S0 and dW0/dT are split up in components of wind generation, bottom friction, surf breaking and dissipation in adverse currents.

Structure :

```
   *TERMD*
   :-------------------------------------------------:
   | CALL TRSY compute transportation terms in       |
   |                                     y-direction |
   |-------------------------------------------------|
   | CALL TRST compute transportation terms in       |
   |                                     θ-direction |
   |-------------------------------------------------|
   | if diffraction is on then                       |
   |    :--------------------------------------------|
   |    | CALL DIFT compute diffraction terms         |
   |    :--------------------------------------------|
   | else                                            |
   |    :--------------------------------------------|
   |    | give diffraction terms the value 0.         |
   |-------------------------------------------------|
   | if wind generation is on then                   |
   |    :--------------------------------------------|
   |    | CALL SWIND compute wind generation terms    |
   |    :--------------------------------------------|
   | else                                            |
   |    :--------------------------------------------|
   |    | give wind generation terms the value 0.     |
   |-------------------------------------------------|
   | if bottom dissipation is on then                |
   |    :--------------------------------------------|
   |    | CALL SBOT compute bottom dissipation terms  |
   |    :--------------------------------------------|
   | else                                            |
   |    :--------------------------------------------|
   |    | give bottom dissipation terms the value 0.  |
   |-------------------------------------------------|
   | if surf breaking is on then                     |
   |    :--------------------------------------------|
   |    | CALL SSURF compute surf breaking terms      |
   |    :--------------------------------------------|
   | else                                            |
   |    :--------------------------------------------|
   |    | give surf breaking terms the value 0.       |
   |-------------------------------------------------|
   | if deep water breaking is on then               |
   |    :--------------------------------------------|
   |    | CALL SDBR compute current dissipation terms |
   |    :--------------------------------------------|
   | else                                            |
   |    :--------------------------------------------|
   |    | give current dissipation terms the value 0. |
   :-------------------------------------------------:
```

The call of this subroutine is :

```
     CALL TERMD(IY,IT)
```

Parameters :

```
     IY (I)
            ) coordinates of point in computational grid
     IT (I)
```

3.13. Subroutine TRSY

Function :
The transportation terms of the two balance equations in y-direction :

$$\frac{@}{@Y}\,(CY0.A0) \text{ and } \frac{@}{@Y}\,(CY0.W0.A0)$$

are determined in this subroutine.

Method :
A conservative central difference scheme is applied :

$$\frac{@f}{@Y} = \frac{f^{IX+1/2,IY+1,IT} - f^{IX+1/2,IY-1,IT}}{2.dY} \qquad (1)$$

Energy entering the computational region through the boundaries Y=0 and Y=LY is not taken into account. At these boundaries somewhat different schemes are used.

Structure :

```
*TRSY*
:-------------------------------------------------:
| if point is located on boundary and wave energy |
|       is entering the computational region then |
|   :---------------------------------------------|
|   | give flux of wave energy at boundary the    |
|   |                                    value 0. |
|-------------------------------------------------|
| compute transportation terms in y-direction     |
:-------------------------------------------------:
```

The call of this subroutine is :
    CALL TRSY(IY,IT)

Parameters :

```
    IY (I)
           ) coordinates of point in computational grid
    IT (I)
```

3.14. Subroutine TRST

Function :
The transportation terms of the two balance equations in θ-direction :

$$\frac{@}{@\theta}\,(C\theta 0.A0) \text{ and } \frac{@}{@\theta}\,(C\theta 0.W0.A0)$$

are determined in this subroutine.

Method :
A conservative central difference scheme is applied :

$$\frac{@f}{@\theta} = \frac{f^{IX+1/2,IY,IT+1} - f^{IX+1/2,IY,IT-1}}{2.d\theta} \qquad (1)$$

Energy entering the computational region through the boundaries θ=θa and θ=θb is not taken into account. At these boundaries somewhat different schemes are used. The leakage through these boundaries |Cθ0|.A0.Wr is kept.

Structure :

```
*TRST*
:--------------------------------------------------:
| if point is located on boundary and wave energy  |
|       is entering the computational region then  |
|    :---------------------------------------------|
|    | give flux of wave energy at boundary        |
|    |                                    value 0. |
|--------------------------------------------------|
| compute transportation terms in θ-direction      |
|--------------------------------------------------|
| if point is located on boundary and wave energy  |
|         is leaving the computational region then |
|    :---------------------------------------------|
|    | compute leakage of wave energy              |
:--------------------------------------------------:
```

The call of this subroutine is :
CALL TRST(IY,IT)

Parameters :
IY (I)
) coordinates of point in computational grid
IT (I)

3.15. Subroutine DIFT

Function :
The diffraction terms in the two balance equations :

$$\frac{\partial}{\partial\theta}\left(Cdif\left(CY0.\frac{\partial}{\partial X}A0-CX0.\frac{\partial}{\partial Y}A0\right)\right) \quad \text{and}$$

$$\frac{\partial}{\partial\theta}\left(Cdif\left(CY0.\frac{\partial}{\partial X}(W0.A0)-CX0.\frac{\partial}{\partial Y}(W0.A0)\right)\right)$$

are determined in subroutine DIFT.

Method :
Derivatives in x,y and θ direction are approximated by central difference schemes :

$$\frac{\partial f}{\partial X} = \frac{f^{IX+1,IY,IT} - f^{IX,IY,IT}}{dX} \tag{1}$$

$$\frac{\partial f}{\partial Y} = \frac{f^{IX+1/2,IY+1,IT} - f^{IX+1/2,IY-1,IT}}{2.dY} \tag{2}$$

$$\frac{\partial f}{\partial \theta} = \frac{f^{IX+1/2,IY,IT+1} - f^{IX+1/2,IY,IT-1}}{2.d\theta} \tag{3}$$

At the boundaries in the y-θ-plane somewhat different schemes are applied.

The call of this subroutine is :

```
     CALL DIFT(IY,IT)
```

Parameters :

```
     IY (I)
            ) coordinates of point in computational grid
     IT (I)
```

3.16. Subroutine SWIND

Function :
In this subroutine the wind generation components

$$S0_{wind} \quad \text{and} \quad \frac{d}{dt}W0_{wind}$$

of the source terms in the two balance equations are determined.

Method :
The following relations are used for the terms mentioned above :

$$S0_{wind} = \frac{U10^{3}}{g} Babcd \left| \frac{E/B}{a} \right|^{\frac{d-1}{d}} \left\langle 1- \left(\frac{E/B}{a}\right)^{\frac{2}{d}} \right\rangle \left\langle \frac{1}{b} \operatorname{atanh}\left(\left(\frac{E/B}{a}\right)^{\frac{1}{d}}\right) \right\rangle^{\frac{c-1}{c}} \qquad (1)$$

$$\frac{d\,W0}{dt}_{wind} = \frac{g^{3}}{U10^{5}} 2PI.a2^{1/b2} \; b2\left(\frac{W}{2PI}\right)^{(b2-1)/b2} \frac{1}{B} S0_{wind} \left\langle \frac{W}{2PI.a2(E/B)^{b2}} \right\rangle^{5} \qquad (2)$$

with

E is the dimensionless wave energy density $E0.\frac{g^{2}}{U10^{4}}$,E0=Wr.A0

W is the dimensionless mean action frequency $W0.\frac{U10}{g}$

B is the directional distribution of waves,E0(Y,0)=B(θ).Et(Y)

U10 is the wind velocity at 10m elevation relative to the current velocity

a,b,c and d are coefficients derived from literature

Eq. 1 and 2 hold only for growing waves (E<aB).In the case E>aB then the wind generation terms are assumed to be 0.

Structure :

```
   *SWIND*
   :-----------------------------------------------------:
   | if current is on then                               |
   |   :-------------------------------------------------|
   |   | determine wind speed vector relative to         |
   |   |                                  current vector |
   |   |-------------------------------------------------|
   |   | CALL DISTR determine directional distribution   |
   |   |                              around wind vector |
   |-----------------------------------------------------|
   | compute dimensionless wave energy density E         |
   |                                 and frequency W     |
   |-----------------------------------------------------|
   | if E < aB then                                      |
   |   :-------------------------------------------------|
   |   | compute wind generation terms                   |
   |   :-------------------------------------------------|
   | else                                                |
   |   :-------------------------------------------------|
   |   | give wind generation terms the value  0.        |
   :-----------------------------------------------------:
```

The call of this subroutine is :
CALL SWIND(IY,IT)

Parameters :

```
     IY (I)
           ) coordinates of point in computational grid
     IT (I)
```

3.17. Subroutine SBOT

Function :
The bottom dissipation terms

$S0_{bot}$ and $\frac{d}{dt} W0_{bot}$

are computed in this subroutine.

Method :
The following relations are applied :

$$S0_{bot} = -W_{bot} .Wr.A0 \qquad (1)$$

It is assumed that the bottom friction primarily affects the wave energy in the lower frequency range. According to part I of this report, it follows that:

$$\frac{dW0}{dt}_{bot} = W0 . (Cg0/C) . (1/(1-B)) . S0_{bot} / E0 \qquad (2)$$

with

$$W_{bot} = (8/PI)^{1/2} \frac{Wr^2}{g.\sinh^2 (K0.D)} .(Cfw.Ubot+Cfc.Ucur) \qquad (3)$$

The second term in eq. 3 is omitted if no current is present. The current velocity Ucur is given by :

$$Ucur = |Ux.\cos\theta + Uy.\sin\theta| \qquad (4)$$

In this formulation the effects of currents on bottom dissipation are included in the same way dissipation due to wave orbital velocities is determined. The same procedure with somewhat different relations is applied in the model CREDIZ (CREDIZ01 system documentation, 1984).
The terms

$$W_{bot} \quad \text{and} \quad \frac{d}{dt}W0_{bot}$$

are determined in point IX+1/2,IY,IT in the computational grid. The wave action A0 in the linear term S0bot is included implicitly in the two balance equations (section 3.20).
The coefficients Cfw,Cfc,a3 and b3 have to be determined, by the user, empirically.

The call of this subroutine is :
CALL SBOT(IY,IT)

Parameters :

IY (I)
) coordinates of point in computational grid
IT (I)

3.18. Subroutine SSURF

Function :
The terms representing dissipation of wave energy due to surf breaking

$$S0_{surf} \quad \text{and} \quad \frac{d}{dt}W0_{surf}$$

are determined in this subroutine.

Method :
Relations for these terms applied in this model are :

$$S0_{surf} = -W_{surf}.Wr.A0 \qquad (1)$$

A similar formula for frequency change is used as equation (2) in section 3.17:

$$\frac{dW0}{dt}_{surf} = W0 . (Cg0/C) . (1/(1-B)) . S0_{surf} / E0 \qquad (2)$$

with

$$W_{surf} = alfa1.\frac{1}{8PI}.Qb.W0.\frac{Hm^2}{Et} \qquad (3)$$

The terms

$$W_{surf} \quad \text{and} \quad \frac{d}{dt}W0_{surf}$$

are determined in point IX+1/2,IY,IT in the computational grid. The wave action A0 in the linear term S0surf is included implicitly in the two balance equations (section 3.20).
The coefficient alfal is of order 1 while the coefficients a4 and b4 should be determined empirically.

The call of this subroutine is :

```
     CALL SSURF(IY,IT)
```

Parameters :

IY (I)
) coordinates of point in computational grid
IT (I)

3.19. Subroutine SDBR

Function :
The terms representing dissipation of wave energy in currents with direction opposite to the direction of wave propagation (so-called blocking of waves)

$$S0_{dbr} \quad \text{and} \quad \frac{d}{dt}W0_{dbr}$$

are determined in this subroutine.

Method :
Relations for these terms applied in this model are :

$$S0_{dbr} = -W_{dbr} .Wr.A0, \quad W_{dbr} = \frac{f}{taul} \qquad (1)$$

$$\frac{dW0}{dt}_{dbr} = - \frac{g}{tau2}.W0 \qquad (2)$$

with

taul, tau2 are time constants to be determined empirically
(initially values $W0^{-1}$ are assumed)

$$f = (5.Wc/4.W0)^{-4} \qquad \text{for } Wc > 5/4 \,.\, W0$$

$$= 1. \qquad \text{for } Wc < 5/4 \,.\, W0$$

$$g = 1 - \frac{1 - (5.Wc/4.W0)^{-3}}{1 - (5.Wc/4.W0)^{-4}} \qquad \text{for } Wc > 5/4 \,.\, W0$$

$$= 1 - Wc/W0 \qquad \text{for } Wc < 5/4 \,.\, W0$$

If E0 = 0. then g is given the value 0..

The value of Wc is determined by the subroutine ITWN.

The call of this subroutine is :

```
     CALL SDBR(IY,IT)
```

Parameters :

```
     IY (I)
            ) coordinates of point in computational grid
     IT (I)
```

3.20. Subroutine SUMDE

Function :
In this subroutine the wave action A0 and -frequency W0 in the grid point IX+1,IY,IT are determined.

Method :
The two balance equations 1 and 2 are solved.

$$\frac{\partial}{\partial X}(CX0.A0) = -(\text{transportation+diffraction terms}) + \frac{S0_{wind}}{Wr} - \left(W_{bot} + W_{surf} + W_{dbr} - \frac{1}{W0}.\frac{dW0}{dt}\right)A0 \qquad (1)$$

$$\frac{\partial}{\partial X}(CX0.W0.A0) = -(\text{transportation+diffraction terms}) + \frac{W0.\ wind}{Wr} - \left(W_{bot} + W_{surf} + W_{dbr}\right)W0.A0 \qquad (2)$$

For brevity the transportation- and diffraction terms as well as the source term dW0/dt have not been written in full in eq. 1 and 2.
The numerical scheme applied to these equations is:

$$\frac{CX0^{IX+1,IY,IT}\,f^{IX+1,IY,IT} - CX0^{IX,IY,IT}\,f^{IX,IY,IT}}{dX} = G - \frac{H}{2}\left(f^{IX+1,IY,IT} + f^{IX,IY,IT}\right) \qquad (3)$$

with :

f represents the terms A0 (eq.1) or W0.A0 (eq.2)
G contains the non-linear terms of eq. 1 and 2
H contains the linear terms of eq. 1 and 2

Further the dissipation of wave energy (Wbot+Wsurf+Wdbr)A0.Wr.dθ is determined in point IX+1/2,IY,IT in the computational grid.

Structure :

```
   *SUMDE*
   :--------------------------------------------------:
   | determine wave action and -frequency             |
   |--------------------------------------------------|
   | if last corrector step is in progress then       |
   |   :----------------------------------------------|
   |   | compute dissipation of wave energy           |
   :--------------------------------------------------:
```

The call of this subroutine is :

```
     CALL SUMDE(IY,IT)
```

Parameters :

```
     IY (I)
             ) coordinates of point in computational grid
     IT (I)
```

3.21. Subroutine WRIRE

Function :
In the subroutine WRIRE arrays containing wave parameters at a line in the computational grid are written to the file REKRES.

Structure :

```
   *WRIRE*
   :----------------------------------------------------:
   | if line ≠ boundary x=0 then                        |
   |   :------------------------------------------------|
   |   | write leakage of energy FL,dissipation of      |
   |   | energy FD,fraction of breaking waves QB and the|
   |   | wave action transport velocity component Cθ0   |
   |   | to the file REKRES                             |
   |----------------------------------------------------|
   | write depth D,current velocity components UX,UY (if|
   | current is off then fill arrays with 0),wave action|
   | A0,-frequency W0,relative frequency Wr,wave number |
   | K0,group velocity CG0 and wave action transport    |
   | velocity components CX0,CY0 to the file REKRES     |
   :----------------------------------------------------:
```

The call of this subroutine is :

```
     CALL WRIRE(IX)
```

Parameter :

```
     IX (I)  results of line IX in the computational grid
             are written to the file REKRES
```

4. OUTPUT PROGRAM HISWA/OUTP

4.1. Structure of the program

The program HISWA/OUTP processes the results computed by the program HISWA/COMPU. The results are stored in the file 'REKRES'. The processing is controlled by the output requests encoded in the pool array OUTREQ or POOL(IOUTR+..), and using data in the array OUTDA or POOL(IOUTD+..).

HISWA/OUTP starts reading the output requests and output data. For each output request the file REKRES is read, for the output points values are interpolated from the data in that file.

linkage diagram of Hiswa/Outp

```
HISWA/OUTP
     |
     |______WRCOMX etc.    read output instructions
     |
     |______BLOKX          prints a block table
     |         |
     |         |______PSET      determines output point set
     |         |
     |         |______UVIPOL    reads results and interpolates
     |         |         |
     |         |         |______IXPOL
     |         |                   |
     |         |                   |______G1      interpolation
     |         |                   |
     |         |                   |______G2          functions
     |         |
     |         |______BLKHFD    block heading
     |         |
     |         |______BLKPT     prints numbers
     |                   |
     |                   |______UVHSIG etc.
     |
     |______TABPRT         prints a table
     |         |
     |         |______PSET      determines output point set
     |         |
     |         |______UVIPOL    reads results and interpolates
     |         |         |
     |         |         |______IXPOL
     |         |                   |
     |         |                   |...        see above
     |         |
     |         |______UVHSIG etc.   determines values
     |
     |______ISOVEC       produces a plot
     |         |
     |         |...       see next page
     |
     |______WSPEC        spectral output to disk
```

linkage scheme of plot routines

```
ISOVEC
    |
    |______PSET
    |
    |______UVIPOL
    |         |
    |         |______ ...
    |
    |______TOGPCP       writes certain numbers to file
    |
    |______PENTO        pen moves to certain position
    |
    |______PLYN         a line is plotted
    |         |
    |         |______SNYPT1     determines crossings
    |         |
    |         |______SNYPT2          with frame edge
    |
    |______PNAMES       plots names in the figure
    |         |
    |         |______PLNAME   plots one name
    |
    |______PCOAST       plots lines in the figure
    |
    |______PENTO        pen moves to certain position
              |
              |______PLYN
                        |
                        |______SNYPT1
                        |
                        |______SNYPT2
```

4.2. Main program

The main program reads the output instructions that have been composed by the Reading program of HISWA. Furthermore it reads some general data concerning the computation from the file containing the computational results.
Depending on the type of output request MAIN branches to the subroutine that produces that particular output. After this subroutine has finished its task, MAIN will process the next output instruction.

```
               * HISWA/OUTP  Main program  *
--------------------------------------------------------------
Read from file INSTU: some general output parameters and
pool arrays IOUTR (output requests) and IOUTD (output data).

Call WRCOMX    $ reads data concerning the computation from the
               $ file REKRES

For each output request do
    Obtain from array OUTREQ:
    - pointer to next request
    - type of output request RTYPE
    ----------------------------------------------------------
    Case RTYPE =
    = 'BLK' : Call BLOKX      $ block output
    = 'TAB' : Call TABPRT     $ tabular output
    = 'PLOT' : Call ISOVEC    $ plot
    = 'SPEC' : Call WSPEC     $ spectral output
----------------------------------------------------------------
```

4.3. Subroutines BLOKX, TABPRT, ISOVEC and WSPEC

Function:
Each of these routines produces certain type of output.

Method:
The output routines BLOKX, TABPRT, ISOVEC and WSPEC have to a large extent the same structure. Each of these works in the following way: It obtains from array IOUTR the necessary parameters for the output at hand. It calls PSET, which determines the type of the output point set, and which stores the user coordinates of the output points in the arrays IXP and IYP. It calls UVIPOL, which reads the computational results and interpolates these to the output points. It calls UVHSIG or another routine which calculates the requested variable, which it then prints, writes to disk, or plots, as the case may be.

Parameters are:

RTYPE	type of output request
IUREC	place where to find the output instruction in array IOUTR

```
            *  BLOKX, TABPRT, ISOVEC and WSPEC  *
-------------------------------------------------------
Obtain from array IOUTR:
   KNAAM       $ name of output point set
Call PSET      $ determines KTYPE, type of the point set,
               $ MIP, number of points in the set.
(in the case of BLOKX and ISOVEC:)
If KTYPE is not 'KADR' (frame)
Then Error message
-------------------------------------------------------
Obtain NVAR, number of variables that must be output.
Determine IUR1, and IUR2, indicating where to find ISOORT,
which indicates the type of variable wanted for output.

Call UVIPOL  $ reads computational results and
             $ interpolates to output points

For JVAR from 1 to NVAR
    Determine ISOORT $ type of variable
    Depending on value of ISOORT,
    output UVHSIG, UVDIR, UVPER etc.
    $ see section on coding and processing of variables
-------------------------------------------------------
```

4.4. Subroutine UVIPOL

Function:
Reading the computational results from file REKRES, and interpolating these to output points.

```
                         *  UVIPOL  *
    ---------------------------------------------------
    Transform coordinates of output points to comput.
    grid; store these in pool arrays IXR and IYR

    For each type of variable in output
        Determine necessary space in auxiliary array
        IFOP
    ---------------------------------------------------
    Call ADPOOL  $ reservs space for array IFOP

    For each line in computational grid
        Read from file REKRES:
        depth, Ux, Uy, action density, frequency, etc.

        For each variable in output
            Carry out first step of processing
            $ see section 4.6
            Call IXPOL  $ carries out interpolation
            $ and stores result in array IFOP
    ---------------------------------------------------
```

4.5. Subroutine IXPOL

Function: see subr. UVIPOL

```
                    *  IXPOL  *
    ---------------------------------------------------
    For all output points do
        If point near prsent line of computational grid
        Then Determine points on grid line which
             contribute to this output point;
             Determine contributions from grid points;
             Multiply by coefficients for interpolation
             in x-direction;
             Add result to record in pool array IFOP,
             in which results for the output point are
             stored.
    ---------------------------------------------------
```

4.6. Coding of output requests

The output requests are coded in array OUTREQ (pool array with pointer IOUTR). The coding of the output requests as given by the user is carried out by the subroutine PROUT in the reading program. Each output request is represented by one record in the array.

The structure of the array OUTREQ is as follows:
- OUTREQ(1) number of places occupied in the array.
- OUTREQ(2) place in the array where the last record starts.

for each record the following data:
- NEXT pointer to next record in the array
- TYPE type of output request
- SNAME name of point set for which output is requested,
  occupying places in the array.
- other data specifying the output request (see below).

The following TYPEs of output request occur:
'BLKP' or 'BLKD' block type print or output to file
'PLOT' plot of iso-lines and/or vectors
'SPEC' output of spectra to file.

The data in the record depend on the type of the output request. For each type they are described in detail as follows:

Block output

- 'BLKP' or 'BLKD' type of output
- SNAME output point set
- NREF file reference number of destination
- NVAR number of variables which is to be written.

For each of the NVAR variables:
- ISOORT type of physical variable
- DFAC factor with which each value is multiplied.

Plot

- 'PLOT' type of output
- SNAME output point set
- PTI title of the plot
- ISOORT type of physical variable in iso-line plot

- FSTEP step of function
- FMIN minimum function value
- FMAX maximum function value.
- ISOORT type of physical variable in vector plot
- SCALE vector scale.
- IPLAC flag for plotting names of places
- ILINS flag for plotting lines

Spectral output

- 'SPEC'
- SNAME
- NREF destination of the output (unit reference number)
- 15 value of ISOORT for spectral wave action
- 16 value of ISOORT for spectral average frequency

4.7. Coding and processing of variable types

In the output requests the type of variable that is wanted on output is encoded by the number ISOORT. With each type of variable a different procedure is carried out by the subroutine UVIPOL.

In the following table one finds for each value of ISOORT: code of the type in the user's output request, description of the variable type, procedure carried out by the subroutine UVIPOL, and the procedure carried out by one of the routines UVHSIG, UVDIR etc.
UVIPOL reads values of computed wave parameters from the file REKRES, and usually it will process these values and put the result into the auxiliary arrays IA1, IA2 and IA3. In the procedure description Sum(..) means the sum over the directions. Then the subroutine IXPOL is called by UVIPOL in order to put interpolated values into the proper places of the array IFOP. With this the task of UVIPOL is completed.
In the second stage one of the routines UVHSIG, UVDEP, UVPER etc. will calculate from the intermediate results written in array IFOP, the value of the significant wave height, the depth, the period etc. F1 in this procedure description denotes the first value written in array IFOP for a certain output point, F2 the second value etc.

```
1.  HSIG  significant wave height
    Proc. UVIPOL: Sum(RFN*WAN) into aux. array A1
                  Into array IFOP: interpolated value of A1
    Proc. UVHSIG: 4*Sqrt(F1)
2.  DIR   dominant wave direction
    Proc. UVIPOL: Sum(Theta*WAN) into A1, Sum(WAN) into A2
                  Into IFOP: interp. A1, interp. A2
    Proc. UVDIR: F1/F2 * 180/PI
3.  PER   mean wave period
    Proc. UVIPOL: Sum(WFN*WAN) into A1, Sum(WAN) into A2
                  Into IFOP: interp. A1, interp. A2
    Proc. UVPER: 2*PI*F2/F1
4.  DEPT  depth
    Proc. UVIPOL: into IFOP: interpolated value of DEN
    Proc. UVDEP: F1
5.  VEL   current velocity
    Proc. UVIPOL: into IFOP: interp. UX, interp. UY
    Proc. UVUX: F1
    Proc. UVUY: F2
6.  FORC   resulting force  exerted by  the waves  (formulae
    adopted from Battjes, 1974).
    Proc. UVIPOL: n=.., cost=cos(theta), sint=sin(theta)
          Sum ((n*cost**2 + (n-.5)) * WAN)  into A1
          Sum (n*cost*sint * WAN)  into A2
          Sum ((n*sint**2 + (n-.5)) * WAN)  into A3
          Into IFOP: x-derivative of A1, y-der. of A2,
                     x-der. of A2, y-der. of A3
    Proc. UVFRX: F1+F2
    Proc. UVFRY: F3+F4
7.  TRAN  energy transport
    Proc. UVIPOL: Sum(CXN*RFN*WAN)  into A1,
                  Sum(CYN*RFN*WAN)  into A2;
                  Into IFOP: interp. A1, interp. A2
    Proc. UVENX: F1
    Proc. UVENY: F2
8.  DSPR  directional spread
    Proc. UVIPOL: Sum(WAN) into A1, Sum(Theta*WAN) into A2,
                  Sum (Theta**2 * WAN)  into A3;
                  Into IFOP: interp. A1, interp. A2, interp.
    A3.
    Proc. UVDSPR: Sqrt(F3/F1-(F2/F1)**2) * 180/PI
9.  DISS  dissipated energy
    Proc. UVIPOL: Into IFOP: interp. FD
    Proc. UVDISS: F1
10. LEAK  energy leak over sector boundary
    Proc. UVIPOL: Into IFOP: interp. FL
    Proc. UVLEAK: F1
11. QB    fraction breaking waves
    Proc. UVIPOL: Into IFOP: interp. QB
    Proc. UVQB: F1
12. XP    x-coordinate of output point
    Proc. UVXP: Pool(IXP+IP)
13. YP    y-coordinate of output point
    Proc. UVYP: Pool(IYP+IP)
14. DIST  distance of output point along curve
    Proc. UVDIST: 0, if IP=1;
          Value in IP-1 + distance to point IP, otherwise
15. SPAC  spectral wave action
    Proc. UVIPOL:  put into IFOP  the interpolated  value of
    WAN (for each direction separately).
    Proc.  UVFF:  F1  is returned  (no  further  calculation
    needed).
16. SPFR  spectral average frequency
    Proc. UVIPOL:  put into IFOP  the interpolated  value of
    WFN (for each direction separately).
    Proc. UVFF: F1
```

5. SERVICE ROUTINES

5.1. Introduction

There are several categories of service routines, such as the pool handling routines INPOOL, ADPOOL etc., the command reading routines INKEY, INREAL, LEESEL etc., the error message routine MSGERR, the trace routine STRACE and the routines for communication between HISWA programs WRCOM and WRCOMX.

Most of the service routines are copied from CREDIZ with minor adjustments. In case the documentation is absent or insufficient, reference is made to the CREDIZ manual (CREDIZ01 system documentation, 1984).

5.2. Subroutine OPENF

In this subroutine all necessary files are opened in order to reserve input/output buffers. This action is taken in connection to repeated calls of the standard routine REQDA. This subroutine is copied from CREDIZ.

5.3 Subroutine INPOOL

The dynamic data pool is initialized by this subroutine.The dimension of the pool is determined from the common variable NPOOL (NPOOL * 1024). A number of empty arrays is initiated (50 in he program COMPU).
INPOOL is copied from CREDIZ with minor adjustments.

5.4 Subroutine REQDA

The standard routine REQDA is used for expansion of the dynamic data pool.
REQDA is a standard RWS/DIV routine.

5.5 Subroutines WRCOM and WRCOMX

A major part of the common blocks is written to and read from a file by the subroutines WRCOM and WRCOMX. This is necessary for the communication between the programs PREP, COMPU and OUTP. The difference between WRCOM and WRCOMX is the fact that WRCOMX does not read or write the common block UITVDA, containing instructions and information for the program OUTP.
WRCOM and WRCOMX are copied from CREDIZ with minor adjustments.

5.6 Subroutine WRPOOL

The subroutine WRPOOL reads or writes a pool array (unformatted) from resp. to a file.
WRPOOL is copied from CREDIZ with minor adjustments.

5.7 Subroutine ADPOOL

This routine is called by WRCOM,WRCOMX and in the main program for shrinking or expansion of an array in the dynamic data pool.
Subroutine ADPOOL is copied from CREDIZ with a few adjustments.

5.8 Subroutine WRDUMP

The contents of an array is printed by the subroutine WRDUMP.
WRDUMP is copied from CREDIZ with minor adjustments.

5.9 Subroutine DIVKOP

Function:
A title is printed above the output of the program.
DIVKOP is a standard RWS/DIV routine.

5.10 Subroutine STRACE

This subroutine, called at the start of every subroutine and the main program, provides a message of the entry of this subroutine resp. program.
STRACE is copied from the model CREDIZ.

5.11 Subroutine MSGERR

The subroutine MSGERR, called when an error is encountered during the execution of the program COMPU, provides an error message.
MSGERR is copied from CREDIZ.

5.12 Subroutine COPYCH

In subroutine COPYCH character strings are copied to real variables and back. It is copied from CREDIZ.

6. STORAGE OF DATA

6.1 Dynamic data pool

As in CREDIZ a dynamic data pool is used to obtain an efficient and flexible storage of arrays. With the subroutine ADPOOL the dimension of an array can be extended or reduced. The structure of the pool is the same as in CREDIZ.
An element of an array A is found by :

```
     A(I)=POOL(IA+I), IA is the adres of array A
```

A two-dimensional array is stored row by row :

```
     eg. : array B(1:n,1:m)
           storage in pool :
     B(1,1),...,B(n,1),B(1,2),...,B(n,2),B(1,m),...,B(n,m)
     thus B(k,l)=POOL(IB+(l-1)n+k),
```

where IB is the pointer of array B. For more detailed information on the structure of the dynamic data pool, reference is made to CREDIZ01 system documentation (1984).

In the programs HISWA/PREP and HISWA/OUTP the sequence of pool arrays is the same. The following table gives all pool arrays; in the first column the sequence number of the array is found, in the second column the name of the pointer to the array, and the third column describes the content of the array.

1	IDEP	depths in the bottom grid
2	IVX	x-velocities in the bottom grid
3	IVY	y-velocities in the bottom grid
4	IOUTR	output requests
5	IOUTD	output data
6	IFF	unused
7	IWAN	wave action density
8	IWFN	mean frequency
9	IRFN	mean relative frequency
10	IWKN	wave number
11	ICGN	group velocity
12	ICXN	x-comp. of group velocity
13	ICYN	y-comp. of group velocity
14	IFL	energy leakage on a line halfway between two lines in comp. grid
15	IFD	dissipation
16	IQB	fraction of breaking waves
17	ICT	transfer velocity in Theta-dir.
18	IXP	the set of output points in problem coordinates
19	IYP	,,
20	IXC	the set of output points in computational grid coordinates
21	IYC	,,
22	IDEN	depths on a line of the computational grid
23	IUXN	x-velocities
24	IUYN	y-velocities
25	INF	indicates for each variable JVAR where data can be found in array IFOP
26	IFOP	function values in the output points
27	IA1	auxiliary array
28	IA2	,,
29	IA3	,,

In the program HISWA/COMPU the following arrays are included in the pool; now the second column gives the name of the array as used in this manual, the third the name of the pointer, and the fourth the contents.

no	name	adres	description
1	DEP	IDEP	depths
2	VX	IVX	x-component current velocity
3	VY	IVY	y-component current velocity
4	WAO	IWAO	wave action (old line)
5	WFO	IWFO	wave frequency (old line)
6	WKO	IWKO	wave number (old line)
7	RFO	IRFO	relative frequency (old line)
8	CGO	ICGO	group velocity (old line)
9	CXO	ICXO	x-component wave action transport velocity (old line)
10	CYO	ICYO	y-component wave action transport velocity (old line)
11	WA	IWA	wave action (between old and new line)
12	WF	IWF	wave frequency (between old and new line)
13	WK	IWK	wave number (between old and new line)
14	RF	IRF	relative frequency (between old and new line)
15	CG	ICG	group velocity (between old and new line)
16	CX	ICX	x-component wave action transport velocity (between old and new line)
17	CY	ICY	y-component wave action transport velocity (between old and new line)
18	CT	ICT	θ-component wave action transport velocity (between old and new line)
19	WAN	IWAN	wave action (new line)
20	WFN	IWFN	wave frequency (new line)
21	WKN	IWKN	wave number (new line)
22	RFN	IRFN	relative frequency (new line)
23	CGN	ICGN	group velocity (new line)
24	CXN	ICXN	x-component wave action transport velocity (new line)
25	CYN	ICYN	y-component wave action transport velocity (new line)
26	DD	IDD	derivatives in bottom geometry
27	DC	IDC	derivatives in current field
28	HM	IHM	local maximum wave height
29	QB	IQB	fraction of breaking waves
30	ET	IET	directionally integrated wave energy
31	FD	IFD	dissipation of wave energy
32	FL	IFL	leakage of wave energy
33	UBOT	IUBOT	orbital velocity near the bottom
34	CRN	ICRN	maximum frequency that is able to transport wave energy against current (new line)
35	DEO	IDEO	depths (old line)
36	UXO	IUXO	x-component current velocity (old line)
37	UYO	IUYO	y-component current velocity (old line)
38	DEM	IDEM	depths (between old and new line)
39	UXM	IUXM	x-component current velocity (between old and new line)
40	UYM	IUYM	y-component current velocity (between old and new line)
41	DEN	IDEN	depths (new line)
42	UXN	IUXN	x-component current velocity (new line)
43	UYN	IUYN	y-component current velocity

			(new line)
44	CRO	ICRO	maximum frequency that is able to transport wave energy against current (old line)
45	CR	ICR	maximum frequency that is able to transport wave energy against current (between old and new line)

6.2 Common blocks

A number of common blocks are defined in which principal data for the model HISWA is included.Each of these blocks contains a certain category of information :

name	description
TITEL	title
DEPROS	location and dimensions of bottom grid
REKROS	location and dimensions of computational grid
TRANSF	transformation coefficients between different grids
NUMS	information on the numerical scheme
TERMDE	terms of the balance equations
TESTDA	information for output control (especially the tracing of errors)
FYSPAR	physical parameters
UITVDA	information for the program OUTP (not used by the program COMPU)
REFNRS	data set reference numbers
LEESDA	information for reading data
POOL	references to arrays in the dynamic data pool

Most of the common blocks are copied from the model CREDIZ. In the following only the changes with regard to the common blocks in CREDIZ are discussed. For more information reference is made to the CREDIZ01 system documentation (1984).

TITEL as in CREDIZ

DEPROS the elements AKX,CCGX and WKCX are omitted

REKROS the following elements are added :
MTR number of grid points in θ-direction
TETAA
) boundaries of computational grid in θ-direction
TETAB
DT grid size in θ-direction

TRANSF as in CREDIZ

NUMS contains the following elements :
NCOR number of corrector steps
IPRE indicator of predictor step
ICOR indicator of corrector step
IOBW option for the representation of the boundary condition at x=0
ICUR switch for the introduction of current
IDIF switch for the introduction of diffraction
IWIND switch for the introduction of wind

IBOT switch for the introduction of bottom dissipation
ISURF switch for the introduction of surf breaking
IDBR switch for the introduction of deep water breaking
INCHS significant wave height of incoming wave field
INCPER average period of incoming waves
INCDIR average direction of incoming waves
INCAT wave action for incoming wave field
INCFR average frequency of incoming waves
NBOUR switch for right boundary condition (y=0)
=0: no waves from outside
=1: equal to incident wave field
=2: read from file NFBR
NBOUL idem, for left boundary
NEGMES counts number of messages of neg. action etc.
MAXMES maximum number of messages (see NEGMES)
WDIP wind direction relative to problem grid
WDIC wind direction relative to computational grid
U10 wind velocity at 10 m. elevation
U10C wind velocity at 10 m. elevation relative to current
ADIR coefficient for directional distribution of waves
AT(300) wave action at boundary x=0
WT(300) wave frequency at boundary x=0
BDIR(30) directional distribution of waves
CDIF diffraction coefficient
PWIND(10) parameters of wind generation (a,b,c,d,a2,b2)
PBOT(5) parameters of bottom dissipation (Cfw,Cfc,a3,b3)
PSURF(5) parameters of surf breaking (alfa1,gamma,a4,b4)
PDBR(5) parameters of deep water breaking terms (tau1,tau2)
CNORM normalization coefficient of directional distribution

TERMDE contains the following elements :
FYA
) transportation terms in y-direction
FYF
FTA
) transportation terms in θ-direction
FTF
DIFA
) diffraction terms
DIFF
WINDA
) wind generation terms
WINDF
WBOT
) bottom dissipation terms
BOTF
WSURF
) surf breaking terms
SURFF
WDBR
) deep water breaking terms
DBRF
TDIS dissipation of wave energy
TLEAK leakage of energy

TESTDA as in CREDIZ

FYSPAR the following elements are omitted :
IM,DEP1,DEP2,DEP3,UX1,UX2,UX3,UY1,UY2,UY3,
AK1,AK2,AK3,CCG1,CCG2,CCG3,WKC1,WKC2,WKC3,
ISTA1,ISTA2,ISTA3,SIGMA1,SIGMA2,SIGMA3,
SINH1,SINH2,SINH3,AMPL1,AMPL2,AMPL3
the following elements are added :
DEPTH depth in a point in the bottom grid

```
        UXC
            ) current velocity components relative to computational grid
        UYC
        WNU wave number

UITVDA  same as in CREDIZ, with the addition of:
        XPQ    coord. of frame base point in user coordinates
        YPQ                          ,,
        UCOS   coeff. to transform from computational to
        USIN          frame coordinates


REFNRS  the elements HULPF1 and HULPF2 are omitted;
        the following items are added:
        WERKF2  second workfile
        NFBR  ref. number for right boundary condition
        NFBL  ref. number for left boundary condition
        LFNEST  ref. number for incoming waves and for nesting

LEESDA  as in CREDIZ

POOL    this block will be adjusted to the new construction
        of the dynamic data pool (section 6.1)
```

6.3 Files

In the model HISWA a number of files are used that serve as communication tools between the programs PREP, COMPU and OUTP. Two of these files are used by the program COMPU :

a) INSTR

This file contains instructions formulated in PREP and data necessary for the computations carried out in the program COMPU.

```
contents :
     NPOOL dimension of pool
     ITEST test parameter
     common blocks :
            TITEL, DEPROS, REKROS, TRANSF,
            NUMS, FYSPAR, REFNRS, UITVDA, TESTDA
     arrays :
            DEP,VX,VY,OUTREQ,OUTDA
```

b) REKRES

In file REKRES the results of the computations carried out in the program COMPU are stored. This file will be read by the program OUTP.

```
contents :
     common blocks :
            TITEL, DEPROS, REKROS, TRANSF,
            NUMS, FYSPAR, REFNRS
     arrays :
            DEP, VX, VY

            DEN, UXN, UYN, WAN, WFN,
                                     ) line x=0
            RFN, WKN, CGN, CXN, CYN

            FL, FD, QB, CT           ) line x=(i-1/2).dx )

            DEN, UXN, UYN, WAN, WFN,                     )i=1,MXR-1
                                     ) line x=i.dx       )
            RFN, WKN, CGN, CXN, CYN                      )
```

Each array in the files described above is preceded by the dimension of the array. The same conventions as described under 6.1 are applied with regard to he storage of two-dimensional arrays. Files are read and written unformatted.

REFERENCES

Abramowitz,M. and Stegun,I.A.,1968,
Handbook of mathematical functions,Dover publications

Battjes, J.A., 1974, Computation of set-up, longshore currents, run-up and overtopping due to wind-generated waves.
Doctor's thesis, Delft.

CREDIZ01 system documentation,1984 (in Dutch)
Rijkswaterstaat, Dienst Informatieverwerking (Data Processing Division), Rijswijk Z.H., the Netherlands.

Holthuijsen,L.H. and Booij,N.,1983,Selection and formulation of a numerical shallow water wave hindcast model,Delft University of Technology,Report no 17-83