SWASH

USER MANUAL

SWASH version 11.01A

by : The SWASH team

mail address : Delft University of Technology
Faculty of Civil Engineering and Geosciences
Environmental Fluid Mechanics Section
P.O. Box 5048
2600 GA Delft
The Netherlands

website : *http://www.tudelft.nl/swash*

Contents

Chapter 1

About this manual

The information about the SWASH package is distributed over three different documents. This User Manual describes the specifications for the input of the SWASH model. The Implementation Manual explains the installation procedure and the usage of SWASH on a single- or multi-processor machine with distributed memory. The Scientific/Technical documentation discusses the physical and mathematical details and the discretizations that are used in the SWASH program. Apart from these documents, programmers who want to further develop SWASH can also consult the Programming rules as applied for SWAN; see *http://www.swan.tudelft.nl* for further details.

To make this manual more accessible we briefly describe the contents of each chapter. In Chapter 2 general description of the model and some instructions concerning the usage of SWASH, the treatment of grids, boundary conditions, etc. are given. It is advised to read this chapter before consulting the rest of the manual. Chapter 3 gives some remarks concerning the input and output files of SWASH. Chapter 4 describes the complete set of commands of the program SWASH. This chapter will be the most consulted part of the manual. In Chapter 5 some guidelines for setting up a command file is outlined.

This manual also contains some appendices. In Appendix A definitions of some parameters are given. Next, Appendix B outlines the syntax of the command file (or input file). Finally, a complete set of all the commands use in SWASH can be found in Appendix C.

If this is your first time to start working with SWASH we suggest you to first read Chapters 2 and 3. Next, Chapter 5 is recommended to practice the use of SWASH while guiding you through the main steps to set up a wave model.

Chapter 2

General description and instructions for use

2.1 Introduction

The purpose of this chapter is to provide the user with relevant background information on SWASH and to give some general advice in choosing the basic input for SWASH computations.

A general suggestion is: start simple. SWASH helps in this with default options. Moreover, it is a good idea to read Chapter 5 first when setting up a command file for the first time. Furthermore, suggestions are given that should help the user to choose among the many options conditions and in which mode to run SWASH (1D or flume-like, 2D or basin-like, depth-averaged or multi-layered, etc.). In addition, the way you need to specify the parameters and options resembles to that of SWAN. Hence, those users who are familiar with SWAN should be able to use SWASH without much effort. It is recommended to carry out some available test cases first to get acquaint with the program.

2.2 Background, features and applications

2.2.1 Objective and context

SWASH is a general-purpose numerical tool for simulating non-hydrostatic, free-surface, rotational flows and transport phenomena in one, two or three dimensions. The governing equations are the nonlinear shallow water equations including non-hydrostatic pressure and some transport equations, and provide a general basis for simulating

- wave transformation in both surf and swash zones due to nonlinear wave-wave interactions, interaction of waves with currents, interaction of waves with structures, wave damping due to vegetation, and wave breaking as well as runup at the shoreline,

- complex changes to rapidly varied flows typically found in coastal flooding resulting from e.g. dike breaks, tsunamis, and flood waves,
- density driven flows in coastal seas, estuaries, lakes, and rivers, and
- large-scale ocean circulation, tides and storm surges.

The model is referred to as a wave-flow model and is essentially applicable in the coastal regions up to the shore. This has prompted the acronym SWASH for the associated code, standing for Simulating WAves till SHore. The basic philosophy of the SWASH code is to provide an efficient and robust model that allows a wide range of time and space scales of surface waves and shallow water flows in complex environments to be applied. As a result, SWASH allows for the entire modelling process to be carried out in any area of interest. This includes small-scale coastal applications, like waves approaching a beach, wave penetration in a harbour, flood waves in a river, oscillatory flow through canopies, salt intrusion in an estuary, and large-scale ocean, shelf and coastal systems driven by Coriolis and meteorological forces to simulate tidal waves and storm surge floods.

SWASH is close in spirit to SWAN (Simulating WAves Nearshore) with respect to the pragmatism employed in the development of the code in the sense that comprises are sometimes necessary for reasons of efficiency and robustness. On the one hand, it provides numerical stability and robustness, and on the other hand it gives accurate results in a reasonable turn-around time.

2.2.2 A bird's-eye view of SWASH

The open source code of SWASH has been developed based on the work of Stelling and Zijlema [4], Stelling and Duinmeijer [2], Zijlema and Stelling [6, 7], Smit et al. [1], Zijlema et al. [8] and Zijlema [5]. The main elements of the SWASH code are:

- It is based on an explicit, second order finite difference method for staggered grids whereby mass and momentum are strictly conserved at discrete level. As a consequence, this simple and efficient scheme is able to track the actual location of incipient wave breaking. Also, momentum conservation enables the broken waves to propagate with a correct gradual change of form and to resemble steady bores in a final stage. Yet, this approach is appropriate for hydraulic jumps, dam-break problems and flooding situations as well.
- In the case of flow contractions, the horizontal advective terms in momentum equations are approximated such that constant energy head is preserved along a streamline.
- By considering the similarity between breaking waves and moving hydraulic jumps, energy dissipation due to wave breaking is inherently accounted for. In addition, nonlinear wave properties under breaking waves such as asymmetry and skewness are preserved.

- For accuracy reason, the pressure is split-up into hydrostatic and non-hydrostatic parts. Time stepping is done in combination with a projection method, where correction to the velocity field for the change in non-hydrostatic pressure is incorporated. Moreover, space discretization precedes introduction of pressure correction, so that no artificial pressure boundary conditions are required.

- Because of the abovementioned pressure splitting, hydrostatic flow computation can be done easily as well by simply switching off the non-hydrostatic pressure. As such, SWASH is appropriate for the simulation of large-scale tides and storm surges.

- With respect to time integration of the continuity and momentum equations, the second order leapfrog scheme is adopted, as it does not alter the wave amplitude while its numerical dispersion is favourable. This will prove beneficial to wave propagation.

- Alternatively, time discretization may take place by explicit time stepping for horizontal advective and viscosity terms and semi-implicit time stepping using the $\theta-$method for both surface level and pressure gradients as well as the free-surface condition. As a consequence, unconditional stability is achieved with respect to the celerity of gravity waves. The enhanced stability of this time stepping allows larger time steps, by a factor of five to ten, compared to the leapfrog scheme, and may thus be beneficial to large-scale applications such as tidal flow, and wind and density driven circulation.

- The physical domain can be discretized by subdivision of the continuum into cells of arbitrary shape and size. Furthermore, a distinction is made between the definition of the grid in the horizontal and vertical direction. In general, we consider two types of grids: structured and unstructured. A structured grid is employed where each interior cell is surrounded by the same number of cells. In unstructured grids, however, this number can be arbitrarily. For this reason, the level of flexibility with respect to the grid point distribution of unstructured grids is far more optimal compared to structured grids. In the horizontal planes, rectilinear, orthogonal curvilinear, boundary-fitted, (non)uniform grid or unstructured grid can be considered. In the current version of SWASH (since 7.01) only triangular meshes can be employed. Either Cartesian coordinates on a plane or spherical coordinates on the globe can be defined. In the vertical direction, the computational domain is divided into a fixed number of layers in a such a way that both the bottom topography and the free surface can be accurately represented. In this way, it permits more resolution near the free surface as well as near the bed.

- In order to resolve the frequency dispersion up to an acceptable level of accuracy, a compact difference scheme for the approximation of vertical gradient of the non-hydrostatic pressure is applied in conjunction with a vertical layer mesh employing equally distributed layers. This scheme receives good linear dispersion up to $kd \approx 8$ and $kd \approx 16$ with two and three equidistant layers, respectively, at 1% error in phase velocity of primary waves (k and d are the wave number and still water depth, respectively). The model improves its frequency dispersion by simply increasing the number of vertical layers.

- The three-dimensional simulation of flows exposing strong vertical variation amounts to the accurate computation of vertical turbulent mixing of momentum and some constituents, such as salt, heat and suspended sediment, combined with an appropriate vertical terrain-following grid employing sufficient number of non-equidistant layers. The vertical variation may be generated by wind forcing, bed stress, Coriolis force or density stratification.

- The combined effects of wave-wave and wave-current interaction in shallow water are automatically included and do not need any additional modelling, such as calculating the radiation stresses explicitly and subsequently solving a wave-averaged hydrodynamic model separately.

- For a proper representation of the interface of water and land, a simple approach is adopted that tracks the moving shoreline by ensuring non-negative water depths and using the upwind water depths in the momentum flux approximations.

- Since version 7.01, the wave-flow model SWASH has been extended to unstructured triangular meshes. The main motivation for the application of such meshes is the ease of local grid refinements for the simulation of wave dynamics. The covolume method has been adopted for the spatial discretization of the shallow water equations with the primitive variables. The velocity components normal to the cell faces are employed as the primary unknowns in the discretization, whereas the water level and non-hydrostatic pressure are resided at cell centers. Although the covolume method is free of spurious pressure modes, it is practically limited to Delaunay-Voronoi meshes. This may impede the user flexibility to generate adequate grids comprising the necessary refinements. Nevertheless, the associated orthogonality requirement is found not to be a limiting factor in wave-related applications. This is explained by the fact that the required mesh resolution is reasonably non-uniform for the scale of wave dynamics across the entire domain. The unstructured version of SWASH is applicable to a wide range of wave-flow problems to investigate the nonlinear dynamics of free surface waves over varying bathymetries.

Like SWAN, the software package of SWASH includes user-friendly pre- and post-processing and does not need any special libraries (e.g. PETSc, HYPRE). In addition, SWASH is highly flexible, accessible and easily extendible concerning several functionalities of the model. As such, SWASH can be used operationally and the software can be used freely under the GNU GPL license (https://swash.sourceforge.io).

2.2.3 Model features and validity of SWASH

SWASH is a free-surface, terrain-following, multi-dimensional hydrodynamic simulation model to describe unsteady, rotational flow and transport phenomena in coastal waters as driven by e.g., waves, tides, buoyancy and wind forces. It solves the continuity and momentum equations, and optionally the equations for conservative transport of salinity, temperature and suspended load for both noncohesive sediment (e.g. sand) and cohesive sediment (mud, clay, etc.). In addition, the vertical turbulent dispersion of momentum and diffusion of salt, heat and suspended sediment are calculated by means of the standard $k - \varepsilon$ turbulence model. The transport equations are coupled with the momentum equations through the baroclinic forcing term, whereas the equation of state is employed that relates density to salinity, temperature and suspended sediment.

SWASH accounts for the following physical phenomena:

- wave propagation, frequency dispersion, shoaling, refraction and diffraction,
- depth-limited wave growth by wind,
- nonlinear wave-wave interactions (including surf beat and triads),
- wave breaking,
- wave runup and rundown,
- moving shoreline,
- bottom friction,
- partial reflection and transmission,
- wave interaction with structures,
- wave interaction with moving rigid bodies,
- wave-current interaction,
- wave-induced currents,
- vertical turbulent mixing,
- subgrid turbulence,
- turbulence anisotropy,
- wave damping induced by aquatic vegetation,
- rapidly varied flows,

- tidal waves,
- bores and flood waves,
- wind driven flows,
- space varying wind and atmospheric pressure,
- density driven flows,
- transport of suspended load for (non)cohesive sediment,
- turbidity flows, and
- transport of tracer.

The model has been validated with a series of analytical, laboratory and field test cases. Overall, the level of agreement between predictions and observations is quite favourable, particularly in view of the fact that a wide range of wave conditions and topographies were modelled.

SWASH is proved to reproduce the main features of surf zone dynamics, such as nonlinear shoaling, wave breaking, wave runup and wave-driven currents. For instance, considering a typical surf zone where the dominant processes of triad interaction and depth-induced breaking can be isolated, it was found that the model yields a realistic representation of the observed frequency spectra, including the overall spectral shape at frequencies above the spectral peak, and the inclusion of subharmonics. This is followed by a transformation toward a broadband spectral shape as the waves approach the shoreline.

Such phenomena appear to be rooted in the ability of the staggered (covolume) momentum-conservative scheme to mimic the dynamics within travelling bores associated with wave breaking across the surf zone. In addition, because of the use of discrete analogs of the physical properties of the governing PDEs (e.g. topology, flux conservation) the effect of discretization error originated from the discretization approximations is thus limited, in that the numerical solution is merely influenced by the mesh resolution and mesh quality. Also, the employed SWASH scheme rules out non-physical artefacts that can occur when using a traditional discretization approach (e.g. finite volume methods).

2.2.4 Relation to Boussinesq-type wave models

SWASH is not a Boussinesq-type wave model. In fact, SWASH may either be run in depth-averaged mode or multi-layered mode in which the three-dimensional computational domain is divided into a fixed number of vertical terrain-following layers. SWASH improves its frequency dispersion by increasing this number of layers rather than increasing the order of derivatives of the dependent variables like Boussinesq-type wave models. Yet, it contains at most second order spatial derivatives, whereas the applied finite difference approximations are at most second order accurate in both time and space.

In addition, SWASH does not have any numerical filter nor dedicated dissipation mechanism to eliminate short wave instabilities. Neither does SWASH include other ad-hoc measures like the surface roller model for wave breaking, the slot technique for moving shoreline, and the alteration of the governing equations for modelling wave-current interaction. As such, SWASH is very likely to be competitive with the extended Boussinesq-type wave models in terms of robustness and the computational resource required to provide reliable model outcomes in challenging wave and flow conditions. Therefore, it can be seen as an attractive alternative to the Boussinesq-type wave models.

2.2.5 Relation to circulation and coastal flow models

SWASH is a non-hydrostatic wave-flow model and is originally designed for wave transformation in coastal waters. However, with the extension of meteorological and baroclinic forcing and solute transport (since version 2.00), this model is capable of using for large-scale flow and transport phenomena driven by tidal, wind and buoyancy forces. In this respect, SWASH is very similar to other traditional hydrodynamic models, such as WAQUA, Delft3D-FLOW, ADCIRC, ROMS, FVCOM, UNTRIM, SLIM and SUNTANS. They mainly differ in numerics and geometric flexibility, though SWASH supports unstructured triangular grids (since version 7.01).

The need to accurately predict small-scale coastal flows and transport of contaminants encountered in environmental issues is becoming more and more recognized. The aforementioned models, however, are orginally designed to simulate large-scale circulation. The development of these models is often dictated by model limitations, numerical techniques and computer capabilities. For instance, the hydrostatic pressure assumption prohibits the models to appropriately simulate surface waves, internal waves, and small-scale flows around hydraulic structures.

In principle, SWASH has no limitations and can capture flow phenomena with spatial scales from centimeters to kilometers and temporal scales from seconds to hours. Yet, this model can be employed to resolve the dynamics of wave transformation, buoyancy flow and turbulent exchange of momentum, salinity, heat and suspended sediment in shallow seas, coastal waters, surf zone, estuaries, reefs, rivers and lakes.

2.3 Internal scenarios, shortcomings and coding bugs

Sometimes the user input to SWASH is such that the model produces unreliable or unstable results. This may be the case, for instance, if the bathymetry is not well resolved or the boundary conditions are wrong (not well-posed). In addition, SWASH may invoke some

internal scenarios instead of terminating the computations. The reasons for this model policy is that

- SWASH needs to be robust, and
- the problem may be only very local, or
- the problem needs to be fully computed before it can be diagnosed.

Examples are:

- The minimum depth for checking drying and flooding may be adapted as soon as the water level is below the bottom.
- A dynamically adjusted time step controlled by the Courant number in a user prescribed range is implemented. This time step control for a particular SWASH run is provided in the `PRINT` file.
- Based on a stability criterion due to the explicit treatment of the horizontal eddy viscosity term in the momentum equations, a maximum of the eddy viscosity is determined at each time step. The user will be informed about this measure by means of a warning in the `PRINT` file when the number of instable points is more than 1% of the total number of grid points.

Some other problems which the SWASH user may encounter are due to more **fundamental shortcomings**, e.g., turbulence modelling, and unintentional **coding bugs**.

Because of the issues described above, the results may look realistic, but they may (locally) not be accurate. Any change in these scenarios or shortcomings, in particular newly discovered coding bugs and their fixes, are published on the SWASH website and implemented in new releases of SWASH.

2.4 Units and coordinate systems

SWASH expects all quantities that are given by the user to be expressed in S.I. units: m, kg, s and composites of these with accepted compounds, such as Newton (N) and Pascal (Pa). Consequently, the water level and water depth are in m, flow velocity in m/s, etc. For wind, (incident) wave and flow direction both the Cartesian and *a* nautical convention can be used (see below). Directions and spherical coordinates are in degrees (^{0}) and not in radians.

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

In the input and output of SWASH the direction of wind, (incident) wave and current are defined according to either

- the Cartesian convention, i.e. the direction to where the vector points, measured counterclockwise from the positive $x-$axis of this system (in degrees) or
- a nautical convention (there are more such conventions), i.e. the direction where the wind or the waves come from, measured clockwise from geographic North.

All other directions, such as orientation of grids, are according to the Cartesian convention!

For regular grids, i.e. uniform and rectangular, Figure 4.1 (in Section 4.5) shows how the locations of the various grids are determined with respect to the problem coordinates. All grid points of curvilinear and unstructured grids are relative to the problem coordinate system.

2.5 Choice of grids and time windows

2.5.1 Introduction

Both spatial grids and time windows need to be defined.

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

- a computational spatial grid on which SWASH performs the computations,
- one (or more) spatial input grid(s) for the bottom, (initial) current field, (initial) water level, bottom friction, wind, atmospheric pressure, porosity regions, grain sizes of armour rocks, heights of breakwaters and quays, vegetation density and drafts/labels of floating objects (each input grid may differ from the others),
- one (or more) spatial input grid(s) for transport of constituents to define initial and stationary boundary conditions for constituents, and
- one (or more) spatial output grid(s) on which the user requires output of SWASH.

Wind, bottom friction, grain sizes, heights of porous structures and vegetation density do not require a grid if they are uniform over the area of interest.

For one-dimensional situations, i.e. $\partial/\partial y \equiv 0$, SWASH can be run in 1D mode.

If a uniform, rectangular computational grid is chosen in SWASH, then all input and output grids must be uniform and rectangular too, but they may all be different from each other.

If an orthogonal curvilinear computational grid is chosen in SWASH, then each input grid should be either uniform, rectangular or identical to the used curvilinear computational grid.

If an unstructured computational spatial grid is chosen in SWASH, then each input grid should be either uniform, rectangular or identical to the used unstructured grid.

Also, SWASH may operate with different time windows with different time steps (each may have a different start and end time and time step):

- one computational time window in which SWASH performs the computations,
- one (or more) input time window(s) in which wind and pressure field (if present) are given by the user (each input window may differ form the others) and
- one (or more) output time window(s) in which the user requires output of SWASH.

During the computation SWASH obtains bottom, current, water level, wind, pressure, bottom friction, porosity, grain size, structure height and vegetation density information by tri-linear **interpolation** from the given input grid(s) and time window(s). The output is in turn obtained in SWASH by bi-linear interpolation in space from the computational grid; there is no interpolation in time, the output time is shifted to the nearest computational time level. Interpolation errors can be reduced by taking the grids and windows as much as equal to one another as possible (preferably identical). It is recommended to choose output times such that they coincide with computational time levels.

2.5.2 Computational grid and time window

The **computational grid** must be defined by the user. The orientation (direction) can be chosen arbitrarily.

If the computational grid extends outside the input grid, the reader is referred to Section 2.5.3 to find the assumptions of SWASH on depth, current, water level, wind, bottom friction, porosity, grain size, structure height and vegetation density in the non-overlapping area.

The spatial resolution of the computational grid should be sufficient to resolve relevant details of the wave field. Usually a good choice is to take the resolution of the computational grid approximately equal to that of the bottom or current grid. See Chapter 5 for further details.

Alternatively, the user may apply a curvilinear, boundary-fitted grid. This grid must be orthogonal, but may either be uniform or non-uniform. The domain boundaries may be curved. The exception is when a wave spectrum is imposed, in which case the wavemaker boundaries must be non-curved (see also Chapter 5 for details).

If necessary, an unstructured triangular grid may be used for a further extensive local mesh refinements (e.g. in the surf zone).

SWASH may not use the entire user-provided computational grid if the user defines exception values on the bottom grid (see command `INPGRID BOTTOM`) or on the curvilinear computational grid (see command `CGRID`). A computational grid point is either

- wet, i.e. the grid point is included in the computation since it represents water or
- dry, i.e. the grid point is excluded from the simulation since it represents land which may vary as moving shoreline or
- exceptional, i.e. the grid point is permanently excluded from the computations since it is so defined by the user. This provides a means to make a line of dams or screens through the computational domain, separating the flow on both sides. This line of thin dams may represent a small obstacle with subgrid dimensions that possibly influence the local flow (e.g. breakwater, jetty, or small harbour).

It must be noted that for parallel runs using MPI the user must indicate an exception value when reading the bottom levels (by means of command `INPGRID BOTTOM EXCEPTION`), if appropriate, in order to obtain good load balancing.

For further suggestions regarding choice of the resolution and orientation of the computational grid the user is kindly referred to Chapter 5.

The **computational time window** must be defined by the user. The computational window in time must start at a time that is early enough that the initial state of SWASH has propagated through the computational area before reliable output of SWASH is expected. Before this time the output may not be reliable since usually the initial state is not known.

The computational time step must be given by the user. Since, SWASH is based on explicit schemes, it is limited by a Courant stability criterion (which couples time and space steps). Moreover, the accuracy of the results of SWASH are obviously affected by the time step. Generally, the time step in SWASH should be small enough to resolve the time variations of computed wave field itself. Usually, it is enough to consider the time variations of the wave boundary conditions.

2.5.3 Input grid(s) and time window(s)

The bathymetry, current, water level, bottom friction (if spatially variable), wind (if spatially variable), atmospheric pressure, porosity regions, grain sizes, heights of porous structures (if spatially variable) vegetation density (if spatially variable) and drafts (and optionally labels) of floating objects (e.g. moored ships, WECs) need to be provided to SWASH on so-called input grids. It is best to make an input grid so large that it completely covers the computational grid.

When the atmospheric pressure is included, it must be combined with space varying wind. They may be read from a meteorological file. Space varying wind and pressure is of particular importance for the simulation of storm surges.

In the region outside the input grid SWASH assumes that the bottom level, the water level, bottom friction, atmospheric pressure, stone diameter and vegetation density are identical to those at the nearest boundary of the input grid (lateral shift of that boundary). In the regions not covered by this lateral shift (i.e. in the outside quadrants of the corners of the input grid), a constant field equal to the value at the nearest corner point of the input grid is taken. For the current and wind velocity, SWASH takes 0 m/s for points outside the input grid, while for porosity and structure height, SWASH takes 1 and 99999 (i.e. emerged), respectively, for points outside the input grid.

One should choose the spatial resolution for the input grids such that relevant spatial details in the bathymetry, current, bottom friction, wind and pressure and floating objects are properly resolved. Special care is required in cases with sharp and shallow ridges (sand bars, shoals, breakwaters) in the sea bottom and extremely steep bottom slopes. Very inaccurate bathymetry can result in very inaccurate wave transformation or flooding and drying. In such cases the ridges are vitally important to obtain good SWASH results. This requires not only that these ridges should be well represented on the input grid but also after interpolation on the computational grid. This can be achieved by choosing the grid lines of the input grid along the ridges (even if this may require some slight "shifting" of the ridges) and choosing the computational grid to be identical to the input grid (otherwise the ridge may be "lost" in the interpolation from the bottom input grid to the computational grid). An alternative is to smooth the bottom gradients. But this should be done in a way that the quality and feature of the bathymetric data is more or less the same.

In SWASH, wind, pressure and bottom friction may be time varying. In that case they need to be provided to SWASH in so-called input time windows (they need not be identical with the computational, output or other input windows). It is best to make an input window larger than the computational time window. SWASH assumes zero values at times before the earliest begin time of the input parameters (which may be the begin time of any input parameter such as wind). SWASH assumes constant values (the last values) at times after the end time of each input parameter. The input windows should start early enough so that the initial state of SWASH has propagated through the computational area before reliable output of SWASH is expected.

Finally, one should use a time step that is small enough that time variations in the wind, pressure and bottom friction are well resolved.

2.5.4 Input grid(s) for transport of constituents

With SWASH some transport phenomena of constituents can be simulated that result from tidal, wind and wave forcing in stratified flows. The considered constituents are salinity, temperature and suspended sediment load. The presence of these constituents will influence the density of water and consequently, they will induce flow through the baroclinic pressure gradient. In this way, transport of constituents and water flow are coupled. Examples are salt intrusion in an estuary, sediment transport in turbidity flows and transport of dissolved matter in lakes and rivers.

Only the background temperature is considered in the model, and the heat exchange flux at the air-water interface is not taken into account.

With respect to the sediment transport, the following assumptions are made.

- Only suspended load is modelled; bed load is not taken into account.
- In the depth-averaged mode, there is no mass exchange of suspended sediment between the bed and the flow.
- The intergranular interactions are excluded.

The inclusion of transport of constituent in SWASH must be done by means of an input grid for each constituent. Such an input grid represents the ambient or background concentration of the corresponding constituent as an initial state, while it provides information along the open boundaries of the computational domain. The use of this information to impose a boundary condition for constituent depends on the flow direction. At inflow, the concentration is prescribed using this information delivered by the input grid. At outflow, the concentration is determined solely by the concentration in upstream part of the domain due to pure advection. It is assumed that the area of interest in which transport phenomena occur is far away from the open boundaries. The ambient concentration at open boundaries is therefore supposed to be steady state. However, for unsteady salt intrusion in a tidal inlet, the concentration is prescribed at inflow by means of a so-called Thatcher-Harleman boundary condition. In this way, a smooth (sinusoidal) transition from the outflow concentration to the inflow boundary condition can be described; see also command `TRANSPORT`.

It is advised to make an input grid so large that it completely covers the computational grid. Otherwise, SWASH assumes that in the region outside the input grid, the constituent equals to the value at the nearest boundary of the input grid (lateral shift of that boundary).

Finally, both the initial state and boundary conditions of any constituent may be vary in the vertical direction. This needs to be provided to SWASH with an input grid for each vertical layer.

2.5.5 Output grids

SWASH can provide output on uniform, rectilinear grids that are independent from the input grids and from the computational grid. In the computation with an orthogonal curvilinear computational grid, curvilinear output grids are available in SWASH. An output grid has to be specified by the user with an arbitrary resolution, but it is of course wise to choose a resolution that is fine enough to show relevant spatial details. It must be pointed out that the information on an output grid is obtained from the computational grid by bi-linear interpolation (output always at computational time level). This implies that some inaccuracies are introduced by this interpolation. It also implies that bottom or wind information on an output plot has been obtained by interpolating twice: once from the input grid to the computational grid and once from the computational grid to the output grid. If the input, computational and output grids are identical, then no interpolation errors occur.

In the regions where the output grid does not cover the computational grid, SWASH assumes output values equal to the corresponding exception value. For example, the default exception value for the surface elevation is -99. The exception values of output quantities can be changed by means of the `QUANTITY` command.

Output can be requested at regular intervals starting at a given time always at computational times.

2.6 Boundary conditions

The boundaries of the computational grid in SWASH are either land, beach or water. SWASH provides the following specification of boundary conditions:

- different wavemakers:
  - Regular waves by means of Fourier series or time series.
  - Irregular unidirectional waves by means of 1D spectrum. The spectrum may be obtained from observations or by specifying a parametric shape (Pierson-Moskowitz, Jonswap or TMA).

  - Irregular multidirectional waves by means of 2D spectrum. The spectrum may be obtained from a SWAN run or by specifying a parametric shape (Pierson-Moskowitz, Jonswap or TMA) while the directional spreading can be expressed with the well-known cosine power or in terms of the directional standard deviation.

- velocity or discharge,
- periodic boundary conditions,
- full reflection at closed boundaries or solid walls,
- Riemann invariants,
- Sommerfeld or radiation condition,
- internal wave generation, and
- sponge layers.

When imposing irregular waves at a boundary segment by means of a spectrum it is assumed that the variation of the depth along the boundary segment is slowly.

SWASH has the option to make a computation that is nested in SWAN. In such a run, SWASH interpolates the locations, as specified in the SWAN run with `POINTS` or `CURVE`, to the user-defined boundary, either side or segment (see command `BOUNDCOND`), of the concerning SWASH run. The SWAN spectra are written to those locations using the SWAN command `SPECOUT`. These wave spectra are employed as boundary conditions using the `SPECSWAN` command. It is assumed that the wave spectra are stationary. Also, both SWASH and SWAN runs must used the same coordinate system, either Cartesian or spherical.

2.7 Time and date notation

SWASH employs the following time notation: `hhmmss.msc` with *hh*, *mm*, *ss* and *msc* denoting hours, minutes, seconds and milliseconds, respectively. Alternatively, SWASH can run for dates

- between the years 0 and 9999, if ISO-notation is used in the input (recommended) or
- between the years 1931 and 2030 if two-digit code for years is used (formats 2-6 in every command that contains moments in time).

2.8 Troubleshooting

Sometimes SWASH produces an error message concerning an instability due to the fact that the water level is below the bottom level and stops. It is general difficult to find the cause of this problem. However, some suggestions about possible reasons and what to do in such cases are given below.

- Checking of the input should always be done at first. It is important that dimensions, model parameters, numerical parameters and boundary conditions are given in a correct manner. Also, consult Chapter 5, if necessary. In any case, check the `PRINT` file.

- If all input is correct and the model is supposed to converge then the common measure is always to reduce the time step (see command `COMPUTE`). Especially in the case of a transient or spin up a small time step may be necessary. Since this is the easiest way to overcome problems, it is always a good practice to start with this measure. Note that sometimes the maximum Courant number should be decrease as well (see command `TIMEI EXPL [cflhig]`).

- Sometimes instabilities may be due to wiggles in the solution. A possible remedy is to choose an upwind discretization or even the first order upwinding. Another improvement might be due to choosing a flux limiter (e.g. minmod).

- Sometimes the solution of the Poisson pressure equation can not be found as the BiCGSTAB solver did not converged. This may happen, for instance, when a considerable number of layers ($\sim$ 30) is involved. In such a case the choice for the ILU preconditioner might be a good alternative (see command `NONHYDrostatic ... PRECond ILU`).

Chapter 3

Input and output files

3.1 General

SWASH is one single computer program. The names of the files provided by the user should comply with the rules of file identification of the computer system on which SWASH is run. In addition: SWASH does not permit file names longer than 36 characters. Moreover, the maximum length of the lines in the input files for SWASH is 180 positions.

The user should provide SWASH with a number of files (input files) with the following information:

- a file containing the instructions of the user to SWASH (the command file),
- file(s) containing: grid, bottom, (initial) current and water level, friction, porosity, and wind and pressure (if relevant) and
- file(s) containing boundary conditions.

3.2 Input / output facilities

To assist in making the command file, an edit file is available to the user (see Appendix C). In its original form this file consists only of comments; all lines begin with exclamation mark. In the file, all commands as given in this User Manual (Chapter 4) are reproduced as mnemonics for making the final command file. Hence, the user does not need to consult the User Manual every time to check the correct spelling of keywords, order of data, etc. The user is advised to first copy the edit file (the copy file should have a different name) and then start typing commands between the comment lines of the edit file.

SWASH is fairly flexible with respect to output processing. Output is available for many different quantities. However, the general rule is that output is produced by SWASH only at the user's request. The instructions of the user to control output are separated into three categories:

- Definitions of the geographic location(s) of the output. The output locations may be either on a grid, or along user specified lines (e.g., a given depth contour line) or at individual output locations.
- Times for which the output is requested.
- Type of output quantities (water level, velocity or discharge, pressure, etc.).

3.3 Print file and error messages

SWASH always creates a print file. Usually the name of this file is identical to the name of the command file of the computations with the extension (.SWS) replaced with (.PRT). Otherwise, it depends on the batch file that is used by the user. Consult the Implementation Manual for more information.

The print file contains an echo of the command file, an overview of the actual physical and numerical parameters to be used in the simulation run, and possibly warning and error messages. These messages are usually self-explanatory. The print file also contains computational results if the user so requests (with command `BLOCK` or `TABLE`).

IN ANY CASE, ALWAYS CHECK THE PRINT FILE!

Chapter 4

Description of commands

4.1 List of available commands

The following commands are available to users of SWASH (to look for the commands quickly, see table of contents and index).

Start-up commands

(a) Start-up commands:

`PROJECT`	title of the problem to be computed
`SET`	sets values of certain general parameters
`MODE`	requests a 1D-mode / 2D-mode of SWASH
`COORD`	to choose between Cartesian and spherical coordinates

Commands for model description

(b) Commands for computational grid:

`CGRID`	defines dimensions of computational grid
`READGRID`	reads an orthogonal curvilinear or unstructured computational grid
`VERT`	defines vertical grid schematisation

(c) Commands for input fields:

`INPGRID`	defines dimensions of e.g. bottom, porosity and friction grids
`READINP`	reads input fields
`INPTRAN`	defines dimensions of grids for transport of constituents

READTRA	reads stationary input fields of transport constituents
INPAMBI	defines dimensions of grids for ambient current
READAMB	reads stationary input fields of ambient current

(d) Commands for initial and boundary conditions:

INITIAL	specifies an initial flow field and turbulence quantities
BOUND	specifies boundary conditions at the boundaries of domain
SOURCE	generates waves within domain using (mass) source function
SPONGE	defines sponge layers
FLOAT	specifies some parameters for floating objects

(e) Commands for physics:

WIND	specifies wind speed, direction and wind drag
FRIC	specifies bottom friction
VISC	includes horizontal and/or vertical turbulent eddy viscosity
POROS	includes effects of porous structures
VEGET	activates wave damping induced by aquatic vegetation
CORI	includes effects of the Coriolis force
TRANSP	specifies some relevant parameters for transport of constituents
BRE	controls wave breaking
AMB	specifies ambient current and mean water level

(f) Commands for numerics:

NONHYD	to choose approximation for nonhydrostatic pressure
DISCRET	to choose spatial discretization
BOTCEL	to determine bottom values in cell centers
TIMEINT	to choose time integration

Output commands

(g) Commands for output locations:

FRAME	defines an output frame (a regular grid)
GROUP	defines an output group (for regular and curvilinear grids)
CURVE	defines an output curve

`RAY`	defines a set of straight output lines (rays)
`ISOLINE`	defines a depth or bottom contour (for output along that contour)
`POINTS`	defines a set of individual output points

(h) Commands to write or plot output quantities:

`QUANTITY`	defines properties of output quantities
`OUTPUT`	influence format of block and/or table
`BLOCK`	requests a block output (geographic distribution)
`TABLE`	requests a table output (set of locations)

(i) Commands to write or plot intermediate results:

`TEST`	requests an output of intermediate results for testing purposes

Lock-up commands

(j) Commands to lock-up the input file:

`COMPUTE`	starts a computation
`STOP`	end of user's input

4.2 Sequence of commands

SWASH executes the above command blocks (a,...,j) in the above sequence except (f), (i) and (j). The commands of the blocks (f) and (i) may appear anywhere before block (j), except that `TEST POINTS` must come after `READINP BOTTOM`. The commands of block (j) may appear anywhere in the command file (all commands after `COMPUTE` are ignored by SWASH, except `STOP`). A sequence of commands of block (g) is permitted (all commands will be executed without overriding). Also a sequence of commands of block (h) is permitted (all commands will be executed without overriding).

Within the blocks the following sequence is to be used:

In block (a) : no prescribed sequence in block
In block (b) : `READGRID` after `CGRID`, and `VERT` after `CGRID`
In block (c) : `READINP` after `INPGRID` (repeat both in this sequence for each quantity)
In block (d) : no prescribed sequence in block

In block (e) : no prescribed sequence in block
In block (f) : no prescribed sequence in block
In block (g) : `ISOLINE` after `RAY` (`ISOLINE` and `RAY` can be repeated independently)
In block (h) : no prescribed sequence in block
In block (i) : no prescribed sequence in block
In block (j) : `STOP` after `COMPUTE`

It must be noted that a repetition of a command may override an earlier occurrence of that command.

Many commands provide the user with the opportunity to choose an option (e.g. discretization scheme) or assign values to coefficients (e.g. bottom friction coefficient). If the user does not use such option SWASH will use a default value.

Some commands cannot be used in 1D-mode and in case of unstructured grids (see individual command descriptions below).

4.3 Command syntax and input / output limitations

The command syntax is given in Appendix B.

Limitations:

- The maximum length of the input lines is 180 characters.
- The maximum length of the file names is 36 characters.
- The maximum number of file names is 99. This can be extended (edit the file `swashinit` to change highest unit number of 99 to a higher number).

4.4 Start-up

```
PROJect 'name' 'nr'

        'title1'

        'title2'

        'title3'
```

With this required command the user defines a number of strings to identify the SWASH run (project name e.g., an engineering project) in the print and plot file.

`'name'`	is the name of the project, at most 16 characters long. Default: blanks
`'nr'`	is the run identification (to be provided as a character string; e.g. the run number) to distinguish this run among other runs for the same project; it is at most 4 characters long. It is the only required information in this command.
`'title1'`	is a string of at most 72 characters provided by the user to appear in the output of the program for the user's convenience. Default: blanks
`'title2'`	same as `'title1'`.
`'title3'`	same as `'title1'`.

```
SET [level] [nor] [depmin] [maxmes] [maxerr] [seed]                           &
    [grav] [rhowat] [temp] [salinity] [dynvis] [rhoair] [rhosed]              &
    [cdcap] [prmean] [backvisc] [kappa] [latitude]                            &

     |    NAUTical   |
    <                 > [outlev]
     | -> CARTesian  |
```

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

`[level]` still water level (in m).
Default: `[level]`=0.

`[nor]` direction of North with respect to the $x-$axis (measured counterclockwise); default `[nor]`$= 90^o$, i.e. $x-$axis of the problem coordinate system points East.
When spherical coordinates are used (see command `COORD`) the value of `[nor]` may not be modified.

`[depmin]` threshold water depth (in m). Any grid point with water depth smaller than `[depmin]` is taken out of the computation.
Default: `[depmin]` = 0.00005

`[maxmes]` maximum number of error messages (not necessarily the number of errors!) during the computation at which the computation is terminated. During the computational process messages are written to the `PRINT` file.
Default: `[maxmes]` = 200

`[maxerr]` during pre-processing SWASH checks input data. Depending on the severity

of the errors encountered during this pre-processing, SWASH does not start computations. The user can influence the error level above which SWASH will not start computations (at the level indicated the computations will continue). The error level [maxerr] is coded as follows:
1 : warnings,
2 : errors (possibly automatically repaired or repairable by SWASH),
3 : severe errors.
Default: [maxerr] = 1

[seed] is the seed of the random number generator used to select phase angles for the different Fourier components. By selecting different seeds different time series can be obtained from the same wave spectrum (see command BOUNDCOND).
Default: [seed] = 12345678.

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

[rhowat] is the density of water (in kg/m^3).
Default: [rhowat] = 1000.
Note: this is usually the reference density. The actual density depends on the temperature and salinity of water.

[temp] is the (ambient) temperature of water (in ^{o}C).
Default: [temp] = 14.

[salinity] is the (ambient) salinity of water (in ppt).
Default: [salinity] = 31.

[dynvis] is the dynamical viscosity of water (in kg/ms).
Default: [dynvis] = 0.001

[rhoair] is the density of air (in kg/m^3).
Default: [rhoair] = 1.205

[rhosed] is the density of sediment (in kg/m^3).
Default: [rhosed] = 2650.

[cdcap] is the maximum value for the wind drag coefficient. A value of [cdcap] = 99999 means no cutting off the drag coefficient. A suggestion for this parameter is [cdcap] = 2.5× 10^{-3}.
Default: [cdcap] = 99999.

[prmean] is the mean atmospheric pressure (in N/m^2). This will be used to correct the water level along the water level boundaries so that it is consistent with the local atmospheric pressure. A value of [prmean] = -1 means no correction.
Default: [prmean] = -1.

[backvisc] is the background viscosity (in m^2/s). It may be convenient to specify this ambient viscosity to account for all forms of unresolved vertical mixing. The value must be small compared to the vertical viscosity calculated by the standard $k - \varepsilon$ model. Experiences suggest a value of 10^{-4} to 10^{-3} m^2/s.
Default: [backvisc] = 0.

[kappa] is the Von Karman constant.

Default: `[kappa]` = 0.4

`[latitude]` is the geographical position of the model domain expressed as the angle of latitude (in degrees North).
Default: `[latitude]` = 52.

`NAUTICAL` indicates that the Nautical convention for wind and current direction (SWASH input and output) and for incident wave boundary condition will be used instead of the default Cartesian convention. For definition, see Section 2.4 or Appendix A.

`CARTESIAN` indicates that the Cartesian convention for wind and current direction (SWASH input and output) and for incident wave boundary condition will be used. For definition, see Section 2.4 or Appendix A.

`[outlev]` defines the amount of output for iterative solvers.
It is coded as follows:
0 : no output,
1 : only fatal errors will be printed,
2 : gives output concerning the iteration process,
3 : additional information about the iteration is printed.
Default: `[outlev]` = 0

```
      |                 |     |-> TWODimensional |
MODE <   NONSTationary   >   <                    >
      |                 |     |   ONEDimensional |
```

With this optional command the user indicates that the run will be either one-dimensional (1D-mode, flume) or two-dimensional (2D-mode, basin).

Note that the keyword `NONSTATIONARY` is obliged.

The default option is `NONSTATIONARY TWODIMENSIONAL`.

```
             | -> CARTesian
COORDINATES <                 | -> CCM
             |    SPHErical   <
                              |  QC
```

Command to choose between Cartesian and spherical coordinates (see Section 2.4).

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

SPHERICAL all coordinates of locations and geographical grid sizes are given in degrees; x is longitude with $x = 0$ being the Greenwich meridian and $x > 0$ is East of this meridian; y is latitude with $y > 0$ being the Northern hemisphere. Input and output grids have to be oriented with their $x-$axis to the East; mesh sizes are in degrees. All other distances are in meters.

CCM defines the projection method in case of spherical coordinates. CCM means central conformal Mercator. The horizontal and vertical scales are uniform in terms of cm/degree over the area shown. In the center of the scale is identical to that of the conventional Mercator projection (but only at that center). The area in the projection center is therefore exactly conformal.

QC the projection method is quasi-cartesian, i.e. the horizontal and vertical scales are equal to one another in terms of cm/degree.

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

Note that in case of spherical coordinates regular grids must always be oriented E-W, N-S, i.e. `[alpc]`=0, `[alpinp]`=0, `[alpfr]`=0 (see commands `CGRID`, `INPUT GRID` and `FRAME`, respectively). In addition, spherical coordinates are not supported in case of unstructured grids.

4.5 Model description

4.5.1 Computational grid

```
       | -> REGular [xpc] [ypc] [alpc] [xlenc] [ylenc] [mxc] [myc] |
       |                                                           |
CGRID <     CURVilinear [mxc] [myc]  (EXCeption  [xexc]  [yexc])    >  &
       |                                                           |
       |    UNSTRUCtured                                           |

                      | -> X
          REPeating  <
                      |  Y
```

With this required command the user defines the geographic location, size, resolution and orientation of the computational grid in the problem coordinate system (see Section 2.5.2) in case of a uniform, rectilinear computational grid, an orthogonal curvilinear grid or an unstructured triangular mesh. The origin of the regular grid and the direction of the positive $x-$axis of this grid can be chosen arbitrary by the user.

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

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

`UNSTRUCTURE` this option indicates that the computational grid is to be taken as unstructured. The user must provide the coordinates of the vertices and the numbering of triangles with the associated connectivity table with vertices with command `READGRID UNSTRUC`.

`[xpc]` geographic location of the origin of the computational grid in the problem coordinate system ($x-$coordinate, in m). See command `COORD`.
Default: `[xpc]` = 0.0 (Cartesian coordinates).
In case of spherical coordinates there is no default, the user must give a value.

`[ypc]` geographic location of the origin of the computational grid in the problem coordinate system ($y-$coordinate, in m). See command `COORD`.
Default: `[ypc]` = 0.0 (Cartesian coordinates).
In case of spherical coordinates there is no default, the user must give a value.

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

`[xlenc]` length of the computational grid in $x-$direction (in m). In case of spherical coordinates `[xlenc]` is in degrees.

`[ylenc]` length of the computational grid in $y-$direction (in m). In 1D-mode, `[ylenc]` should be 0. In case of spherical coordinates `[ylenc]` is in degrees.

`[mxc]` number of meshes in computational grid in $x-$direction for a uniform, rectilinear grid or $\xi-$direction for a curvilinear grid (this number is one less than the number of grid points in this domain!).

`[myc]` number of meshes in computational grid in $y-$direction for a uniform, rectilinear grid or $\eta-$direction for a curvilinear grid (this number is one less than the number of grid points in this domain!). In 1D-mode, `[myc]` should be 0.

`EXCEPTION` only available in the case of a curvilinear grid. If certain grid points are to be ignored during the computation (e.g. land points that remain dry i.e. no flooding; saving computer time and memory), then this can be indicated by identifying these grid points in the file containing the grid point coordinates (see command `READGRID`). For an alternative, see command `INPGRID BOTTOM`.

`[xexc]` the value which the user uses to indicate that a grid point is to be ignored in the computations (this value is provided by the user at the location of the $x-$coordinate considered in the file of the $x-$coordinates, see command

READGRID COOR). Required if the option EXCEPTION is used.
Default: [xexc] = 0.0

[yexc] the value which the user uses to indicate that a grid point is to be ignored in the computations (this value is provided by the user at the location of the $y-$coordinate considered in the file of the $y-$coordinates, see command READGRID COOR). Required if the option EXCEPTION is used.
Default: [yexc] = [xexc]

REPEATING this option indicates that the grid is repeated in one specific direction. It means that information leaving at one end of the domain enters at the opposite end. So, the current or wave field is periodic in one direction with the length of the domain in that direction.
NOT FOR 1D-MODE AND UNSTRUCTURED GRIDS.

X the computational grid is repeated in $x-$ or $\xi-$direction.
This is default.

Y the computational grid is repeated in $y-$ or $\eta-$direction.

For illustration of a regular grid with its dimensions, see Figure 4.1.

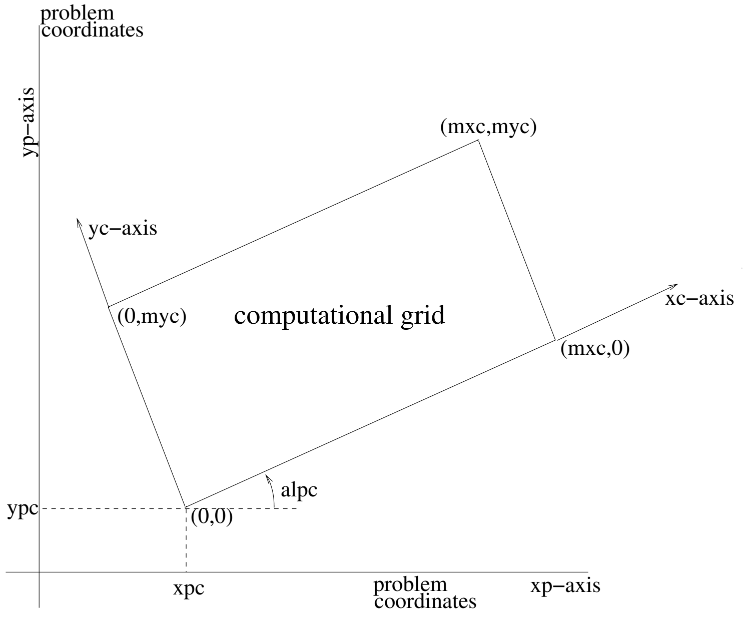

Figure 4.1: Coordinates of the origin [xpc] and [ypc], the orientation [alpc] and the grid point numbering of the computational grid with respect to the problem coordinate system. Note that in case of spherical coordinates the $xc-$ and $xp-$axes both point East.

```
READgrid COORdinates [fac] 'fname' [idla] [nhedf] [nhedvec] &
```

```
                        | -> FREe                |
                        |                        |
                        |             | 'form' |  |
                       <     FORmat  <          >  >
                        |             | [idfm] |  |
                        |                        |
                        |    UNFormatted         |
```

CANNOT BE USED IN 1D-MODE.

This command `READGRID COOR` must follow a command `CGRID CURV`. With this command (required if the computational grid is orthogonal curvilinear; not allowed in case of a regular grid) the user controls the reading of the coordinates of the computational grid points. These coordinates must be read from a file as a vector ($x-$coordinate, $y-$coordinate of each single grid point). See command `READINP` for the description of the options in this command `READGRID`. SWASH will check whether all angles in the grid are > 0 and < 180 degrees. If not, it will print an error message giving the coordinates of the grid points involved. It is recommended to use grids with angles between 45 and 135 degrees.

```
                            | -> TRIAngle |
READgrid UNSTRUCtured  <                   > 'fname'
                            |    EASYmesh |
```

CANNOT BE USED IN 1D-MODE.

This command `READGRID UNSTRUC` must follow a command `CGRID UNSTRUC`. With this command (required if the computational grid is unstructured which must be Delaunay; not allowed in case of a regular or curvilinear grid) the user controls the reading of the (x, y) coordinates of the vertices including boundary markers and a connectivity table for triangles (or elements). This table contains three corner vertices around each triangle in counterclockwise order. This information should be provided by a number of files generated by one of the following grid generators currently supported by SWASH:

- Triangle (*http://www.cs.cmu.edu/afs/cs/project/quake/public/www/triangle.html*)
- Easymesh (*https://web.mit.edu/easymesh_v1.4/www/easymesh.html*)

These generators produce Delaunay-type grids. After setting up the vertices and the connectivity tables for cells and faces (automatically done in SWASH), SWASH will print some information concerning the used mesh, among others, number of vertices, cells and faces and minimum and maximum gridsizes.

`TRIANGLE` the necessary grid information is read from two files as produced by Triangle. The `.node` and `.ele` files are required. The basename of these files must be indicated with parameter `'fname'`.

`EASYMESH` the necessary grid information is read from two files as produced by Easymesh. The `.n` and `.e` files are required. The basename of these files must be indicated with parameter `'fname'`.

`'fname'` basename of the required files, i.e. without extension. Only meant for Triangle and Easymesh.

```
                                | M
VERTical [kmax] < [thickness] <            >
                                | -> PERC
```

With this optional command the user indicates that the run will be in multi-layered mode and controls the distribution of vertical layers.

`[kmax]` number of vertical layers.

`[thickness]` layer thickness (in meters or as percentage of water depth).

`M` thickness is given in meters. This layer has a fixed thickness.

`PERC` thickness is given as percentage. This layer has a variable thickness. This is default.

Notes:

- The layers are numbered from top (=1) to bottom (=`[kmax]`).
- If no `[thickness]` is given, the layers are distributed equidistantly.
- The sum of thicknesses defined as percentages must be 100.
- At least one layer with variable thickness must be given.
- The layers interfaces are equivalent to the well-known sigma planes, if all the layers have a variable thickness.
- For short wave simulations, it is advised to choose variable thicknesses only, preferably equidistantly distributed.
- Layers with a fixed thickness are not supported in case of unstructured grids.

4.5.2 Input grids and data

```
             | BOTtom       |
             |              |
             | WLEVel       |
             |              |
             |  | CURrent   |
             | <            |
             |  | VX        |
             |  | VY        |
             |              |
             | FRiction     |
             |              |
             |  | WInd      |
             | <            |
             |  | WX        |
             |  | WY        |
INPgrid     (<               >)                                            &
             | PRessure     |
             |              |
             | CORIolis     |
             |              |
             | POROsity     |
             |              |
             | PSIZe        |
             |              |
             | HSTRUCture   |
             |              |
             | NPLAnts      |
             |              |
             | DRAFt        |
             |              |
             | LABel        |

   | -> REGular [xpinp] [ypinp] [alpinp] [mxinp] [myinp] [dxinp] [dyinp] |
   |                                                                     |
  <     CURVilinear  STAGgered                                            > &
   |                                                                     |
   |    UNSTRUCtured                                                     |

    (EXCeption  [excval])                                                  &
```

```
                                   | -> Sec  |
(NONSTATionary [tbeginp] [deltinp] <     MIn    >  [tendinp])
                                   |    HR   |
                                   |    DAy  |
```

OPTION CURVILINEAR AND UNSTRUCTURED NOT FOR 1D-MODE.

With this required command the user defines the geographic location, size and orientation of an input grid and also the time characteristics of the variable if it is not stationary. If this is the case (the variable is not stationary), the variable should be given in a sequence of fields, one for each time step `[deltinp]`. The actual reading of values of bottom, wind, pressure, etc. from file is controlled by the command `READINP`.
This command `INPGRID` must precede the following command `READINP`.

There can be different grids for bottom level (`BOTTOM`), current (`CURRENT`), bottom friction coefficient (`FRICTION`), wind velocity (`WIND`), atmospheric pressure (`PRESSURE`), Coriolis parameter (`CORIOLIS`), porosity layers (`POROSITY`), stone diameters (`PSIZE`), heights (`HSTRUCTURE`) of porous structures and vegetation density (`NPLANTS`) and drafts/labels of floating objects like a vessel or WECs (`DRAFT` and `LABEL`).

If the current velocity components are available on different grids, then option `VX`, `VY` can define these different grids for the $x-$ and $y-$component of the current, respectively (but the grids must have identical orientation). Different grids for `VX` and `VY` may be useful if the data are generated by a circulation model using a staggered grid. The same holds for the wind velocity components, i.e. `WX` and `WY`.

In the case of a regular grid (option `REGULAR` in the `INPGRID` command) the current and wind velocity vectors are defined with the $x-$ and $y-$component of the current or wind vector with respect to the $x-$axis of the input grid. In case of an orthogonal curvilinear grid (option `CURVILINEAR` in the `INPGRID` command) the current and wind velocity vectors are defined with the $x-$ and $y-$component of the current or wind vector with respect to the $x-$axis of the problem coordinate system (see Figure 4.1).

In case of an unstructured grid (option `UNSTRUC` in the `INPGRID` command) the current and wind velocity vectors are defined with the $x-$ and $y-$component of the current or wind vector with respect to the $x-$axis of the problem coordinate system (see Figure 4.1).

Porosity layers can be placed inside the computational domain to simulate reflection and transmission effects of porous structures such as rubble mound breakwaters and jetties. Porosity is defined as the volumetric porosity of the structures and its value is in between 0 and 1. A porosity value of 0.45 is typically used for breakwaters. A small value (< 0.1)

should be interpreted as impermeable, like walls and dams. Also structure heights (relative to the bottom) can be specified so that both submerged and emerged breakwaters is allowed.

If the user specifies an input grid for the atmospheric pressure, then an input grid for wind must be included as well. Both space varying wind and pressure may be read from a meteorological file.

For wind velocity, friction coefficient, Coriolis parameter, grain size, height of porous structures and vegetation density it is also possible to use a constant value over the computational field (see commands `WIND`, `FRICTION`, `CORIOLIS`, `POROSITY` and `VEGETATION`, respectively). No grid definition for wind, friction, Coriolis factor, grain size, structure height or vegetation density is then required.

Note that in case of options `BOTTOM`, `CORIOLIS`, `POROSITY`, `PSIZE`, `HSTRUCTURE`, `NPLANTS` and `LABEL` only stationary input field is allowed.

If the computational grid is unstructured, the input grids can be either regular or identical to the used computational grid.

If land points remain dry during the computation (no flooding!), then these points can be ignored. In this way, simulation time and internal memory can be saved. This can be done by indicating bottom level in these points as exception value. See command `INPGRID BOTTOM EXCEPTION`. For parallel runs using MPI, an exception value for bottom levels should be prescribed in order to have a good load-balancing!

Exception value for bottom levels can also be used to take into account dams, screens, quays or jetties in the domain. In addition, they may represent small obstacles with subgrid dimensions that possibly influence the local flow pattern. In this way, the user can defined a line of thin dams that separate the flow on both sides.

See Section 2.5.3 for more information on input grids.

`BOTTOM` defines the input grid of the bottom level. (For the definition of the bottom level, see command `READINP`).

`WLEV` defines the input grid of the water level. (For the definition of the water level, see command `READINP`).

`CURRENT` defines the input grid of the current field (same grid for $x-$ and $y-$components).

`VX` defines the input grid of the $x-$component of the current field (different grid than $y-$component but same orientation).

`VY` defines the input grid of the $y-$component of the current field (different grid than $x-$component but same orientation).

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

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

WX defines the input grid of the $x-$component of the wind velocity field (different grid than $y-$component but same orientation).

WY defines the input grid of the $y-$component of the wind velocity field (different grid than $x-$component but same orientation).

PRESSURE defines the input grid of the atmospheric pressure. (For the definition of the atmospheric pressure, see command READINP).

CORIOLIS defines the input grid of the Coriolis parameter.
If neither of the commands CORIOLIS and READINP CORIOLIS is used it is assumed that there is no Coriolis effect.

POROSITY defines the input grid of the porosity distribution.
If neither of the commands POROSITY and READINP POROSITY is used it is assumed that there is no porous structure.

PSIZE defines the input grid of the grain sizes of porous structures.

HSTRUCTURE defines the input grid of the heights of porous structures.

NPLANTS defines input grid of the horizontally varying vegetation density (defined in command VEGETATION).

DRAFT defines the input grid of the draft of floating objects. (For the definition of the draft, see command READINP).

LABEL defines the input grid of the labelling of rigid bodies. (For the definition of the labels, see command READINP).

REGULAR means that the input grid is uniform and rectangular.

[xpinp] geographic location ($x-$coordinate) of the origin of the input grid in problem coordinates (in m) if Cartesian coordinates are used or in degrees if spherical coordinates are use (see command COORD).
Default: [xpinp] $= 0$. In case of spherical coordinates there is no default, the user must give a value.

[ypinp] geographic location ($y-$coordinate) of the origin of the input grid in problem coordinates (in m) if Cartesian coordinates are used or in degrees if spherical coordinates are use (see command COORD).
Default: [ypinp] $= 0$. In case of spherical coordinates there is no default, the user must give a value.

[alpinp] direction of the positive $x-$axis of the input grid (in degrees, Cartesian convention). See command COORD.
Default: [alpinp] $= 0$.

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

[myinp] number of meshes in $y-$direction of the input grid (this number is one less

than the number of grid points in this direction!).
In 1D-mode, [myinp] should be 0.

[dxinp] mesh size in $x-$direction of the input grid,
in m in case of Cartesian coordinates or
in degrees if spherical coordinates are used, see command COORD.

[dyinp] mesh size in $y-$direction of the input grid,
in m in case of Cartesian coordinates or
in degrees if spherical coordinates are used, see command COORD.
In 1D-mode, [dyinp] may have any value.
Default: [dyinp] = [dxinp].

CURVILINEAR means that the input grid is curvilinear; this option is available only if the computational grid is curvilinear as well. The input grid is identical to the computational grid.
NOT FOR 1D-MODE.

STAGGERED means that the input grid of a grid-oriented (not Cartesian!) velocity component u or v is staggered in $y-$ or $x-$direction, respectively.
The velocity components are given in their points of definition according to the Arakawa C-grid staggering.
NOT FOR 1D-MODE.

UNSTRUCTURE means that the input grid is unstructured; this option is available only if the computational grid is unstructured as well. The input grid must be identical to the computational grid.
NOT FOR 1D-MODE.

EXCEPTION certain points inside the given grid that are to be ignored during the computation can be identified by means of an exception value as given in the corresponding input file as controlled by the command READINP.
NOT FOR 1D-MODE.

[excval] exception value; required if the option EXCEPTION is used.
Note: if [fac] $\neq 1$ (see command READINP), [excval] must be given as [fac] times the exception value.

NONSTATION the variable is nonstationary (given in a time sequence of fields).
NOT FOR 1D-MODE.

[tbeginp] begin time of the first field of the variable, the format is:

1 :	ISO-notation	19870530.153000
2 :	(as in HP compiler)	'30−May−87 15:30:00'
3 :	(as in Lahey compiler)	05/30/87.15:30:00
4 :		15:30:00
5 :		87/05/30 15:30:00'
6 :	as in WAM	8705301530
7 :		153000.000

This format is installation dependent. See Implementation Manual or ask the person who installed SWASH on your computer. Default is option 7.

[deltinp] time interval between fields, the unit is indicated in the next option:

SEC	unit seconds
MIN	unit minutes
HR	unit hours
DAY	unit days

`[tendinp]` end time of the last field of the variable, the format is:

1 :	ISO-notation	19870530.153000
2 :	(as in HP compiler)	’30−May−87 15:30:00’
3 :	(as in Lahey compiler)	05/30/87.15:30:00
4 :		15:30:00
5 :		87/05/30 15:30:00’
6 :	as in WAM	8705301530
7 :		153000.000

This format is installation dependent. See Implementation Manual or ask the person who installed SWASH on your computer. Default is option 7.

```
          |  BOTtom      |
          |              |
          |  WLEVel      |
          |              |
          |  CURrent     |
          |              |
          |  FRiction    |
          |              |
          |  WInd        |
          |              |
          |  PRessure    |
          |              |           |  'fname1'            |
READinp  <   CORIolis     >  [fac]  <                        >   [idla]     &
          |              |           |  SERIes   'fname2'   |
          |  POROsity    |
          |              |
          |  PSIZe       |
          |              |
          |  HSTRUCture  |
          |              |
          |  NPLAnts     |
          |              |
          |  DRAFt       |
          |              |
          |  LABel       |
                                           | -> FREe                    |
                                           |                            |
                                           |                | 'form' |  |
           [nhedf] ([nhedt]) ([nhedvec])  <      FORmat     <          >  >
                                           |                | [idfm] |  |
                                           |                            |
                                           |    UNFormatted             |
```

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

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

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

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

`CURRENT` rectilinear (curvilinear) input grid: with this option the user indicates that the $x-$ and $y-$component ($\xi-$ and $\eta-$component) (in m/s) are to be read from one and the same file (with one `READINP` command). With this option SWASH reads first all $x-$components ($\xi-$components), and then all $y-$components ($\eta-$components) (see `[idla]`). The first component ($x-$ or $\xi-$component) is always eastward oriented and the second one ($y-$ or $\eta-$component) is always northward oriented. There is one exception: in case of rotated rectilinear grid, the $x-$ and $y-$components are taken along the direction of the grid lines.
unstructured input grid: with this option the user indicates that the $x-$ and $y-$component (in m/s) are to be read from one and the same file (with one `READINP` command). With this option SWASH reads first all $x-$components, and then all $y-$components. The order of these values to be read is identical to that of the unstructured computational grid.

`FRICTION` with this option the user indicates that friction coefficient is to be read from file for Manning or Chezy formula's or Nikuradse roughness height. If the coefficients are constant in space and time: see command `FRICTION`.

`WIND` rectilinear (curvilinear) input grid: with this option the user indicates that the $x-$ and $y-$component ($\xi-$ and $\eta-$component) (in m/s) are to be read from one and the same file (with one `READINP` command). With this option SWASH

reads first all $x-$components ($\xi-$components), and then all $y-$components ($\eta-$components) (see `[idla]`). The first component ($x-$ or $\xi-$component) is always eastward oriented and the second one ($y-$ or $\eta-$component) is always northward oriented. There is one exception: in case of rotated rectilinear grid, the $x-$ and $y-$components are taken along the direction of the grid lines.
If the wind is constant, see command `WIND`.

`PRESSURE` with this option the user indicates that atmospheric pressures (in N/m^2) are to be read from file. Unit can be changed with option `[fac]` (see below).

`CORIOLIS` with this option the user indicates that Coriolis factor (in s^{-1}) is to be read from file. If this parameter is constant then see command `CORIOLIS`.

`POROSITY` with this option the user indicates that volumetric porosity is to be read from file. Porosity values less than 1 indicates the location of porous structures. A value of 1 represents water points. Regions with small porosity values (< 0.1) will be treated as impermeable regions, i.e. land points.

`PSIZE` with this option the user indicates that grain sizes (in m) of porous structures are to be read from file. If the grain size is constant for all porous structures then see command `POROSITY` for specification.

`HSTRUCTURE` with this option the user indicates that heights (in m) of porous structures (relative to the bed level) are to be read from file. If the height is constant for all porous structures then see command `POROSITY` for specification.

`NPLANTS` with this option the user indicates that horizontally varying vegetation density (per m^2) is to be read from file. If the density is constant then see command `VEGETATION` for specifcation.

`DRAFT` with this option the user indicates that the draft of floating object (in m) is to be read from file (draft positive downward relative to an arbitrary horizontal datum level). The sign of the input can be changed with option `[fac]` $= -1$. (see below).

`LABEL` with this option the user indicates that the labels of rigid bodies is to be read from file. It assigns integer values to each of the bodies, starting from 1. See command `BODY DIMENSION` for further specification of body dimensions.

`[fac]` SWASH multiplies all values that are read from file with `[fac]`. For instance if the bottom levels are given in unit decimeter, one should make `[fac]`=0.1 to obtain levels in m. To change sign of bottom level use a negative value of `[fac]`. Note that `[fac]` = 0 is not allowed!
Default: `[fac]`=1.

`'fname1'` name of the file with the values of the variable.

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

`'fname2'` name of file that contains the names of the files where the variables are given. These names are to be given in proper time sequence. SWASH reads the next file when the previous file end has been encountered. In these files the input should be given in the same format as in the above file `'fname1'` (that implies that a file should start with the start of an input time step).

`[idla]` prescribes the order in which the values of bottom levels and other fields should be given in the file.

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

```
1,myc+1     2,myc+1     ...     mxc+1, myc+1
1,myc       2,myc       ...     mxc+1, myc
...         ...         ...     ...
1,1         2,1         ...     mxc+1, 1
```

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

=3: SWASH reads the map from left to right starting in the lower-left-hand corner of the map. A new line in the map should start on a new line in the file. The lay-out is as follows:

```
1,1         2,1         ...     mxc+1, 1
1,2         2,2         ...     mxc+1, 2
...         ...         ...     ...
1,myc+1     2,myc+1     ...     mxc+1, myc+1
```

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

=5: SWASH reads the map from top to bottom starting in the lower-left-hand corner of the map. A new column in the map should start on a new line in the file. The lay-out is as follows:

```
1,1         1,2         ...     1, myc+1
2,1         2,2         ...     2, myc+1
...         ...         ...     ...
mxc+1,1     mxc+1,2     ...     mxc+1, myc+1
```

=6: as `[idla]`=5 but a new column in the map need not start on a new line in the file.

Default: `[idla]`=1

`[nhedf]` is the number of header lines at the start of the file. The text in the header lines is reproduced in the `PRINT` file created by SWASH (see Section 3.3). The file may start with more header lines than `[nhedf]` because the start of the file is often also the start of a time step and possibly also of a vector variable (each having header lines, see below, `[nhedt]` and `[nhedvec]`).

Default: [nhedf]=0

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

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

FREE with this option the user indicates that the values are to be read with free format. Free format is a standard of the computer programming language FORTRAN. The free format conventions in reading from a file are almost the same as the conventions for the command syntax given elsewhere in this manual; the most important differences are:

1. There are no continuation marks, reading continues until the required number of data has been read, or until a slash (/) is encountered.
2. Input lines can be longer than 80 characters (depending on the operating system of the computer).
3. Comment is not allowed.

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

FORMAT with this option the user indicates that fixed format (FORTRAN convention) is to be used when reading the values from file. The format can be defined in one of two ways, by giving the format number [idfm] or the format string 'form'.

'form' a user–specified format string according to Fortran convention, e.g. '(10X,12F5.0)'.

[idfm] this format number is interpreted as follows:
=1: Format according to BODKAR convention (a standard of the Ministry of Transport and Public Works in the Netherlands).
Format string: (10X,12F5.0).
=5: Format (16F5.0), i.e. an input line consists of 16 fields of 5 places each.
=6: Format (12F6.0), i.e. an input line consists of 12 fields of 6 places each.
=8: Format (10F8.0), i.e. an input line consists of 10 fields of 8 places each.

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

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

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

```
           | SALinity    |
           |             |
INPtrans  <  TEMPerature  >                                            &
           |             |
           | SEDiment    |


   | -> REGular [xpinp] [ypinp] [alpinp] [mxinp] [myinp] [dxinp] [dyinp] |
   |                                                                     |
  <     CURVilinear                                                       > &
   |                                                                     |
   |    UNSTRUCtured                                                     |

    (EXCeption  [excval])                                                  &

    (NONUNIForm  [kmax])
```

OPTION CURVILINEAR NOT FOR 1D-MODE.

With this command the user defines the geographic location, size and orientation of a stationary input grid for the transport of constituent. This input grid thus supplies initial and stationary boundary conditions for the considered constituent. The actual reading of constituent values from file is controlled by the command `READTRANS`.
This command `INPTRANS` must precede the following command `READTRANS`.

There can be different grids for salinity (`SALINITY`), temperature or heat (`TEMPERATURE`) and suspended sediment load (`SEDIMENT`).

See command `INPGRID` for the description of the options in this command `INPTRANS`.

See Section 2.5.4 for more information on (input) grids for transport of constituents.

`NONUNIFORM`	the constituent is non-uniform in vertical.
`[kmax]`	the number of layers representing the number of input fields as given in a sequence (see command `READTRANS`). This number must be equal to the number of vertical layers in multi-layered mode (see command `VERTICAL`) or 1 (i.e. uniform in vertical).

Default: `[kmax]` = 1

```
            |  SALinity    |
            |              |           | 'fname1'          |
READtrans  <   TEMPerature  >  [fac]  <                     >  [idla]       &
            |              |           | LAYers   'fname2' |
            |  SEDiment    |

                                       | -> FREe                   |
                                       |                           |
                                       |                 | 'form' |    |
                              [nhedf]  <     FORmat      <          >    >
                                       |                 | [idfm] |    |
                                       |                           |
                                       |    UNFormatted            |
```

With this command the user controls the reading of initial and boundary values of transport constituents from file. This command `READTRANS` must follow a command `INPTRANS`.

The constituents that can be read are salinity (`SALINITY`) (in ppt, psu or kg/m^3), temperature (`TEMPERATURE`) (in ^{o}C) and suspended sediment load (`SEDIMENT`) (in kg/m^3).

See command `READINP` for the description of the options in this command `READTRANS`.

`LAYERS` with this option (only for multi-layered mode) the user indicates that the names of the files containing the non-uniform constituent are resided in a separate file with name `'fname2'` (see below).

`'fname2'` name of file that contains the names of the files where the constituents are given. These names are to be given in proper sequence, i.e. from top (first layer) to bottom (last layer). SWASH reads the next file when the previous file end has been encountered. In these files the input should be given in the same format as in the above file `'fname1'`.

```
         |   | ACURrent |
         |  <           |
INPamb  <    | AVX       >                                                  &
         |   | AVY      |
        |              |
        |  MWL         |


   | -> REGular [xpinp] [ypinp] [alpinp] [mxinp] [myinp] [dxinp] [dyinp] |
  <                                                                       > &
   |    CURVilinear  STAGgered                                            |

    (EXCeption  [excval])                                                   &

    (NONUNIForm  [kmax])
```

OPTION CURVILINEAR NOT FOR 1D-MODE.

With this command the user defines the geographic location, size and orientation of a stationary input grid of the ambient current, such as riverine, tidal and wind-driven flows, and the associated mean water level. With this input grid the effect of ambient currents on the wave dynamics is accounted for in the simulation. The ambient current (ACURRENT) and the corresponding mean water level (MWL) can be spatially varying but are assumed to be constant in time with respect to the temporal wave motion. The actual reading of values of ambient current and/or mean water level from file is controlled by the command READAMB. This command INPAMB must precede the following command READAMB.

There can be different grids for current (ACURRENT) and mean water level (MWL). Additionally, if the ambient velocity components are available on different grids, then options AVX and AVY can define these different grids for the $x-$ and $y-$component of the ambient current, respectively (but the grids must have identical orientation).

In the case of a regular grid (option REGULAR) the ambient current is defined with the $x-$ and $y-$component of the current vector with respect to the $x-$axis of the input grid. Hence, these velocity components are (input) grid oriented.

In case of an orthogonal curvilinear grid (option CURVILINEAR) the current is defined with the $x-$ and $y-$component of the current vector with respect to the $x-$axis of the problem coordinate system (see Figure 4.1).

See also command INPGRID for the description of the options not described here.

It is also possible to use a constant value over the computational field; see command AMBIENT. No grid definition for ambient current or mean water level is then required.

This command is not supported in case of unstructured grids.

ACURRENT defines the input grid of the ambient current (same grid for $x-$ and $y-$components).

AVX defines the input grid of the $x-$component of the ambient current (different grid than $y-$component but same orientation).

AVY defines the input grid of the $y-$component of the ambient current (different grid than $x-$component but same orientation).

MWL defines the input grid of the mean water level.

NONUNIFORM the ambient current is non-uniform in vertical.

[kmax] the number of layers representing the number of input fields as given in a sequence (see command READAMB). This number must be equal to the number of vertical layers in multi-layered mode (see command VERTICAL) or the number of pressure layers in case of the subgrid approach (see command NONHYD SUBGRID [pmax]) or 1, that is, uniform in vertical. Default: [kmax] = 1

```
          |  ACURrent |          | 'fname1'          |
READamb  <             >  [fac]  <                     >  [idla]           &
          |  MWL      |          | LAYers   'fname2' |


                                          | -> FREe                    |
                                          |                            |
                                          |                | 'form' |   |
                              [nhedf]   <      FORmat      <          >  >
                                          |                | [idfm] |   |
                                          |                            |
                                          |    UNFormatted             |
```

With this command the user controls the reading of ambient current and the associated mean water level from file. This command READAMB must follow a command INPAMB.

See also command READINP for the description of the options not described here.

ACURRENT with this option the user indicates that the $x-$ and $y-$component ($\xi-$ and $\eta-$component) (in m/s) are to be read from one and the same file (with one READAMB command). With this option SWASH reads first all $x-$components ($\xi-$components), and then all $y-$components ($\eta-$components) (see [idla]). The first component ($x-$ or $\xi-$component) is always eastward oriented and the second one ($y-$ or $\eta-$component) is always northward oriented. There is one exception: in case of rotated rectilinear grid, the $x-$ and $y-$components are taken along the direction of the grid lines.

MWL with this option the user indicates that water levels (in m) are to be read from file (water level positive upward relative to the same datum level as used in option READ BOTTOM). Sign of input can be changed with option [fac] $= -1$.

LAYERS with this option (only for multi-layered mode) the user indicates that the names of the files containing the non-uniform ambient current are resided in a separate file with name 'fname2' (see below).

'fname2' name of file that contains the names of the files where the current are given. These names are to be given in proper sequence, i.e. from top (first layer) to bottom (last layer). SWASH reads the next file when the previous file end has been encountered. In these files the input should be given in the same format as in the above file 'fname1'.

4.5.3 Initial and boundary conditions

```
          | -> CONstant [wlev] [vx] [vy] [tke] [epsilon]
          |
INITial  <   ZERO
          |
          |  STEAdy
```

This command can be used to specify the initial values for flow variables.

CONSTANT the initial flow and turbulence quantities are set to a constant.
[wlev] the water level.
[vx] the $u-$component of velocity.
[vy] the $v-$component of velocity.
[tke] the turbulent kinetic energy.
[epsilon] the dissipation rate of turbulent kinetic energy.
ZERO Both the initial water level and velocity components are set to zero.
STEADY If this option is specified, the initial velocities will be derived from the water levels using the Chezy formula for steady flow. This can shorten the spin-up time of the SWASH run and can be meaningful in the case of quasi-steady flow condition (e.g. flow in a river).

```
                   |    PM                |
                   |                      |    | -> SIG |     | -> PEAK  |
BOUnd SHAPespec   <  -> JONswap  [gamma]   >  <          >  <            >   &
```

```
                   |                      |    |    RMS |    |    MEAN  |
                   |    TMA               |

                        | -> POWer      |
                DSPR   <                 >
                        |    DEGRees    |
```

This command `BOUND SHAPESPEC` defines the shape of the spectra (both in frequency and direction) at the open boundary of the computational grid or in the computational domain using source function for internal wave generation in case of parametric spectral input (see either command `BOUNDCOND` or command `SOURCE`).

`PM` Pierson-Moskowitz spectrum will be used.
`JONSWAP` JONSWAP spectrum will be used. This is default.
`[gamma]` peak enhancement parameter of the JONSWAP spectrum.
Default: `[gamma]`=3.3
`TMA` A modified JONSWAP spectrum for finite depth will be used.
`SIG` The significant wave height (for definition, see Appendix A) is used as the characteristic wave height.
This is default.
`RMS` The RMS wave height (for definition, see Appendix A) is used as the characteristic wave height.
`PEAK` The peak period is used as the characteristic wave period.
This is default.
`MEAN` T_{m01} (for definition, see Appendix A) is used as the characteristic wave period.
`DSPR` option for expressing the width of the directional distribution; the distribution function itself is $\cos^m(\theta)$.
`POWER` the directional width is expressed with the power m itself.
This option is default.
`DEGREES` the directional width is expressed in terms of the directional standard deviation of the $\cos^m(\theta)$ distribution (for definition, see Appendix A).

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

```
                       | North |
                       | NW    |
                       | West  |
                       | SW    |   | -> CCW     |
           | -> SIDE  <  South  > <              >          |
           |           | SE    |   | CLOCKWise  |           |
           |           | East  |                            |
           |           | NE    |                            |
           |           |       |                            |
BOUndcond <            | [k]   |                             >                    &
           |                                                |
           |                                                |
           |              | -> XY  <  [x]  [y]  >       |   |
           |              |                             |   |
           |    SEGMent  <         | <  [i]  [j]  > |    >  |
                          |    IJ <                  >  |
                          |        | <  [k]  >      |   |

              BTYPe WLEV|VEL|DISCH|RIEMann|LRIEmann|WEAKrefl|SOMMerfeld|OUTFlow   &

              LAYer [k] | LOGarithmic                                             &

              SMOOthing [period] SEC|MIN|HR|DAY                                   &

              ADDBoundwave                                                        &

                    | -> FOURier     [azero] < [ampl] [omega] [phase] >
                    |    REGular     [h] [per] [dir]
                    |    BICHromatic [h1] [h2] [per1] [per2] [dir1] [dir2]
        | CONstant <     SPECTrum    [h] [per] [dir] [dd] [cycle] SEC|MIN|HR|DAY
        |           |    SERIes      ’fname’ [itmopt]
        |           |    SPECFile    ’fname’ [cycle] SEC|MIN|HR|DAY
       <                                                                          &
        |           | -> FOURier  < [len] [azero] < [ampl] [omega] [phase] > >
        |           |    REGular  < [len] [h] [per] [dir] >
        |           |    BICHrom  < [len] [h1] [h2] [per1] [per2] [dir1] [dir2] >
        | VARiable <     SPECTrum < [len] [h] [per] [dir] [dd] [cycle] S|MI|HR|DA >
                    |    SERIes   < [len] ’fname’ [itmopt] >
                    |    SPECFile < [len] ’fname’ [cycle] SEC|MIN|HR|DAY >
                    |    SPECSwan ’fname’ [cycle] SEC|MIN|HR|DAY
```

This command `BOUNDCOND` defines a boundary condition at the open boundary. It consists of two parts, the first part defines the boundary side or segment where the boundary condition will be imposed, the second part defines the type of the boundary condition and the parameters.

There are two ways to define the part of the boundary at which the boundary condition is imposed. The first way (SIDE) is easiest if the boundary is one full side of the computational grid, although it should not be used for curved boundaries. The second method (SEGMENT) can be used if the boundary segment goes around the corner of the grid, or if the segment is only part of one side of the domain.

This BOUNDCOND command can be given a number of times, i.e. to define boundary conditions on various sides or segments of the boundary. One BOUNDCOND command can be used for only one side or one contiguous segment.

When no BOUNDCOND command is specified at a boundary, this boundary is considered to be a closed one where the normal velocity at the boundary is set to zero.

Note that command BOUNDCOND can not be combined with the application of mass source function for internal wave generation (see command SOURCE).

The specifications of the sub keywords are shown below. Please note that reference is made to the examples provided at the end of this section.

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

NORTH, ... — indicates on which side the boundary condition is applied. N means the boundary is the north edge (if present) of the computational domain, likewise for W is west, S is south, E is east, NW is northwest, NE is northeast, SW is southwest and SE is southeast. Note that the boundary is assumed to be a straight line. (Use the SEGMENT command as an alternative.)
Another note: in case of Cartesian coordinates, the direction of the problem coordinate system must be defined by the user (see the SET ...[north] command), by default the positive $x-$axis points East.
ONLY MEANT FOR STRUCTURED GRIDS.

[k] — indicates on which side of the unstructured grid the boundary condition is applied. The value of [k] corresponds to the boundary marker as indicated in file(s) produced by a grid generator (such as in the last column of the Triangle .node file and the Easymesh .n file or the last part of file fort.14). Boundary markers are tags to identify which vertices occur on a boundary of the mesh. It is assumed that the full side in question is represented by a single boundary marker.
ONLY MEANT FOR UNSTRUCTURED MESHES.

CCW, CLOCKWISE — see description of [len] below; these option are only effective if the option VARIABLE is used (see below).

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

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

`[x], [y]` problem coordinates of a point of the boundary segment (see command `COORD`).

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

`[i], [j]` grid indices of a point of the segment. Values of `[i]` range from 1 to `[mxc]`+1 and values of `[j]` from 1 and `[myc]`+1 (`[mxc]` and `[myc]` as defined in the command `CGRID`).
ONLY MEANT FOR STRUCTURED GRIDS.

`[k]` index of boundary vertex of the segment. This can be obtained in a grid generator file (`.node` and `.n` files of Triangle and Easymesh, respectively). The order must be counterclockwise!
ONLY MEANT FOR UNSTRUCTURED MESHES.

`BTYPE` with this option the type of boundary condition is specified.

`WLEV` water level is imposed.
Required further specification by means of Fourier series or time series (see below).

`VEL` velocity normal to the boundary is imposed.
Required further specification by means of Fourier series or time series (see below).

`DISCH` discharge per unit width normal to the boundary is imposed.
Required further specification by means of Fourier series or time series (see below).

`RIEMANN` Riemann invariant is imposed. It is defined as $u \pm 2\sqrt{gh}$ with u the velocity normal to the boundary and h the water depth.
The sign depends on the location of the boundary. The plus sign refers to an inflow velocity at the western/left and southern/lower boundaries, and the minus sign refers to inflow velocity at the eastern/right and northern/upper boundary. This boundary condition is particularly meant for a supercritical flow or hydraulic jump in e.g. rivers or open channels.

Required further specification by means of Fourier series or time series (see below).

`LRIEMANN` linearized Riemann invariant is imposed. It is defined as $u \pm \zeta\sqrt{g/d}$ with u the velocity normal to the boundary, ζ the water level and d the bottom level. The sign depends on the location of the boundary. The plus sign refers to an inflow velocity at the western/left and southern/lower boundaries, and the minus sign refers to inflow velocity at the eastern/right and northern/upper boundary. Note that linearized Riemann invariants can only be applied if the water level (ζ) is small compared to the local bottom level (d). This is mainly applicable for a subcritical flow in relative deep waters. Examples are tidal flows in a continental shelf or in a harbour.

Required further specification by means of Fourier series or time series (see below).

`WEAKREFL` the boundary condition is weakly reflective.
Required further specification by means of Fourier series or time series of water level, or monochromatic, bichromatic or irregular waves by means of a spectrum (see below).

`SOMMERFELD` Sommerfeld radiation condition is imposed. This condition is especially useful for near-linear long waves at very shallow water.
No further specification is needed.

`OUTFLOW` water depth is aligned to bottom level (for supercritical river flow only to obtain a so-called S2 backwater curve.)
No further specification is needed.
ONLY MEANT FOR STRUCTURED GRIDS.

`LAYER` indicates a layer where the boundary condition is imposed.
Only applicable to layer-dependent quantities.

`[k]` layer index (1 ≤ `[k]` ≤ `[kmax]`).

`LOGARITHMIC` the vertical logarithmic profile for velocity at the boundary is assumed.
Only applicable to turbulent flows.

`SMOOTHING` with this option a ramp function is applied to the boundary condition in order to start up the simulation smoothly.

`[period]` the smoothing period of which the unit is indicated in the next option:

SEC	unit seconds
MIN	unit minutes
HR	unit hours
DAY	unit days

`ADDBOUNDWAV` with this option second order bound waves are added to the first order Stokes waves as specified by regular or irregular waves (see below).
Note: these also include sub- and super-harmonics in case of bichromatic waves or irregular waves using a spectrum.

`CONSTANT` with this option the boundary condition is constant along the side or segment.

`FOURIER` the Fourier series is defined by means of the following parameters:

`[azero]` the amplitude for zero frequency is given (unit depending on boundary type).
Note: `[azero]` can also be used to specify a constant-in-time boundary value.

`[ampl]` the amplitudes for a number of components are given (unit depending on boundary type).

`[omega]` the angular frequencies for a number of components are given (in rad s^{-1}).

[phase] the phase for a number of components are given (in degrees).

REGULAR monochromatic (long-crested) wave is defined with the following parameters:

[h] the wave height (in m).

[per] the wave period (in s).

[dir] the wave direction (in degrees; Cartesian or Nautical convention, see command SET).
Default: no specification of [dir] means incident direction is perpendicular to the boundary (only in case of rectilinear grid).

BICHROMATIC bichromatic (long-crested) wave is defined with the following parameters:

[h1] the first wave height (in m).

[h2] the second wave height (in m).

[per1] the first wave period (in s).

[per2] the second wave period (in s).

[dir1] the first wave direction (in degrees; Cartesian or Nautical convention, see command SET).

[dir2] the second wave direction (in degrees; Cartesian or Nautical convention, see command SET).
Default: no specification of [dir1] or [dir2] means incident direction is perpendicular to the boundary (only in case of rectilinear grid).

SPECTRUM the wave spectrum is defined by means of the following spectral parameters (see command BOUND SHAPE for spectral shape):

[h] the characteristic wave height (in m).
[h] is the value of the significant wave height, if option SIG was chosen in command BOUND SHAPE or
[h] is the value of the RMS wave height, if option RMS was chosen in command BOUND SHAPE.

[per] the characteristic wave period (in s).
[per] is the value of the peak period, if option PEAK was chosen in command BOUND SHAPE or
[per] is the value of the mean period T_{m01}, if option MEAN was chosen in command BOUND SHAPE.

[dir] the peak wave direction (in degrees; Cartesian or Nautical convention, see command SET).
Default: no specification of [dir] means incident direction is perpendicular to the boundary (only in case of rectilinear grid).

[dd] coefficient of directional spreading; a $\cos^m(\theta)$ distribution is assumed.
[dd] is interpreted as the directional standard deviation in degrees, if the option DEGREES is chosen in the command BOUND SHAPE.
[dd] is interpreted as the power m, if the option POWER is chosen in the command BOUND SHAPE.
Default: [dd] = 0., i.e. no directional spreading.

[cycle] the cyclic period of the time series of surface elevation to be synthesized. This may correspond to the time period over which surface elevation is

outputted after steady-state condition has been established. The corresponding unit is indicated in the next option:

SEC	unit seconds
MIN	unit minutes
HR	unit hours
DAY	unit days

`SERIES` the time series is given in a file.
This file is for only one location; it has a number of lines which each contain 2 numbers, i.e.: time, quantity. The notation of time is indicated by `[itmopt]` (see below). However, if time is represented as a real number in seconds, then **do not specify** `[itmopt]`. The quantity depends on the type of boundary condition (see command `BTYPE`).
Example of such a file containing surface elevation in meters (`[itmopt]` = 7):

```
000000.000    0.0000000E+00
000001.000    4.2307975E-08
000002.000    1.6923190E-07
000003.000    3.8077177E-07
000004.000    6.7692758E-07
000005.000    1.0576993E-06
000006.000    1.5230870E-06
000007.000    2.0730906E-06
```

`'fname'` name of the file containing the time series.

`[itmopt]` time coding option. The following format can be employed:

1 :	ISO-notation	19870530.153000
2 :	(as in HP compiler)	'30−May−87 15:30:00'
3 :	(as in Lahey compiler)	05/30/87.15:30:00
4 :		15:30:00
5 :		87/05/30 15:30:00'
6 :	as in WAM	8705301530
7 :		153000.000

Default: no specification of `[itmopt]` means time is represented as a real number in seconds.

`SPECFILE` the wave spectrum is given in a file. There are two types of files:

- files containing 1D or non-directional wave spectrum (usually from measurements), and
- files containing 2D or directional wave spectrum (possibly from a SWAN run).

A file containing the 1D wave spectrum is for only one location; it has the string `SPEC1D` on the first line of the file and a number of lines which each contain 2 numbers, i.e.: frequency in Hz, variance density in m^2/Hz.
Example of such a file:

```
SPEC1D
3.8602E-03     5.2168E-01
5.7903E-03     1.0230E+00
7.7204E-03     1.4567E+00
9.6505E-03     1.7232E+00
1.1581E-02     1.8832E+00
1.3511E-02     1.7570E+00
1.5441E-02     1.3429E+00
1.7371E-02     9.8666E-01
```

The structure of the files containing directional spectrum is described in Appendix D of the SWAN User Manual. These files may only contain stationary wave spectra, and can be used for one location only.

`'fname'` name of the file containing the wave spectrum.

`[cycle]` the cyclic period of the time series of surface elevation to be synthesized. This may correspond to the time period over which surface elevation is outputted after steady-state condition has been established. The corresponding unit is indicated in the next option:

SEC unit seconds
MIN unit minutes
HR unit hours
DAY unit days

`VARIABLE` with this option the boundary condition can vary along the side or segment. The boundary condition is prescribed at a number of points of the side or segment, these points are characterized by their distance from the begin point of the side or segment.

`[len]` is the distance from the first point of the side or segment to the point along the side or segment for which the incident wave spectrum is prescribed. Note: these points do no have to coincide with grid points of the computational grid. `[len]` is the distance in m or degrees in the case of spherical coordinates, **not** in grid steps. The values of `[len]` should be given in ascending order. The length along a `SIDE` is measured in clockwise or counterclockwise direction, depending on the options `CCW` or `CLOCKWISE` (see above). The option `CCW` is default. In case of a `SEGMENT` the length is measured from the indicated begin point of the segment.

`SPECSWAN` the wave spectra are given in a file and are obtained from a SWAN run. The structure of this file is described in Appendix D of the SWAN User Manual. This file must contain more than 1 stationary wave spectrum, either 1D or 2D. The coordinates of the locations will be taken into account and SWASH will interpolate the spectra to the given boundary side or segment. The coordinate system of the SWAN run must be the same as the one used by the current SWASH run. See also Section 2.6.

CANNOT BE USED IN 1D-MODE.

`'fname'` name of the file containing the wave spectra.

`[cycle]` the cyclic period of the time series of surface elevation to be synthesized. This may correspond to the time period over which surface elevation is outputted after steady-state condition has been established. The corresponding unit is indicated in the next option:

SEC	unit seconds
MIN	unit minutes
HR	unit hours
DAY	unit days

Further explanation and examples are given below. See also Section 5.3 for more details.

For each type of boundary condition (e.g., `WLEV`, `VEL`) a forcing type such as a Fourier series or a time series must be prescribed. Exceptions to this are the commands `SOMMERFELD` and `OUTFLOW`.

Instead of time-dependent forcing type (e.g., `FOURIER`, `SERIES`), a constant (in time) boundary value can be imposed using command `BOU ...  CON [azero]`. For instance, to impose a constant discharge of 2500 m^3/s, give the following command

```
  BOU ... BTYPE DISCHARGE CON 2500.
```

Note that command `FOURIER` is the default.

In case of velocity or discharge boundary condition the user may specify the vertical distribution depending on the application. For turbulent flows, the user may specify the logarithmic velocity distribution over the vertical at the open boundary. An alternative would be to specify the velocity/discharge for each separate layer.

A uniform distribution over the vertical is assumed when none of the keywords `LAYER [k]` and `LOGARITHMIC` is specified.

Note that by specifying regular or irregular waves at the open boundary using one of the following commands `REGULAR`, `BICHROMATIC`, `SPECTRUM` or `SPECFILE`, the vertical profile of the horizontal velocity derived from the first order Stokes wave theory is assumed. Hence, no type of boundary condition (`WLEV`, `VEL`, etc.) nor vertical velocity profile should be specified. The exception is specifying `BTYPE WEAKREFL` when a weakly reflective boundary is required. Also note that the user can additionally specify the second order Stokes correction, if so desired (see below).

Note that the forcing types `FOURIER` and `SERIES` are a generic one, which means they can be used both for long waves (tidal waves, seiches, etc.) and short waves (monochromatic and bichromatic waves or multiple Fourier modes). Therefore, in these cases, the user must explicitly specify the type of the boundary condition (`BTYPE ...` including quantity) and the vertical distribution of the velocity/discharge (`LAYER [k]` or `LOGARITHMIC` or none of them).

At the wavemaker boundary, SWASH can accurately generate linear and nonlinear waves at any depth using the derived solutions of the vertically discretized model equations based on the first order and second order Stokes pertubation expansions. See Vasarmidis et al. (2024) for a detailed explanation. This is, however, limited to up to four layers (`[kmax]` ≤ 4) and they must be distributed equidistantly (see command `VERT`). In addition, the Keller-box scheme for the non-hydrostatic pressure must be applied (see command `NONHYD BOX`). If one of these three conditions is not fulfilled, then SWASH imposes the vertical hyperbolic cosine velocity profile according to the linear wave theory (see also Chapter 5).

First order monochromatic waves can be imposed using the command

```
BOU ... BTYPE WEAK ... CON REG ...
```

In this case the first order Stokes solutions of the surface elevation and the layer-averaged velocities are imposed at the incident boundary. This also avoids the problem of the well-known drop in the wave height in the first few grid points next to the boundary, especially for high values of dimensionless depth kd. Note that weakly reflection at the open boundary (keyword `WEAK`) is usually recommended in order to prevent re-reflections at the open boundary.

Second order bound waves can be added by the following command

```
BOU ... BTYPE WEAK ... ADDBOUNDWAVE CON REG ...
```

By setting `ADDBOUNDWAVE`, self-interacting second order super-harmonics (doubled frequencies) will be added to the first order Stokes waves. These super-harmonics are bound (phase-locked) to the primary waves. Adding of these second order bound waves is required to suppress the generation of spurious free modes, especially for high (nonlinear) waves. Since the added bound waves are solutions to the discrete model equations, they exactly cancel the corresponding spurious free waves at the wavemaker, thus leading to a spatially homogeneous wave field inside the domain.

Bichromatic waves with periods $2\pi/\omega_\mathrm{m}$ and $2\pi/\omega_\mathrm{n}$ can be imposed at the open boundary similar to monochromatic waves. For instance, second order bichromatic waves can be prescribed where the interaction of two given components forces a second order bound wave group consisting of four components, namely, one sub-harmonic, $\omega_\mathrm{m} - \omega_\mathrm{n}$, and three super-harmonics, $2\omega_\mathrm{m}$, $2\omega_\mathrm{n}$ and $\omega_\mathrm{m} + \omega_\mathrm{n}$. Specification of second order bichromatic waves is as follows

```
BOU ... BTYPE WEAK ... ADDB CON BICHromatic ...
```

With this command, six components will be imposed, two for the primary waves and four for the bound waves. For the specification of only two primary wave components without adding any other modes, remove ADDB or, alternatively, use the forcing type FOURIER instead.

Irregular waves can also be generated at the open boundary using a weakly-reflective wavemaker based on one-dimensional or two-dimensional spectrum with a parametric shape, including second order bound waves (self-interacting super-harmonics and cross-interacting sub- and super-harmonics), as follows

```
  BOU ... BTYPE WEAK ... ADDBoundwave CON SPECTrum ...
```

Instead of SPECTRUM the user can also define a 1D/2D non-parametric wave spectrum using SPECFile or SPECSWAN.

Note that for the cases using unstructured meshes, only incident wave direction perpendicular to the open boundary is allowed.

```
           |    X  |
           |       |
SOURce    <  -> Y   >  [centre] [width] [depth] [delta]                   &
           |       |
           |   [k] |

           | REGular  [h] [per] [dir]
          <                                                               &
           | SPECTrum [h] [per] [dir] [dd] [cycle] SEC|MIN|HR|DAY

          SMOOthing [period] SEC|MIN|HR|DAY
```

This command SOURCE activates the generation of waves within the computational domain using the spatially distributed mass source function. This command can not be combined with the specification of boundary conditions at the boundaries (see BOUNDCOND).

The user specifies a so-called source area, i.e. a rectangular area within the computational domain, and subsequently a wave signal that will be generated in the source area. The computational domain itself is assumed to be rectangular as well, however, the grid can either be uniform or non-uniform. The user is advised to combine this internal wave generation with sponge layers to absorb waves at boundaries effectively (see command SPONGE).

The source area is a rectangle and is determined by means of its centroid (or the centre of gravity) and its width. This rectangle is parallel either to the $xc-$axis or to the $yc-$axis of the (rotated) rectilinear computational domain (see Figure 4.1), while its length extends along the whole domain. In the first case the centroid is with respect to the $xc-$axis such that the connecting line centroid$-$axis is parallel to the $yc-$axis, and the width is parallel to the $yc-$axis as well. In the second case it is the other way around. In case of unstructured mesh, the centroid is with respect to one side of the domain as specified by the boundary marker.

The waves to be generated are either regular or irregular as defined by means of a wave spectrum. First order Stokes (linear) wave theory is assumed. Note that a still water depth is required (for the calculation of wave energy velocity).

Note that only incident wave direction perpendicular to the boundary is allowed in case of unstructured grids.

`X` source area is parallel to $xc-$axis of the rectilinear grid.

`Y` source area is parallel to $yc-$axis of the rectilinear grid.
This is default.

`[k]` indicates from which side of the unstructured grid `[centre]` is measured.
The value of `[k]` corresponds to the boundary marker as indicated in file(s) produced by a grid generator (such as in the last column of the Triangle `.node` file and the Easymesh `.n` file or the last part of file `fort.14`). Boundary markers are tags to identify which vertices occur on a boundary of the mesh. It is assumed that the full side in question is represented by a single boundary marker.
ONLY MEANT FOR UNSTRUCTURED MESHES.

`[centre]` the centre of gravity of source area in meters.

`[width]` the width of source area in meters.

`[depth]` the still water depth at source area in meters.

`[delta]` shape parameter for the source function.
Default: `[delta]` = 0.5

`REGULAR` regular waves will be generated. See command `BOUNDCOND` for the specification of their parameters.

`SPECTRUM` irregular waves will be generated. See command `BOUNDCOND` for the specification of their parameters. See also command `BOUND SHAPE` for spectral shape.

`SMOOTHING` with this option a ramp function is applied to start up the simulation smoothly.

`[period]` the smoothing period of which the unit is indicated in the next option:

SEC	unit seconds
MIN	unit minutes
HR	unit hours
DAY	unit days

```
              | North |
              | NW    |
              | West  |
              | SW    |
SPONgelayer  <  South  >  [width]  | < [k]  [width] >
              | SE    |
              | East  |
              | NE    |
```

This command can be used to specify the sponge layers around the edges of the computational domain.

Sponge layers are very effective in absorbing wave energy at open boundaries where waves are supposed to leave the computational domain freely. So, they prevent reflections at open boundaries. A sponge layer may have a width of 1 to 3 typical wave lengths.

Note that by including a sponge layer of `[width]` meters, the computational domain needs to be extended with `[width]` meters as well (see command `CGRID`).

To specify the sponge layers, a distinction is made between structured and unstructured meshes. The keywords `NORTH, ...` are meant for the structured grid only. Also note that the edge to which a sponge layer is placed is assumed to be a straight line.

The variable `[k]` is to be used for the unstructured mesh (see below) and can be repeated as many sponge layers to be chosen.

In case of Cartesian coordinates, the direction of the problem coordinate system must be defined by the user (see the `SET ...[north]` command), by default the positive $x-$axis points East.

`NORTH`	sponge layer is placed at the north edge of the domain.
`NW`	sponge layer is placed at the northwest edge of the (rotated) domain.
`WEST`	sponge layer is placed at the west edge of the domain.
`SW`	sponge layer is placed at the southwest edge of the (rotated) domain.
`SOUTH`	sponge layer is placed at the south edge of the domain.
`SE`	sponge layer is placed at the southeast edge of the (rotated) domain.
`EAST`	sponge layer is placed at the east edge of the domain.
`NE`	sponge layer is placed at the northeast edge of the (rotated) domain. ONLY MEANT FOR STRUCTURED GRIDS.
`[k]`	indicates on which side of the unstructured grid the sponge layer is applied.

The value of `[k]` corresponds to the boundary marker as indicated in file(s) produced by a grid generator (such as in the last column of the Triangle `.node` file and the Easymesh `.n` file or the last part of file `fort.14`). Boundary markers are tags to identify which vertices occur on a boundary of the mesh. It is assumed that the full side in question is represented by a single boundary marker.
ONLY MEANT FOR UNSTRUCTURED MESHES.

`[width]` the width of sponge layer in meters.

```
FLOAT [alpha] [theta]
```

CANNOT BE USED IN CASE OF UNSTRUCTURED GRIDS.

With this optional command the user can specify some parameters in case of (*fixed*) floating objects or pressurized flow. See commands `INPGRID DRAFT` and `READINP DRAFT` in order to define floating objects. If these commands are not used, SWASH will not account for effects of floating structures on the (pressurized) flow.

`[alpha]` compressibility factor for flow beneath *fixed* floating object.
This option is meant for academic purposes only.
Note: $0 \leq$ `[alpha]` ≤ 1.
Default: `[alpha]` $= 0$.

`[theta]` parameter for the $\theta-$method applied beneath floating object.
`[theta] = 0.5` corresponds to the Crank-Nicolson scheme and
`[theta] = 1` to the implicit Euler scheme.
Default: `[theta]` $= 1$.

```
BODY DIMension [l] [mass] [Ix] [Iy] [Iz] [cogx] [cogy] [cogz]               &

     DOF SUrge SWay HEave ROll PItch YAw                                    &

     ( MLIne < [K] [B] [apbx] [apby] [apbz] [apfx] [apfy] [apfz] [elen] >   &

             PRETension )                                                   &

     ( FENder < [K] [apfx] [apfy] [apfz] > )
```

CANNOT BE USED IN CASE OF UNSTRUCTURED GRIDS.

With this optional command the user can specify some dimensional parameters in case of *moving* rigid bodies. See commands `INPGRID DRAFT` and `READINP DRAFT` in order to define rigid bodies and also the commands `INPGRID LABEL` and `READINP LABEL` to label them (only necessary if more than one body is present). If these commands are not used, SWASH will not account for fluid-structure interactions.

`[l]` the label number for each rigid body.
This number must correspond to that of the label as specified by
the commands `INPGRID LABEL` and `READINP LABEL`.

`[mass]` body mass (in kg).

`[Ix]` moment of inertia of body around the $x-$axis (in kg m^2).

`[Iy]` moment of inertia of body around the $y-$axis (in kg m^2).

`[Iz]` moment of inertia of body around the $z-$axis (in kg m^2).

`[cogx]` $x-$coordinate of center of gravity (COG) of rigid body (in m)
with respect to the computational grid.

`[cogy]` $y-$coordinate of center of gravity (COG) of rigid body (in m).
with respect to the computational grid.

`[cogz]` $z-$coordinate of center of gravity (COG) of rigid body (in m).
with respect to the computational grid.

`DOF` this option indicates which of the six degrees of freedom of movement
of rigid bodies are to be included.

`SURGE` translation in the $x-$axis.

`SWAY` translation in the $y-$axis.

`HEAVE` translation in the $z-$axis.

`ROLL` rotation around the $x-$axis.

`PITCH` rotation around the $y-$axis.

`YAW` rotation around the $z-$axis.

`MLINE` with this option the user can determine coefficients for mooring lines.

`[K]` spring coefficient (in N/m).

`[B]` damping coefficient (in Ns/m).

`[apbx]` $x-$component of attachment point at bed relative to COG (in m).

`[apby]` $y-$component of attachment point at bed relative to COG (in m).

`[apbz]` $z-$component of attachment point at bed relative to COG (in m).

`[apfx]` $x-$component of attachment point at body relative to COG (in m).

`[apfy]` $y-$component of attachment point at body relative to COG (in m).

`[apfz]` $z-$component of attachment point at body relative to COG (in m).

`[elen]` equilibrium length (in m).

`PRETENSION` this option indicates whether pretension of mooring lines is to be included.

`FENDER` with this option the user can determine coefficients for fenders.

`[K]` spring coefficient (in N/m).

[apfx]	$x-$component of attachment point at body relative to COG (in m).
[apfy]	$y-$component of attachment point at body relative to COG (in m).
[apfz]	$z-$component of attachment point at body relative to COG (in m).

```
             |    NEWmark [beta] [gamma] |
             |                           |
             | -> CH [rho]               |
BODY SOLVer <                             > COUPling [tol] [maxiter] [relax] &
             |    HHT [rho]              |
             |                           |
             |    WBZ [rho]              |

     KBC [theta]
```

CANNOT BE USED IN CASE OF UNSTRUCTURED GRIDS.

With this optional command the user can specify some numerical parameters for the rigid body solver. The commands `BODY DIMENSION` and `BODY SOLVER` belong together.

A number of classical one-step implicit ODE solvers for the dynamic response systems are available such as the Newmark's method (1959) and the generalized$-\alpha$ method (Chung and Hulbert, 1993). The method of Newmark is steered by two parameters $0 \leq \beta \leq 1/2$ and $0 \leq \gamma \leq 1$ that control both accuracy and stability. The method is second order accurate if $\gamma = \frac{1}{2}$ and is unconditionally stable when $2\beta \geq \gamma \geq \frac{1}{2}$. However, $\gamma > \frac{1}{2}$ results in numerical damping. Therefore, the common choice is $\beta = \frac{1}{4}$ and $\gamma = \frac{1}{2}$, which is also known as the average acceleration method.

The generalized$-\alpha$ method typically allows for numerical dissipation of high frequency noise while minimizing unwanted low frequency dissipation. Examples are the Chung-Hulbert (CH) scheme (1993), the Hilber-Hughes-Taylor (HHT) scheme (1977) and the Wood-Bossak-Zienkiewicz (WBZ) scheme (1980), and are second order accurate and unconditionally stable.

The parameter that counteracts the amplification of artificial high frequency modes is the spectral radius of the amplification matrix, $0 \leq \rho_\infty \leq 1$. The highest amount of damping that annihilates the highest frequency of interest in one time step is obtained by $\rho_\infty = 0$, while a value of one preserves the highest frequency mode. The CH scheme is known to be optimal in terms of the amount of numerical damping.

The default time-marching method is the Chung-Hulbert scheme with $\rho_\infty = 1$.

In order to predict the wave-induced response of a moored floating vessel (or a multiple WECs) an iterative process is employed that tightly coupled the equations of rigid body motion to the shallow water equations (including the non-hydrostatic pressure). Here, the motion of a rigid body is a direct consequence of the wave-induced forces acting on it and, in turn, the fluid surrounding the floating structure is influenced by the body movement (provided by the kinematic boundary conditions at the fluid-structure interface). The approach is to solve each subsystem independently (fluid flow and body motion, respectively), and subsequently implicitly coupled in an iterative Gauss-Seidel type manner. Convergence is reached when changes to the body motions within the iterative procedure are smaller than a user-defined tolerance. By default this is 10^{-4}, whereas the maximum number of iterations is set to 50.

It is well known that fluid-structure coupling algorithms tend to become unstable in case of light bodies that are subjected to high acceleration. A common approach to improve the convergence rate of such iteration schemes is to apply under-relaxation. The optimal value of the under-relaxation coefficient, as indicated here by `[relax]`, is problem-dependent, in particular, in the context of nonlinear problems. However, here we proposed a fixed low value of 0.3, which seems to be reasonable for moored floating structures.

`NEWMARK` this option indicates that the Newmark scheme will be adopted.
`[beta]` the first Newmark time integration parameter.
Note: $0 \leq$ `[beta]` ≤ 0.5.
Default: `[beta]` $= 0.25$
`[gamma]` the second Newmark time integration parameter.
Note: $0 \leq$ `[gamma]` ≤ 1.
Default: `[gamma]` $= 0.5$
`CH` this option indicates that the Chung-Hulbert scheme will be adopted.
This is the default.
`HHT` this option indicates that the Hilber-Hughes-Taylor scheme will be adopted.
`WBZ` this option indicates that the Wood-Bossak-Zienkiewicz scheme scheme will be adopted.
`[rho]` parameter that controls the spectral distribution of numerical dissipation.
Note: $0 \leq$ `[rho]` ≤ 1.
Default: `[rho]` $= 1.0$.
`COUPLING` with this option the user can determine parameters for the fluid-structure coupling.
`[tol]` error tolerance to terminate the iteration process.
Default: `[tol]` $= 0.0001$.
`[maxiter]` maximum number of iterations.
Default: `[maxiter]` $= 50$.
`[relax]` the under-relaxation coefficient.

Note: $0 \leq$ `[relax]` ≤ 1.
Default: `[relax]` $= 0.3$.

`KBC` with this option the user can determine the parameter below for the kinematic boundary conditions at fluid-structure interface.

`[theta]` the weighting parameter to determine the body velocities (both translational and rotational) at an intermediate time between previous and current time step.
Note: $0 \leq$ `[theta]` ≤ 1, with `[theta]` $= 0$ corresponding to the previous value and `[theta]` $= 1$ to the current value.
Default: `[theta]` $= 0.5$

4.5.4 Physics

```
                  | -> CONstant [cd]                        |
                  |                                         |
                  |    CHARNock [beta] [height]             |
                  |                                         |
                  |    LINear [a1] [a2] [b] [wlow] [whigh]  |
                  |                                         |   | REL  [alpha]
WIND [vel] [dir] <     WU                                    > <
                  |                                         |   | RELW [crest]
                  |    GARRatt                              |
                  |                                         |
                  |    SMIthbanke                           |
                  |                                         |
                  |    FIT                                  |
```

With this optional command the user can specify wind speed, direction and wind drag. Wind speed and direction are assumed constant. If this command is not used, SWASH will not account for wind effects.

This command is usually meant for large-scale wind driven circulation, tides and storm surges. Inclusion of wind effects may also be beneficial to buoyancy driven flows in coastal seas, estuaries and lakes. However, this option may also be useful for applications concerning wind effects on wave transformation in coastal waters, ports and harbours.

In SWASH seven different wind drag formulation are available, i.e., constant, linear on wind speed, Charnock, Wu, Garratt, Smith and Banke and the second order polynomial fit. The Charnock drag formulation is based on an implicit relationship between the wind and the roughness, while the other formulations, those of Wu, Garratt and Smith and Banke, express a linear relationship between the drag and the wind speed.

Recent observations indicate that these linear parameterizations overestimate the drag coefficient at high wind speeds ($U_{10} > 20$ m/s, say). Based on many authoritative studies it appears that the drag coefficient increases almost linearly with wind speed up to approximately 20 m/s, then levels off and decreases again at about 35 m/s to rather low values at 60 m/s wind speed. We fitted a 2nd order polynomial to the data obtained from these studies, and this fit is given by

$$C_d = (0.55 + 2.97\tilde{U} - 1.49\tilde{U}^2) \times 10^{-3}$$

where $\tilde{U} = U_{10}/U_{\mathrm{ref}}$, and the reference wind speed $U_{\mathrm{ref}} = 31.5$ m/s is the speed at which the drag attains its maximum value in this expression. These drag values are lower than in the expression of Wu (1982) by $10\% - 30\%$ for high wind speeds ($15 \leq U_{10} \leq 30$ m/s) and over 30% for hurricane wind speeds ($U_{10} > 30$ m/s).

Usually, the wind stress depends on the drag and the wind speed at a height of 10 m, U_{10}. However, it might be obvious that the influence of wind stress will reduce if the water is flowing in the same direction and it will decrease when the water flow and wind are in opposite directions. This may lead to a smaller wind setup on very shallow areas. Hence, the wind stress may be dependent on the wind velocity relative to the water, i.e. $U_{10} - u$, instead of the wind velocity as such. Here, u is either the depth-averaged flow velocity in the depth-averaged mode or the surface flow velocity in the multi-layered mode. Experiments have shown that the eigenfrequencies damp out much faster when this alternative is employed.

The considered wind is at 10 m above the surface. However, it might be better to consider the wind at the surface in order to relate this wind to the flow velocity. A factor α ($0 < \alpha \leq 1$) is introduced that take into account the difference between the wind velocity at 10 m height and the wind velocity at the surface, $U_s = \alpha\, U_{10}$. With the use of α in this formulation the influence of the flow velocity becomes even stronger. However, the exact value of α is yet unknown; further research on this parameter is needed. Therefore, this parameter is optionally and should be used with care.

Wave growth due to wind in shallow areas is included in the model. It is based on a parameterization of the momentum flux transferred from wind to surface waves similar to the well-known sheltering mechanism of Jeffreys (1925) as described in Chen et al. (J. Waterwy, Port, Coastal, Ocean Engng., 130, 312-321, 2004). The wind stress is expressed by

$$\tau_w = \rho_{\mathrm{air}}\, C_d \,|U_{10} - c|\, (U_{10} - c)$$

where ρ_{air} is the air density and c is the wave celerity. Hence, the wind velocity is taken relative to the wave celerity. The wind stress may vary over a wave length with a larger wind drag on the wave crest than that in the trough (Chen et al., 2004). This effect is implemented in the model by applying the wind stress on the wave crest only.

The default option is a constant wind drag coefficient, while the wind stress is related to the wind velocity at 10 m height only.

`[vel]` wind velocity at 10 m height (in m/s).

`[dir]` wind direction at 10 m height (in degrees; Cartesian or Nautical convention, see command `SET`).

`CONSTANT` this option indicates that a constant drag coefficient will be adopted.

`[cd]` dimensionless coefficient.
Default: `[cd]` = 0.002

`CHARNOCK` indicates that the Charnock drag formulation will be adopted.

`[beta]` dimensionless Charnock coefficient.
Default: `[beta]` = 0.032

`[height]` height (in m) above the free surface where the wind speed has been measured.
Default: `[height]` = 10.

`LINEAR` indicates that the wind drag depends linearly on wind speed.
For this, both lower and upper bounds of the wind speed, `[wlow]`, `[whigh]`, and two coefficients, `[a1]`, `[b]`, need to be specified as follows:
$c_\mathrm{d} = 0.001\,($ `[a1]` $+$ `[b]` $U_{10})$, with c_d the drag coefficient and U_{10} the wind speed in between `[wlow]` and `[whigh]`.

`[a1]` coefficient in the above linear function.

`[a2]` not used.

`[b]` coefficient in the above linear function.

`[wlow]` lower bound of wind speed.

`[whigh]` upper bound of wind speed.

`WU` indicates that the drag formulation of Wu will be adopted.

`GARRATT` indicates that the drag formulation of Garratt will be adopted.

`SMITHBANKE` indicates that the drag formulation of Smith and Banke will be adopted.

`FIT` drag coefficient is based on the 2nd order polynomial fit.

`RELATIVE` indicates that the wind stress depends on the wind velocity relative to the water.

`[alpha]` parameter to relate the wind velocity at 10 m height to wind velocity at surface.
Note: $0 <$ `[alpha]` ≤ 1.
Default: `[alpha]` = 1.

`RELWAVE` indicates that the wind stress depends on the wind velocity relative to the wave celerity. This option enables to include wind effects on the nearshore wave transformation.

`[crest]` free parameter representing a ratio of the forced crest height to maximum surface elevation. Use of this parameter implies that the wind stress is only applied on wave crests of which the surface elevation is larger than the given fraction `[crest]` of the maximum elevation with respect to the datum level.
Note: $0 \leq$ `[crest]` ≤ 1.

Default: [crest] = 0.4

The quantities [vel] and [dir] are required if this command is used except when the command READINP WIND is specified.

```
           |    CONstant [cf]
           |
           |    CHEZy [cf]
           |
           | -> MANNing [cf]
FRICtion  <
           |    COLEbrook [h]
           |
           |             | -> SMOOTH
           |    LOGlaw <
                         |    ROUGHness [h]
```

With this optional command the user can activate bottom friction. If this command is not used, SWASH will not account for the bottom friction.

For typically depth-averaged calculations, four different bottom friction values are available, i.e., constant, Chezy, Manning and Colebrook-White values. Note that the Colebrook-White friction value equals the Nikuradse roughness height. Although they are associated with depth-averaged flow velocities, they may be applied in the multi-layered mode as well. However, some inaccuracies may occur in the vertical structure of the velocity, in particular when the depth-averaged velocity is zero. Alternatively, the logarithmic wall law may be applied. In this case, a distinction is made between smooth and rough beds. For rough beds, the user must apply a Nikuradse roughness height.

The aforementioned friction formulations are usually derived for quasi-steady flow condition (e.g. flow in a river). However, numerical experiments have indicated that the Manning formula provides a good representation of wave dynamics in the surf zone, and even better to that returned by other friction formulations.

The default option is: MANNING with a constant friction coefficient.

CONSTANT this option indicates that a dimensionless friction coefficient will be adopted.
[cf] dimensionless coefficient.
Default: [cf] = 0.002

Note that [cf] is allowed to vary over the computational region; in that case use the commands INPGRID FRICTION and READINP FRICTION to define and read the friction data. The command FRICTION is still required to define the type of friction expression. The value of [cf] in this command is then not required (it will be ignored).

CHEZY indicates that the Chezy formula will be activated.

[cf] Chezy coefficient (in $m^{1/2}/s$).
Default: [cf] = 65.
Note that [cf] is allowed to vary over the computational region; in that case use the commands INPGRID FRICTION and READINP FRICTION to define and read the friction data. The command FRICTION is still required to define the type of friction expression. The value of [cf] in this command is then not required (it will be ignored).

MANNING indicates that the Manning formula will be activated.

[cf] Manning coefficient (in $m^{-1/3}$ s).
Default: [cf] = 0.019
Note that [cf] is allowed to vary over the computational region; in that case use the commands INPGRID FRICTION and READINP FRICTION to define and read the friction data. The command FRICTION is still required to define the type of friction expression. The value of [cf] in this command is then not required (it will be ignored).

COLEBROOK indicates that the Colebrook-White formula will be activated.

[h] Nikuradse roughness height (in m).
Note that [h] is allowed to vary over the computational region; in that case use the commands INPGRID FRICTION and READINP FRICTION to define and read the roughness heights. The command FRICTION is still required to define the type of friction expression. The value of [h] in this command is then not required (it will be ignored).

LOGLAW indicates that the logarithmic wall law will be activated.

SMOOTH indicates that the bottom is smooth, i.e. the roughness height is zero.
This option can be used in the depth-averaged mode (a logarithmic velocity profile is then assumed). Note that this option must be combined with the standard $k - \varepsilon$ model in the multi-layered mode (see command VISC).
This option is default.

ROUGHNESS indicates that the bottom is rough and is determined by the roughness height.
This option can be used in the depth-averaged mode (a logarithmic velocity profile is then assumed). Note that this option must be combined with the standard $k - \varepsilon$ model in the multi-layered mode (see command VISC).

[h] Nikuradse roughness height (in m).
Note that [h] is allowed to vary over the computational region; in that case use the commands INPGRID FRICTION and READINP FRICTION to define and read the roughness heights. The command FRICTION is still required to

define the type of friction expression. The value of [h] in this command is then not required (it will be ignored).

```
                          | -> CONstant [visc]
                          |
           | -> Horizontal <     SMAGorinsky [cs]
           |              |
           |              |    MIXing [lm]
           |
           |
VISCosity <     Vertical  KEPS [cfk] [cfe]
           |
           |
           |              | -> LINear
           |    FULL  KEPS <
           |              |    NONLinear
```

With this optional command the user can activate turbulent mixing. If this command is not used, SWASH will not account for turbulent mixing.

The turbulence structure of the flow in shallow water is characterized by the coexistence of three-dimensional turbulence, having length scales less than the water depth, and horizontal two-dimensional eddies with much larger length scales. Such a non-isotropic character of shallow water turbulence may produce a large difference between horizontal and vertical eddy viscosity coefficients.

Three-dimensional turbulence is mainly generated by vertical velocity gradients (including bottom and wind stresses) whose length scale is smaller than the water depth. Therefore, a turbulence model is necessary to properly evaluate the vertical momentum exchange in flows. This is represented by the vertical eddy viscosity coefficient in the momentum equations.

In estuaries and coastal seas the large-scale flow patterns are usually determined by tidal forcing, the baroclinic pressure gradient, wind and bottom friction, whereas the horizontal turbulent stresses have little impact. Hence, for far field calculations the use of an horizontal eddy viscosity model has a negligible role. On the other hand, for near field flows, horizontal large-scale eddies generated by lateral shear may have a significant role in horizontal mixing. Examples are mixing layers developing at harbour entrances, along groyne fields and floodplains, separation flows, wakes and jets, and flows near discharges. The horizontal exchange of turbulent momentum need to be modelled properly, since a constant value for the horizontal eddy viscosity is too crude as an assumption for such near field predictions.

In SWASH both the horizontal and vertical eddy viscosities can be specified, either separately or combined. In such cases, the pressure is assumed to be hydrostatic. In addition, the Boussinesq hypothesis is applied that basically assumes the isotropy of turbulence.

Three different horizontal eddy viscosity models are available, i.e., a constant viscosity, the Smagorinsky model and the Prandtl mixing length hypothesis. Vertical mixing can be modelled by using the standard $k-\varepsilon$ model, with k the turbulent kinetic energy per unit mass and ε the dissipation rate of turbulent kinetic energy per unit mass (Launder and Spalding, 1974).

Within the vegetation canopy, it is assumed that all energy of the mean flow is converted to turbulent energy due to the plant drag (see command `VEGETATION`). This process is modelled by means of the vegetation-induced turbulence production terms in the $k-\varepsilon$ model. They are accompanied with two empirical constants `[cfk]` and `[cfe]` associated with k and ε, respectively. We have selected the values as suggested by Shimizu and Tsujimoto (1994), i.e. `[cfk]` = 0.07 and `[cfe]` = 0.16 (see also Defina and Bixio, 2005).

SWASH offers the user the possibility to simulate a full three-dimensional turbulent flow where both non-hydrostatic pressure and anisotropic state of turbulent stresses becomes important. Additionally, the length scales in all directions can be in the same order. Examples are curved open channel flows with heterogeneous roughness conditions and compound channel flows with different floodplains. Their flow characteristics are represented by, amongst others, secondary currents (of Prandtl's second kind) generated due to anisotropy of turbulence.

For such cases a three-dimensional turbulence model should be activated, where the Reynolds stresses in all directions are equally important. These turbulent stresses are supposed to be linearly related to the deformation rates of the mean flow. This Boussinesq hypothesis introduces the concept of eddy viscosity that can be computed by means of a two-equation model. The standard $k-\varepsilon$ model of Launder and Spalding (1974) is employed, whereas the eddy viscosity is applied to all directions. The Boussinesq hypothesis can be considered as the leading term in a series expansion of products of strain and rotation tensors implying isotropy of turbulence. This works quite well for many flows where the primary shear stress is the dominant one. This is, however, not the case when secondary shear stresses and normal stresses become relevant. Hence, the Boussinesq hypothesis may not be suitable for turbulent flows involving strong three-dimensional effects. This weakness can potentially be removed by expanding the stress-strain relationship to include the quadratic terms of the mean velocity gradient tensor. These terms approximate the deviations from the isotropic state of the turbulent normal stresses. To this end, the nonlinear $k-\varepsilon$ model of

Speziale (1987) is applied.

The default option is a constant horizontal eddy viscosity.

`HORIZONTAL` indicates that the horizontal mixing will be activated.

`CONSTANT` this option indicates that a constant horizontal eddy viscosity will be adopted.

`[visc]` constant viscosity value (in m^2/s).

`SMAGORIN` indicates that the Smagorinsky model will be employed.

`[cs]` Smagorinsky constant.
Default: `[cs]` $= 0.2$

`MIXING` indicates that the Prandtl mixing length hypothesis will be used.

`[lm]` mixing length (in meters).

`VERTICAL` indicates that the vertical mixing will be activated.

`KEPS` indicates that the standard $k - \varepsilon$ model will be used.

`[cfk]` vegetative drag-related constant for turbulent kinetic energy.
Default: `[cfk]` $= 0.07$
Note that this constant is only relevant when vegetation is included (see command `VEGETATION`).
ONLY MEANT FOR STRUCTURED GRIDS.

`[cfe]` vegetative drag-related constant for dissipation rate.
Default: `[cfe]` $= 0.16$
Note that this constant is only relevant when vegetation is included (see command `VEGETATION`).
ONLY MEANT FOR STRUCTURED GRIDS.

`FULL` indicates that the full three-dimensional turbulence modelling will be activated.
Note that this option is in lieu of the above mixing options.
ONLY MEANT FOR STRUCTURED GRIDS.

`KEPS` indicates that the standard $k - \varepsilon$ model will be used.

`LINEAR` indicates that the linear stress-strain relation will be employed. This amounts to the application of the Boussinesq hypothesis and is the default.

`NONLINEAR` indicates that the nonlinear stress-strain relationship of Speziale (1987) will be used.

```
POROsity [size] [height] [alpha0] [beta0] [wper]
```

CANNOT BE USED IN CASE OF <u>UNSTRUCTURED GRIDS</u>.

This command indicates the use of porosity layers inside the computational domain to simulate full/partial reflection and transmission through porous structures such as breakwaters, quays and jetties. Also the interaction between waves and porous coastal structures can be simulated in this way. The mean flow through porous medium is described by the volume-averaged Reynolds-averaged Navier-Stokes (VARANS) equations. The laminar and turbulent frictional forces in porous medium is modelled by means of the empirical formula's of Van Gent (1995). In the case of an oscillatory wave motion the turbulent loss will enhance, which depends on the Keulegan-Carpenter number.

See commands `INPGRID POROSITY` and `READINP POROSITY` in order to define porosity layers. If neither of this command nor the command `READINP POROSITY` is used, SWASH will not account for wave interactions with porous structures.

`[size]` characteristic grain size of porous structure(s) (in m).
Note that `[size]` is allowed to vary over the computational region; in that case use the commands `INPGRID PSIZE` and `READINP PSIZE` to define and read the grain sizes of different porous structures. The value of `[size]` in this command is then not required (it will be ignored).

`[height]` structure height (relative to the bottom in meters). Both submerged and emerged porous structures can be defined in this way. An emerged structure is the default.
Default: `[height]` = 99999.
Note that `[height]` is allowed to vary over the computational region; in that case use the commands `INPGRID HSTRUC` and `READINP HSTRUC` to define and read the structure heights of different structures. The value of `[height]` in this command is then not required (it will be ignored).

`[alpha0]` dimensionless constant for laminar friction loss (surface friction).
Default: `[alpha0]` = 200.

`[beta0]` dimensionless constant for turbulent friction loss (form drag).
Default: `[beta0]` = 1.1

`[wper]` characteristic wave period (either mean of peak period in s).
In case of wave interaction with porous structures this parameter is required.

```
VEGEtation < [height] [diamtr] [nstems] [drag] > INERtia [cm] POROsity Vertical
```

CANNOT BE USED IN CASE OF UNSTRUCTURED GRIDS.

With this optional command the user can activate wave damping induced by aquatic vegetation. If this command is not used, SWASH will not account for vegetation effects.

The vegetation (rigid plants) can be divided over a number of vertical segments and so, the possibility to vary the vegetation vertically is included. Each vertical segment represents some characteristics of the plants. These variables as indicated below can be repeated as many vertical segments to be chosen.

The vegetation effect is due to the drag force on a fixed body in an oscillatory flow which can be determined using the well-known Morison equation. In this case only vertical cylinders are considered and the direction of the drag force is horizontal. Apart from the drag force, the inertia force can be optionally included, which is specified by means of the added mass coefficient. Note that this coefficient is one less than the inertia coefficient (=Froude-Krylov force + added mass) and is uniform over the plant.

For the case of densely spaced cylinders, like dense mangrove fields and porous brushwood groins, the effect of porosity can be optionally included. This porosity depends on the spatial occupation of vegetation per unit volume.

In case horizontal cylinders are included (e.g. box filled with branches or mangrove with both horizontal and vertical roots), the drag forces in both the horizontal and vertical directions must be modelled.

`[height]` the plant height per vertical segment (in m).

`[diamtr]` the diameter of each plant stand per vertical segment (in m).

`[nstems]` the number of plant stands per square meter for each segment. Note that `[nstems]` is allowed to vary over the computational region to account for the zonation of vegetation. In that case use the commands `INPGRID NPLANTS` and `READINP NPLANTS` to define and read the vegetation density. The (vertically varying) value of `[nstems]` in this command will be multiplied with this horizontally varying plant density.
Default: `[nstems]` = 1

`[drag]` the drag coefficient per vertical segment.

`INERTIA` indicates that the inertia force will be included.

`[cm]` the added mass coefficient.

`POROSITY` indicates that the porosity effect will be included.

`VERTICAL` indicates that the drag force in vertical direction will be included.

```
CORIolis [fpar]
```

With this optional command the Coriolis force will be included in the momentum equations. If this command is not used, SWASH will not account for Coriolis effects.

In case of a Cartesian grid (either rectilinear or curvilinear one), the latitude of the (usually small) model area must be specified by the user (see the command `SET ...[latitude]`). For a spherical grid the Coriolis parameter will be computed from the local latitude coordinates. Note that the Coriolis force in a spherical model varies only in the North-South direction.

Alternatively, the user may specify a constant or space varying Coriolis parameter other than the latitude-dependent one (e.g., at laboratory scale), see below.

`[fpar]` constant Coriolis parameter (in s^{-1}).
Note that a positive value corresponds to the Northern hemisphere and a negative value to the Southern hemisphere.
Also note that `[fpar]` is allowed to vary over the computational domain. In that case use the commands `INPGRID CORIOLIS` and `READINP CORIOLIS` to define and read the Coriolis parameter. The value of `[fpar]` in this command is then not required (it will be ignored).

```
                        | -> Sec |    | -> NONCohesive [size]            |
TRANSPort [diff] [retur] <     MIn  >  <                                  > &
                        |    HR  |    | COHesive [tauce] [taucd] [erate] |
                        |    DAy |

                                  | -> Yes  |
          [fall] [snum] [ak]  DENSity <          >
                                  | No      |

          ANTICreep  None | STAndard | SVK
```

With this optional command the user can specify some relevant parameters in case of transport of constituent. These parameters are only relevant when transport of salinity, temperature, or suspended sediment load is included.

Inclusion of transport of constituent is indicated by the commands `INPTRANS SALINITY` and `READTRANS SALINITY` in case of salinity, or the commands `INPTRANS TEMPERATURE` and `READTRANS TEMPERATURE` in case of temperature, or the commands `INPTRANS SEDIMENT` and `READTRANS SEDIMENT` in case of suspended sediment load. Using these commands, both the initial and stationary boundary conditions for constituent are thus specified. If none of these commands is used, SWASH will not account for the transport of any of these constituents.

The first parameter that may be specified in this command is the horizontal eddy diffusivity. A uniform eddy diffusivity value may be chosen that can be used as a calibration parameter to account for all forms of unresolved horizontal mixing. This parameter may be chosen independently from the eddy viscosity (see command `VISCOSITY HOR`). The eddy diffusivity depends on the flow and the grid size used in the simulation. A typical small-scale model with grid sizes of tens of meters or less, the eddy diffusivity typically ranges from 1 to 10 m^2/s. For a large-scale (tidal) areas with grid sizes of at least hundreds of meters, the parameter is typically in the range of 10 to 100 m^2/s. Alternatively, when not specified, the eddy diffusivity is related to the eddy viscosity that is determined by either the Smagorinsky model or the Prandtl mixing length model. Otherwise it is zero.

Note that in 3D simulations the vertical eddy diffusivity is automatically included and is related to the vertical mixing (see command `VISCOSITY VERT`).

The second parameter in this command is the return time for unsteady salt intrusion in a tidal flow. A boundary condition at the seaward side is required. This is usually the ambient or background concentration of salt sea water. However, at the transition between sea and river, alternating conditions hold regarding inflow of salt sea water during flood tide and outflow of fresh river water during ebb tide. Immediately after low water, the salinity of the inflowing water will not be equal to the salinity of the sea water. It will take some time before this happens at the boundary. This time lag is the return time for salinity from its value at the outflow depending on conditions in the interior of model domain relative to its background value specified at the inflow, see Figure 4.2. This memory effect is characterised by the so-called Thatcher-Harleman condition that specifies an appropriate time lag. The return time depends on the tidal flow conditions outside the estuary.

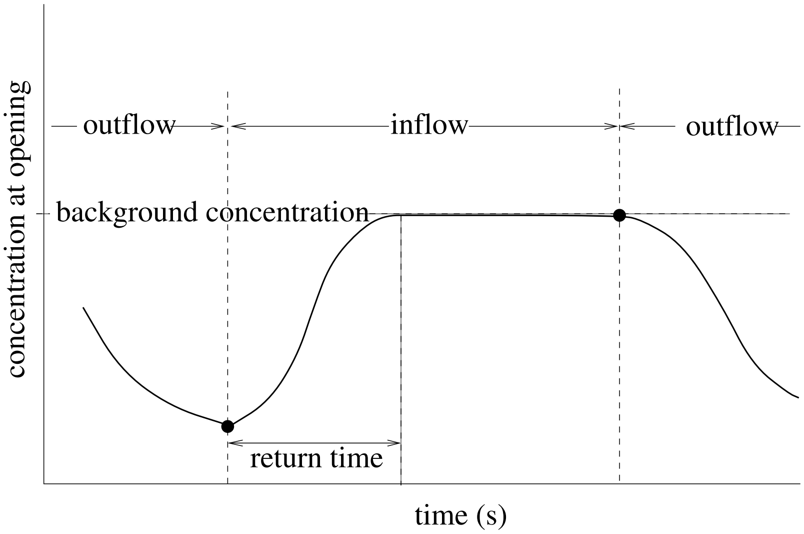

Figure 4.2: Return time for unsteady salt intrusion.

Either noncohesive suspended sediment (sand) or cohesive suspended sediment (mud or clay) transport can be considered. We assume a single sediment class. Bed load is not taken into account. In case of noncohesive suspended load transport, median sediment diameter can be specified. In this respect, a pickup function is employed to model the upward sediment flux in which the amount of noncohesive sediment (sand) is eroded from the bed surface into the flow. The following pickup function is used (Van Rijn, 1984):

$$-\frac{\nu_t}{\sigma_c}\frac{\partial c}{\partial z} = 0.00033 \left(\frac{\theta - \theta_{\mathrm{cr}}}{\theta_{\mathrm{cr}}}\right)^{3/2} \frac{(s-1)^{0.6}\, g^{0.6}\, d_{50}^{0.8}}{\nu^{0.2}} \, , \qquad \theta > \theta_{\mathrm{cr}}$$

with c the (volumetric) sediment concentration, ν_t the vertical eddy viscosity (see command `VISCOSITY VERT`), σ_c the Schmidt number for sediment, θ the Shields parameter related to the bed shear stress, $\theta_{\mathrm{cr}} = 0.05$ the critical Shields parameter, $s = 2.65$ the sediment specific gravity (see command `SET [rhosed]`), d_{50} the median sediment diameter, and ν the kinematic viscosity of water. Sediment deposition is determined by the downward flux related to the settling velocity w_s. If this fall velocity is not specified by the user, then the fall velocity will be calculated by SWASH depending on the particle size (Rubey, 1933):

$$w_s = \sqrt{(s-1) g d_{50}} \left(\sqrt{\frac{2}{3} + \frac{36\nu^2}{(s-1) g d_{50}^3}} - \sqrt{\frac{36\nu^2}{(s-1) g d_{50}^3}} \right)$$

Note that if the sediment diameter is not specified by the user, then no mass exchange between the bed and the flow will be taken into account.

For cohesive sediment the mass exchange of suspended load between the bed and the flow are calculated with the well-known Partheniades-Krone formulations, which include the erosion and deposition fluxes:

$$-\frac{\nu_t}{\sigma_c}\frac{\partial c}{\partial z} = S_e - S_d$$

The sediment flux for erosion is given by

$$S_e = E \left(\frac{\tau_b}{\tau_{ce}} - 1\right) \, , \qquad \text{if } \tau_b > \tau_{ce}$$

where E is the entrainment rate for erosion flux, τ_{ce} is the critical bed shear stress for erosion and τ_b is the actual bed shear stress. The sediment flux for deposition is given by

$$S_d = c_b \, w_s \left(1 - \frac{\tau_b}{\tau_{cd}}\right) \, , \qquad \text{if } \tau_b < \tau_{cd}$$

where c_b is the (volumetric) sediment concentration near the bed and τ_{cd} is the critical bed shear stress for sedimentation. Here, the critical bed shear stresses for erosion, τ_{ce}, and sedimentation, τ_{cd}, the sediment erosion rate E and the fall velocity w_s must be specified by the user.

It is assumed that the interaction between sediment and turbulent flow is mainly governed by sediment-induced buoyancy effects. In this respect, the standard $k - \varepsilon$ model and the logarithmic wall law near the bed surface must be applied. This wall law is used to calculate the bed shear stress τ_b, which in turn serves as one of the parameters for the mass exchange between the bed and the flow. For sand transport the roughness height may depend on the sediment diameter (if the user wants so) and is determined as $5.5\,d_{50}$.

Because of the assumption of the upward sediment flux being equal to the pickup rate, the Schmidt number σ_c for sediment becomes a free parameter. Experiences have shown that sediment diffusivity is rather sensitive to this parameter. The sediment diffusivity is usually larger than the eddy viscosity, and so $\sigma_c < 1$.

The turbidity flow is usually considered as a mixture of water and sediment with a mixture density, i.e. the effect of sediment on the density of (salt) water is included. However, in some cases it may be desirable not to include this effect. In this case the density of water remains unchanged, while the sediment transport is only influenced by the flow and (turbulent) dispersion. Also, there is no sedimentation and erosion near the bed. Hence, sediment can be considered here as a passive tracer.

SWASH makes use of the terrain-following coordinates of which the advantages are a better representation of bottom topography and a better resolution in shallow areas. A disadvantage is the transformation of the transport equation due to the geometrical properties of the curved z-planes, so that the curvature terms are involved, which may complicate the computation. However, when these curvature terms are neglected, this may lead to a false generation of vertical mixing. This effect, known as the artificial creeping, becomes evident when the bottom slope is relatively large in regions of strong stable stratification. Hence, in such as case, inclusion of the curvature terms, known as the anti-creepage terms, may reduce significantly the artificial creeping. The standard method is based on the actual transformation. An alternative is the method of Stelling and Van Kester [3], which computes the horizontal diffusion along strictly horizontal planes.

`[diff]` constant horizontal eddy diffusivity (in m^2/s).

`[retur]` return time, the unit is indicated in the next option:

SEC	unit seconds
MIN	unit minutes
HR	unit hours
DAY	unit days

`NONCOHES` indicates that noncohesive suspended load transport (sand) will be activated.
This is the default.

`[size]` median sediment diameter (in μm).

Note that if [size] is not specified, no sedimentation and erosion will be taken place at the bed.

COHESIVE indicates that cohesive suspended load transport (mud) will be activated.

[tauce] critical bed shear stress for erosion (in N/m^2).

[taucd] critical bed shear stress for deposition (in N/m^2).

[erate] entrainment rate for erosion (in kg/m^2/s).

[fall] fall velocity (in mm/s).
In case of cohesive sediment transport this parameter is required.
For sand transport, [fall] may not be given by the user, which it will then be calculated based on the sediment diameter.

[snum] Schmidt number for sediment.
Default: [snum] = 0.7

[ak] empirical constant to reduce the Von Karman constant and thereby the bed shear stress in sediment-laden bottom boundary layer.
Adams and Weatherly (1981) suggest [ak] = 5.5.

DENSITY this option indicates whether to include the density effect of suspended sediment in the fluid mixture or not.

YES indicates that the mixture density will be computed.
This is the default.

NO indicates that the density of water will not be changed by the presence of sediment in the water column.

ANTICREEP this option indicates whether to include the anti-creepage terms or not.
NOT FOR UNSTRUCTURED GRIDS.

NONE indicates that the anti-creepage terms will not be included.
This is the default.

STANDARD indicates that the anti-creepage terms as derived from the local transformation will be employed.

SVK indicates that the Stelling and Van Kester method will be adopted.

```
BREaking [alpha] [beta]
```

With this optional command the user can control depth-limited wave breaking in the case of relatively coarse resolution in the vertical. If this command is not used, SWASH will not account for this control. Note that SWASH will account for energy dissipation due to wave breaking anyhow. Also note that the use of this command in the case of current-limited wave breaking (see command AMBIENT) is strongly discouraged.

By considering the similarity between breaking waves and bores or moving hydraulic jumps, energy dissipation due to wave breaking is inherently accounted for. However, when a few vertical layers are to be employed, the amount of this energy dissipation may be underestimated due to the inaccuracy with which the phase velocity at the front face of the breaking wave is approximated. To initiate the wave breaking process correctly, steep bore-like wave fronts need to be tracked and this can be controlled by the vertical speed of the free surface. When this exceeds a fraction of the shallow water celerity, as follows,

$$\frac{\partial \zeta}{\partial t} > \alpha \sqrt{gh}$$

the non-hydrostatic pressure in corresponding grid points is then neglected and remains so at the front face of the breaker. The parameter $\alpha > 0$ represents the maximum local surface steepness and determines the onset of the breaking process. A threshold value of $\alpha = 0.6$ is advised. (This corresponds to a local front slope of 25^o.) This single value is not subject to calibration and seems to work well for all the test cases we have considered, both regular and irregular waves.

To represent persistence of wave breaking (even if $\partial_t \zeta < \alpha\sqrt{gh}$), we also label a grid point for hydrostatic computation if a neighbouring grid point has been labelled for hydrostatic computation and the local steepness is still high enough, i.e.,

$$\frac{\partial \zeta}{\partial t} > \beta \sqrt{gh}$$

with $\beta < \alpha$. In all other grid points, the computations are non-hydrostatic.

This approach combined with a proper momentum conservation leads to a correct amount of energy dissipation on the front face of the breaking wave. Moreover, nonlinear wave properties such as asymmetry and skewness are preserved as well.

Note that by taking a sufficient number of vertical layers (10 or so) the phase velocity at the breaking front will be computed accurately enough and hence, this option should not be activated.

`[alpha]`	threshold parameter at which to initiate wave breaking. Note: `[alpha]` > 0. Default: `[alpha]` $= 0.6$
`[beta]`	threshold parameter at which to stop wave breaking. Note: $0 <$ `[beta]` $<$ `[alpha]`. Default: `[beta]` $= 0.3$

```
AMBient  [U] [V] [eta]
```

CANNOT BE USED IN CASE OF UNSTRUCTURED GRIDS.

With this optional command the user can specify spatially constant ambient current and mean water level to include their effect on waves. If this command is not used, SWASH will not account for the effect of ambient current on the wave dynamics unless the commands `INPAMB` and `READAMB` are activated.

`[U]` the $x-$component of the ambient current (in m/s).
Note that `[U]` is allowed to vary over the computational region; in that case use the commands `INPAMB AXV` and `READAMB ACURR` to define and read the associated data. The value of `[U]` in this command will be ignored.

`[V]` the $y-$component of the ambient current (in m/s).
Note that `[V]` is allowed to vary over the computational region; in that case use the commands `INPAMB AVY` and `READAMB ACURR` to define and read the associated data. The value of `[V]` in this command will be ignored.

`[eta]` the mean water level (in m).
Note that `[eta]` is allowed to vary over the computational region; in that case use the commands `INPAMB MWL` and `READAMB MWL` to define and read the associated data. The value of `[eta]` in this command will be ignored.

4.5.5 Numerics

```
NONHYDrostatic  /    STAndard \  [theta]                                 &
                \ -> BOX      /

                SUBGrid [pmax]  REDuced [qlay]                           &

                SOLVer [rhsaccur] [initaccur] [maxiter] [relax]          &

                PRECondItioner  ILUD|ILU                                 &

                PROJection  ITERative [tol] [maxiter]
```

With this optional command the user can include the non-hydrostatic pressure in the shallow water equations. If this command is not used, SWASH will not account for the non-hydrostatic pressure, i.e. pressure is assumed to be hydrostatic.

Hydrostatic pressure assumption can be made in case of propagation of long waves, such as large-scale ocean circulations, tides and storm surges. This assumption does not hold in case of propagation of short waves, flows over a steep bottom, unstable stratified flows, and other small-scale applications where vertical acceleration is dominant.

In SWASH two different schemes for the vertical pressure gradient are available, i.e., the classical central differencing (option `STANDARD`) and the Keller-box scheme (option `BOX`). The former approximation is particularly meant for applications where vertical structures are important, e.g. stratified flows with density currents, undertow and flows over steep and rapidly varying bottoms, while the latter will be mainly used for accurate short wave propagation with two or three layers employed. In addition, when applying the Keller-box scheme, it is advised to choose an equidistant layer distribution (variable thicknesses only) in order to get the optimal linear frequency dispersion accuracy (i.e. the relative error of the phase speed predicted by the model compared to that from the linear dispersion theory is minimal).

The time integration of the vertical pressure gradient is the so-called $\theta-$scheme (a mix of explicit and implicit Euler schemes). With `[theta] = 0.5` we have the well-known second order accurate Crank-Nicolson scheme with the smallest truncation error, while `[theta] = 1` indicates the first order implicit Euler scheme. Note that only values of `[theta]` in the range [0.5,1] are allowed for stability reasons.

The number of vertical layers is chosen sufficiently large to resolve the vertical structure of the flow field (see command `VERTICAL`). However, this grid resolution in the vertical is also employed in the discretization of the non-hydrostatic pressure field, whereas a few number of layers usually suffices to compute wave dispersion without deteriorating accuracy. With a subgrid approach the user has an option to resolve the flow and pressure field with a different resolution in the vertical. The vertical is equidistantly divided into a few number of layers for the non-hydrostatic pressure and a relative large number of (non-equidistant) layers for the horizontal velocities and turbulent stresses. In this way, the solution for the non-hydrostatic pressure and vertical acceleration can be obtained on a relatively coarse vertical grid, while on a subgrid with a high vertical resolution the vertical structure of turbulent flow is resolved. Note this subgrid approach cannot be applied in the case of pressurized flow.

Since most of the computational effort is devoted to inverting the Poisson pressure matrix, an effective way to minimize this effort is by reducing the rank of the Poisson matrix. This leads to less computational cost and memory. With the keyword `REDUCED` the dimension of the Poisson equation, i.e. the number of pressure unknowns per water column, can be diminished. Note this option must be combined with the `BOX` scheme. In spite of this reduction it can still provide an accurate description of dispersive waves. For instance, a model with two layers but one reduced pressure layer, the so-called reduced two-layer model, has more or less the same linear dispersion accuracy as the full two-layer model but saves about 30% CPU time. Note this option cannot be applied in the case of pressurized flow. The stencil of the pressure equation can be reduced by assuming a constant pressure in the vertical near the bottom. This is determined by the parameter `[qlay]` which indicates the number of layers, started from the bottom, where the pressure is constant. In this way, it is possible to eliminate the pressure in these layers from the set of equations. As an example, we have a model with five pressure layers (`[pmax] = 5`) and we assume that the pressure is constant in the two lowest layers (`[qlay] = 2`). So there are effectively three (upper) layers in which the pressure need to be computed. An important remark needs to be made. Although this reduced pressure equation method is more efficient in terms of CPU time, it is less so concerning the dispersion accuracy. To optimize this accuracy often a different than equidistant layer distribution needs to be sought. For instance, the optimal linear dispersion relation of the reduced two-layer model (`[kmax] = [pmax] = 2`, `[qlay] = 1`) is obtained by setting the following layer distribution: the top layer has a relative thickness of 84% whereas the thickness of the bottom layer is 16% of the total depth.

By inclusion of the non-hydrostatic pressure, a solution of the Poisson pressure equation is required. In SWASH this equation is solved by an iterative solution method and the user may controls this by means of the keyword `SOLVER`. Two linear solvers are adopted: Strongly Implicit Procedure (SIP) and BiCGSTAB preconditioned with an incomplete LU factorization. The former one is particularly meant for the depth-averaged case, while the latter one is to be applied for the multi-layered case. The incomplete LU factorization is either ILU or ILUD. The latter is restricted to the main diagonal of the matrix, i.e. the off-diagonals remain unchanged. The choice of the preconditioner is indicated with the keyword `PRECONDITIONER`. For parallel computing, the ILUD preconditioner is a good choice. However, the ILU preconditioner is more robust. For instance, when high waves or very short waves are involved, or when the bottom topography exhibits steep slopes, or when a considerable number of layers (> 20) is involved, it may be wise to choose the ILU preconditioner. Note this ILU preconditioner will be chosen automatically in the case of pressurized flow and/or unstructured triangular meshes. The weighting parameter α, as indicated by `[relax]`, may improve the rate of convergence. With this parameter a combination of the classical ILU and the modified ILU (MILU) can be given. This combination is given by (1-α)ILU + αMILU, and is also hold for ILUD and its modified variant (MILUD). Based on several numerical experiments, an optimum in the convergence rate is found by taking 55% of MILU and 45% of ILU in case of the Keller-box scheme, and 99% of MILUD and 1% ILUD for the same scheme, while for the standard discretization of the vertical pressure gradient, a combination of 90% of MILU(D) and 10% of ILU(D) is chosen. Note that in the case of unstructured triangular meshes, the default value is the ILU preconditioner (thus 0% MILU).

It is common to use the reduction of the residual as a stopping criterion, because the BiCGSTAB method requires calculation of the residual. When solving the system $Ax = b$, after m iterations we have an approximate solution x_m and the residual $r_m = b - Ax_m$ is related to the convergence error $e_m = x - x_m$ by $Ae_m = r_m$, so the reduction of the residual results in the reduction of the convergence error. This does not necessarily mean that the relative error of the solver is identical to the decrease of the residual. The iteration process stops at each time step if the ratio of the 2-norm of the residual and of the right-hand side or initial residual is less than a given accuracy: $\|r_m\|_2/\|b\|_2 < \epsilon$ and $\|r_m\|_2/\|r_0\|_2 < \epsilon$, respectively. They are indicated by the parameters `[rhsaccur]` and `[initaccur]`, respectively. If both these accuracies are given, the sum of the two is used as termination criterion. Often, the stopping criterion for the iterative methods is basically a compromise between efficiency and accuracy. Decreasing the required accuracy can save a considerable amount of CPU time. Numerical experiments showed $\epsilon = 0.01$ gives the optimum.

The Poisson pressure equation is obtained by means of the pressure correction method of Van Kan (1986). This method is second order accurate in time and thus appropriate for the simulation of wave transformation, or in general, free surface flows. To deal with the pressurized flow underneath a floating object, the user is advised to choose the first order pressure projection method of Chorin (1968). However, to retain the second order accuracy and to avoid wave damping when simulating free surface flows, the accuracy of the pressure projection method can be improved. This is accompanied by an iteration process to solve the global continuity equation containing the contribution of the non-hydrostatic pressure. This equation is then consistent with the divergence-free velocity field obtained after the pressure projection step (Vitousek and Fringer, 2013).

The default option is the Keller-box scheme with `[theta] = 1.0`, while the Poisson pressure equation obtained with the second order pressure correction technique is solved with either SIP with `[rhsaccur]=0.01`, in the case of depth-averaged mode, or ILUD-BiCGSTAB with `[rhsaccur]=0.01` and `[initaccur]=0`, in the case of multi-layered mode. For unstructured meshes, the ILU-BiCGSTAB solver is chosen instead.

`STANDARD` this option indicates that the classical central differencing for the vertical pressure gradient will be employed.

`BOX` this option indicates that the Keller-box scheme will be adopted. This is the default.

`[theta]` parameter for the $\theta-$method. `[theta] = 0.5` corresponds to the Crank-Nicolson scheme and `[theta] = 1` to the implicit Euler scheme.
Default: `[theta]` = 1.0

`SUBGRID` this option indicates that the subgrid approach will be adopted.
Note: this option cannot be applied in the case of pressurized flow.
ONLY MEANT FOR STRUCTURED GRIDS.

`[pmax]` number of vertical pressure layers and $1 \leq$ `[pmax]` $\leq$ `kmax`.

Default: `[pmax] = kmax`

`REDUCED` this option indicates that the reduced pressure equation method will be adopted.
Note: this option cannot be applied in the case of pressurized flow.

`[qlay]` number of layers, started from the bottom layer, for which pressure is constant in the vertical to reduce the pressure equation.
Default: `[qlay]` = 1

`SOLVER` with this option the user can influence some of the numerical parameters of iterative solvers.

`[rhsaccur]` relative accuracy with respect to the right-hand side.
Default: `[rhsaccur]` = 0.01

`[initaccur]` relative accuracy with respect to the initial residual.
Default: `[initaccur]` = 0.0

`[maxiter]` maximum number of iterations.
Default: `[maxiter]` = 500

`[relax]` parameter for a linear solver:

- relaxation parameter for the ILU(D) preconditioning. `[relax] = 0` corresponds to ILU(D) and `[relax] = 1` corresponds to MILU(D).
Default: model dependent (see above).
- under relaxation parameter for the SIP solver.
Default: `[relax]` = 0.91

`PRECONDI` with this option the user can choose a preconditioner.
ONLY MEANT FOR STRUCTURED GRIDS.

`ILUD` indicates the ILUD preconditioner.
This is the default.

`ILU` indicates the ILU preconditioner.

`PROJECTION` with this option the user can choose the first order pressure projection method instead of the default second order pressure correction method.

`ITERATIVE` with this option the accuracy of the pressure projection method is improved through an iteration process.

`[tol]` error tolerance to terminate the iteration process.
Default: `[tol]` = 0.0001

`[maxiter]` maximum number of iterations.
Default: `[maxiter]` = 50

```
|         | UMOM  MOMentum|HEAD  / -> Horizontal   |
|         |                      \    Vertical     |
| UPWind <                                         |
|         | WMOM  / -> Horizontal                  |
|         |       \    Vertical                    |
                |                                                  |
                | CORRdep                                          |
DISCRETization <                                                    >      &
                | TRANSPort  / -> Horizontal                       |
                |            \    Vertical                         |
                |                                                  |
                |           | -> Umom                              |
                | ACURrent <                                       |
                |           | Wmom                                 |

                   | NONe                                 |
                   |                                      |
                   | FIRstorder                           |
                   |                                      |
                   | HIGherorder [kappa]                  |
                   |                                      |
                   |          | -> SWEBy [phi]            |
                   |          |                           |
                   | LIMiter <  RKAPpa [kappa]            |
                   |          |                           |
                   |          | PLKAPpa [kappa] [mbound]  |
                   | FROmm                                |
                   |                                      |
                   | -> BDF|LUDs                          |
                   |                                      |
                  <  QUIck                                 >
                   |                                      |
                   | CUI                                  |
                   |                                      |
                   | MINMod                               |
                   |                                      |
                   | SUPerbee                             |
                   |                                      |
                   | VANLeer                              |
                   |                                      |
                   | MUScl                                |
                   |                                      |
                   | KORen                                |
                   |                                      |
                   | SMArt                                |
```

With this optional command the user can influence the spatial discretization.

For advection-dominated flows it is possible that wiggles in the solution arise. In that case upwind discretization might be the appropriate choice. Three types of upwind schemes are implemented:

- the standard first order upwind scheme
- higher order upwind schemes obtained with the $\kappa-$formulation
- TVD schemes with several classes of flux limiters:
  - Sweby $\Phi-$limiter
  - R$-\kappa$ limiter
  - PL$-\kappa$ limiter

Schemes of up to third order accuracy can be constructed by piecewise polynomial interpolation, the so-called $\kappa-$formulation. For all values of $\kappa \in [-1, 1]$, a blended form arises between second order backward difference scheme (BDF) and second order central differencing. The schemes BDF, QUICK and CUI are obtained by setting $\kappa = -1$, $1/2$ and $1/3$, respectively. The value $\kappa = 0$ gives the Fromm's scheme, while $\kappa = 1$ corresponds to central differencing. For $\kappa \neq 1/3$ the local truncation error is of second order; for $\kappa = 1/3$ it is of third order. By employing structured grids, all upwind schemes are applied in each computational direction. Hence no streamline upwinding is used.

Implementation of high resolution and TVD schemes on unstructured meshes is by far non-trivial. Especially, the so-called $r-$ratio (measuring the local monotonicity and an argument for the flux limiter) as it should comply the principles of the TVD theory. In UnSWASH, the $r-$ratio formulation of Casulli and Zanolli (2005) is employed.

In case of transport equations for salinity, temperature, or suspended sediment (keyword `TRANSPORT`) a TVD scheme must be applied to prevent wiggles in the solution or to avoid negative concentration values.

The water depth in velocity points is not uniquely defined. An appropriate approximation is based on first order upwinding instead of the usual interpolation. To achieve second order accuracy in space, we add a higher order interpolation augmented with a flux limiter. See keyword `CORRDEP`.

Conservation properties become crucial for rapidly varied flows. These properties are often sufficient to get solutions that are acceptable in terms of local energy losses, location of incipient wave breaking, propagation speed of a bore, etc. In flow expansions, the horizontal advective terms in $u-$ and $v-$momentum equations are approximated such that they are consistent with momentum conservation; see option `MOMENTUM`. In flow contractions, the approximation is such that constant energy head is preserved along a streamline, i.e. the Bernoulli equation; see option `HEAD`.

With respect to unstructured triangular meshes, a robust and efficient upwind-biased scheme for the horizontal advection terms in the momentum equation has been implemented [5]. The scheme complies with the Rankine-Hugoniot jump relations and is specifically designed with a view to preserving the local momentum flux. This is crucial to the simulation of breaking waves and unsteady bores.

The default option is the second order backward difference (BDF) scheme ($\kappa = -1$) for all horizontal advective terms in both $u/v-$momentum and $w-$momentum equations, and the MUSCL scheme for the vertical term in the $u/v-$momentum equation and also in the $w-$momentum equation. The exception is when the `BREAK` command is employed, which in that case the central differences ($\kappa = 1$) are applied to all horizontal advective terms in the $u/v-$momentum equations.

In addition, the water depth in velocity points is approximated with the MUSCL limiter. All the advection terms in any transport equation are approximated with the second order Van Leer limiter.

By default, SWASH decides whether energy head or momentum conservation is to be applied, though restricted to structured meshes. In principle, energy head conservation will be applied only in strong flow contractions, while elsewhere the momentum conservation is applied.

`UPWIND` indicates the type of discretization for momentum equations.
`UMOM` indicates the discretization employed for $u/v-$momentum equation.
`MOMENTUM` indicates that momentum must be conserved everywhere.
`HEAD` indicates that energy head must be conserved everywhere.
`HORIZONTAL` indicates the discretization of the horizontal advective terms.
`VERTICAL` indicates the discretization of the vertical advective terms.
`WMOM` indicates the discretization employed for $w-$momentum equation.
`CORRDEP` indicates the type of discretization for water depth in velocity points.
`TRANSPORT` indicates the discretization employed for transport equation(s).
`ACURRENT` indicates the discretization of the advection terms added to the momentum equations due to the ambient current.
`NONE` indicates that no upwinding is applied, hence the standard central difference scheme is used.
This is the default in case of vertical advective terms.
`FIRSTORDER` indicates that the standard first order upwind scheme is used.
`HIGHERORDE` a higher order upwind $\kappa-$scheme is activated.
`[kappa]` parameter defining the type of $\kappa-$scheme used.
For instance, `[kappa]` = 0.5 gives the QUICK scheme
and `[kappa] = 0` the Fromm's scheme.

Note: $-1 \leq$ [kappa] ≤ 1.
LIMITER indicates that a TVD scheme with flux limiter must be applied.
SWEBY a Sweby limiter is activated.
[phi] parameter defining the type of Sweby limiter used. For instance,
[phi] $= 1$ defines the minmod limiter.
Note: $1 \leq$ [phi] ≤ 2.
RKAPPA an R$-\kappa$ limiter is activated.
PLKAPPA a PL$-\kappa$ limiter is activated.
[mbound] parameter indicating the upper bound of the PL$-\kappa$ limiter.
Note: [mbound] > 0.
FROMM indicates the Fromm's scheme.
BDF indicates the second order backward upwind scheme.
This is the default in case of horizontal advective terms.
QUICK indicates the QUICK scheme.
CUI indicates the CUI scheme.
MINMOD indicates the minmod limiter.
SUPERBEE indicates the superbee limiter.
VANLEER indicates the Van Leer limiter.
MUSCL indicates the MUSCL limiter.
This is the default in case of CORRDEP.
KOREN indicates the Koren limiter.
SMART indicates the SMART limiter.

```
           |    MIN
           |
           | -> MEAN
BOTCel    <
           |    MAX
           |
           |    SHIFt
```

CANNOT BE USED IN CASE OF UNSTRUCTURED GRIDS.

With this optional command the user can determine how the bottom levels need to be computed in cell centers.

The bottom levels are usually defined in the corners of computational cells. These bottom levels have been extracted from the input grid (see command INPGRID BOTTOM). Nevertheless, SWASH uses a staggered grid and to determine the total water depth at water level points, a bottom level in the cell center is then required. As a rule, we take the average bottom level from the surrounding bottom corners to determine the total depth at cell center. This is fine for many cases. However, in the vicinity of steep bottom slopes, use of the average bottom level may lead to an inaccurate flooding and drying process (e.g. too much or too less volume of water left on tidal flats or flood plains) or inaccurate tide propagation. Another example is the inaccurate computation of pressure in front of the vertical wall which may negatively affect wave overtopping. For those situations it is more appropriate to take the bottom level at cell center being equal to the minimum of the surrounding bottom levels at the corner points. This so-called tiled bottom approach is also suitable to represent a small channel of one grid cell width.

Alternatively, it is possible to specify the bottom level at cell center as the maximum of the surrounding bottom levels or being shifted from the upper-right corner of the computational cell. This latter implies that, if the bottom input grid is identical to the computational grid in terms of resolution and orientation, the user-defined bottom values are specified on input at cell centers (instead of upper-right corners).

Note that for determining the bottom level in cell centers the positive downwards orientation is considered. So the minimum operation results into the shallowest bottom level. As a consequence, when the bottom level is above the reference level, i.e. a negative value, the maximum operation should be chosen, instead of the minimum one, in the context of the tiled bottom approach.

The default option is `MEAN`.

`MIN` the bottom level in cell center is the minimum of the four surrounding bottom corner points.
`MEAN` the bottom level in cell center is the average of the four surrounding bottom corner points.
`MAX` the bottom level in cell center is the maximum of the four surrounding bottom corner points.
`SHIFT` the bottom level in upper-right corner is shifted to the cell center.

```
            | EXPL [cfllow] [cflhig]                                        |
TIMEI METH <                                                                 > &
            | IMPL [thetac] [thetas] SOLVer [tol] [maxiter] [weight] NEWTon |

      VERTical [thetau] [thetaw] [thetat]
```

With this optional command the user can influence the time integration.

Note that the default option depends on the choice of the computational mesh, either structured or unstructured (see below).

If time integration is explicit, a time step restriction must be applied based on a Courant number associated with the long wave speed. For definition, see Appendix A. Note that a maximum Courant number of 0.5 is advised in case of high waves, nonlinearities (e.g. wave breaking, wave-wave interactions), and wave interaction with structures with steep slopes (e.g. quays, piers).

An automatic time step control is implemented as follows. The actual maximum of the Courant number over all wet grid points is determined. The time step is halved when this number becomes larger than a preset constant `[cflhig]` < 1, and the time step is doubled when this number is smaller than another constant `[cfllow]`, which is small enough to be sure the time step can be doubled. Information on the actual time step changes is provided in the `PRINT` file.

If time integration is semi-implicit, then the gradient of the water level in the momentum equations and the velocity divergence in the (global) continuity equation are discretized implicitly by means of the $\theta-$method, while both the horizontal advective and viscosity terms are discretized explicitly. As a consequence, the stability of the method will not depend upon the long wave speed. However, the time step will be restricted owing to the explicit treatment of the horizontal advective terms, although this restriction is mild. This method is particular useful and efficient for the simulation of three-dimensional circulation driven by buoyancy, tides and winds, combined with unstructured meshes, as the wave Courant number can easily be larger than 1 while still providing sufficiently accurate solution.

This semi-implicit time stepping requires the solution of a system of linear equations to obtain the water levels. This system is symmetric and positive definite and can be solved efficiently by using a preconditioned conjugate gradient (PCG) method. The keyword `SOLVER` controls the use of this iterative method. The iteration process stops if the norm of the residual falls below a small fraction of the initial residual; this small fraction is the user prescribed error tolerance `[tol]`. The preconditioner is a weighted combination of ILUD and its modified variant MILUD in the case of structured grids and a weighted combination of ILU and MILU in the case of triangular meshes. For details on the weighting parameter α, which may improve the convergence, see command `NONHYDROSTATIC`. This parameter is indicated by `[weight]`.

Regarding the wet areas falling dry, the flow Courant number is required to be smaller or equal 1 [2]. Consult Section 5.4.6 for further details. Hence, this poses no additional restriction on the time step in the case of explicit time integration. In contrast, there is a risk that the water depth will become negative when semi-implicit time integration is applied as a larger time step is allowed. As an alternative, one can instead enforce the depth to be non-negative during the whole simulation. This introduces a nonlinear volume term in the system of equations for the water levels that can be efficiently solved by a Newton-type iteration method (Casulli, 2009). This can be activated with command NEWTON.

When dealing with floating objects interacting with waves (e.g. a moored ship, WECs), the flow is locally pressurized. In this case explicit methods cannot be employed. Instead, the semi-implicit approach must be applied which amounts to the solution of a piecewise linear system of equations to obtain the water levels for free surface flow and the piezometric head for pressurized flow. Since each grid cell is labeled with either free surface or pressurized in a proper way, this system can still be solved efficiently by the PCG solver. Alternatively, the system of equations can also be solved by a nested Newton iteration method as described in Brugnano and Casulli (2009), which may allow a larger time step.

In the multi-layered mode, the vertical advective and viscosity terms in the momentum and transport equations are discretized implicitly by means of the $\theta-$method.

METHOD with this option the type of time integration is indicated.

EXPLICIT indicates that the time integration is explicit.
This is the default for structured grids.

[cfllow] minimum Courant number to be used for automatic time step control.
Default: [cfllow] $= 0.4$

[cflhig] maximum Courant number to be used for automatic time step control.
Default: [cflhig] $= 0.8$

IMPLICIT indicates that the time integration is semi-implicit.
This is the default for unstructured meshes.

[thetac] parameter for the $\theta-$method applied to the continuity equation.
[thetac] = 0.5 corresponds to the Crank-Nicolson scheme and
[thetac] = 1 to the implicit Euler scheme.
Default: [thetac] $= 0.5$

[thetas] parameter for the $\theta-$method applied to the water level gradient.
[thetas] = 0.5 corresponds to the Crank-Nicolson scheme and
[thetas] = 1 to the implicit Euler scheme.
Default: [thetas] $= 0.5$

SOLVER with this option the user can influence some of the parameters of
the preconditioned conjugate gradient method.

[tol] error tolerance to terminate the iteration process.
Default: [tol] $= 0.01$
Note: in case of Newton iteration, the default is [tol] $= 0.001$

[maxiter] maximum number of iterations.
Default: [maxiter] $= 100$

[weight] weighting parameter for the (M)ILU(D) preconditioning.
[weight] = 0 corresponds to ILU(D) and [weight] = 1 corresponds

to MILU(D).
Default: [weight] = 0.9
Note: in case of an unstructured mesh, the default is [weight] = 0.

NEWTON indicates that the (nested) Newton iteration method is employed.
This is the default for unstructured meshes.

VERTICAL with this option the time integration of vertical terms is indicated.

[thetau] parameter for the $\theta-$method in case of the $u/v-$momentum equation.
[thetau] = 0.5 corresponds to the Crank-Nicolson scheme and
[thetau] = 1 to the implicit Euler scheme.
Default: [thetau] = 0.5

[thetaw] parameter for the $\theta-$method in case of the $w-$momentum equation.
[thetaw] = 0.5 corresponds to the Crank-Nicolson scheme and
[thetaw] = 1 to the implicit Euler scheme.
Default: [thetaw] = 0.5

[thetat] parameter for the $\theta-$method in case of the transport equation.
[thetat] = 0.5 corresponds to the Crank-Nicolson scheme and
[thetat] = 1 to the implicit Euler scheme.
Default: [thetat] = 0.5

4.6 Output

There are two categories of output commands:

1. Locations
   commands defining sets of output locations at which the user requires output. Each set is indicated with a name ('sname' in this manual) which must be unique and not more than 8 characters long.

   Types of sets of output points:

   FRAME to define a set of output locations on a regular grid
   GROUP to define a set of output locations on a regular or curvilinear grid
   CURVE to define a set of output locations along a curve
   RAY to define a set of output locations along a depth or bottom contour line (with ISOLINE)
   ISOLINE to define a set of output locations along a depth or bottom contour line (with RAY)
   POINTS to define a set of isolated output locations

   Commands RAY and ISOLINE cannot be used in 1D-MODE. If one gives one name for two sets of output locations, the first set is lost (first in the sequence in the command file). Three special names BOTTGRID, COMPGRID and NOGRID are reserved for use by SWASH (see below). The user may not define sets with these names.

2. Write / plot
   commands defining data file output (write) at the above defined set(s) of output locations:

   BLOCK write spatial distributions (only for FRAMEs and GROUPs)
   TABLE write output for (set of) output location(s)

   Command BLOCK cannot be used in 1D-MODE.

4.6.1 Output locations

```
FRAme 'sname' [xpfr] [ypfr] [alpfr] [xlenfr] [ylenfr] [mxfr] [myfr])
```

With this optional command the user defines output on a rectangular, uniform grid in a regular frame.

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

'sname' name of the frame defined by this command

[xpfr] $x-$coordinate of the origin of the frame in problem coordinates
if Cartesian coordinates are used in m
if spherical coordinates are used in degrees (see command COORD)

[ypfr] $y-$coordinate of the origin of the frame in problem coordinates
if Cartesian coordinates are used in m
if spherical coordinates are used in degrees (see command COORD)

[alpfr] direction of the $x-$axis of the frame (in degrees, Cartesian convention; must be 0 in case of spherical coordinates)

[xlenfr] length of the frame in $x-$direction
if Cartesian coordinates are used in m
if spherical coordinates are used in degrees (see command COORD)

[ylenfr] length of the frame in $y-$direction
if Cartesian coordinates are used in m
if spherical coordinates are used in degrees (see command COORD)

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

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

Some output may be required on a frame that is identical with the bottom input grid or with the computational grid. These frames need not be defined by the user with this command FRAME; the frames are always generated automatically by SWASH under the names 'sname' = 'BOTTGRID' (for the bottom grid) and 'sname' = 'COMPGRID' (for the computational grid).

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

CANNOT BE USED IN CASE OF UNSTRUCTURED GRIDS.

With this optional command the user defines a group of output locations on a rectangular or curvilinear grid that is identical with (part of) the computational grid (rectilinear or curvilinear). Also, the flow variables (surface elevation, velocity components and pressure) will be outputted in their points of definition according to the Arakawa C-grid staggering. Such a group may be convenient for the user to obtain output that is not affected by interpolation errors.

Command CGRID should precede this command GROUP.

The subgrid contains those points (ix,iy) of the computational grid for which:
[ix1] $\leq$ ix $\leq$ [ix2] and [iy1] $\leq$ iy $\leq$ [iy2]

For convenience the size of the group, the corner coordinates and the angle with the problem coordinate system are written to PRINT file. The origin of the computational grid is (ix=1,iy=1)!

'sname' name of the set of output locations defined by this command
[ix1] lowest grid index of subgrid in terms of computational grid in ix-direction
[iy1] lowest grid index of subgrid in terms of computational grid in iy-direction
[ix2] highest grid index of subgrid in terms of computational grid in ix-direction
[iy2] highest grid index of subgrid in terms of computational grid in iy-direction

Limitations:
[ix1]$\geq$1, [ix2]$\leq$[mxc]+1, [iy1]$\geq$1, [iy2]$\leq$[myc]+1 ([mxc] and [myc] as defined in

the command `CGRID`).

```
CURve 'sname' [xp1] [yp1]  < [int] [xp] [yp] >
```

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

`'sname'` name of the curve
`[xp1],[yp1]` problem coordinates of the first point of the curve
if Cartesian coordinates are used in m
if spherical coordinates are used in degrees (see command `COORD`)
`[int]` SWASH will generate output at `[int]`−1 equidistant locations between two subsequent corner points of the curve
`[xp],[yp]` problem coordinates of a corner point of the curve. Repeat the group `[int] [xp] [yp]` in proper order if there are more corner points on the curve.

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

CANNOT BE USED IN 1D-MODE.

With this optional command the user provides SWASH with information to determine output locations along the depth contour line(s) defined subsequently in command `ISOLINE` (see below).

The locations are determined by SWASH as the intersections of the depth contour line(s) and the set of straight rays defined in this command `RAY`. These rays are characterized by a set of master rays defined by their start and end positions (`[xp],[yp]`) and (`[xq],[yq]`). Between each pair of sequential master rays thus defined SWASH generates `[int]`−1 intermediate rays by linear interpolation of the start and end positions.

Note that the rays thus defined have nothing in common with wave rays (e.g. as obtained from conventional refraction computations).

`'rname'` name of the set of rays defined by this command.
`[xp1],[yp1],` problem coordinates of the begin and end points of the first master ray

[xq1],[yq1] if Cartesian coordinates are used in m
if spherical coordinates are used in degrees (see command COORD)
[int] number of subdivisions between the previous master ray and the following master ray defined by the following data (number of subdivisions is one more than the number of interpolated rays)
[xp],[yp], problem coordinates of the begin and end points of each subsequent master ray
[xq],[yq] if Cartesian coordinates are used in m
if spherical coordinates are used in degrees (see command COORD)

```
                               | -> DEPth   |
ISOline  'sname'  'rname'  <               >  [dep]
                               |    BOTtom  |
```

CANNOT BE USED IN 1D-MODE AND IN CASE OF CURVILINEAR GRIDS.

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

'sname' name of the set of output locations defined by this command
'rname' name of the set of rays (as defined in command RAY)
DEPTH indicates the water depth, i.e. still water depth plus still water level (see command SET) or still water depth plus initial spatially varying water level (see command INPGRID WLEV).
BOTTOM indicates the still water depth, i.e. the depth with respect to datum level.
[dep] the depth (in meters) of the depth contour line along which output locations are generated by SWASH.

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

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

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

`'sname'` name of the points
`[xp],[yp]` problem coordinates of one output location
if Cartesian coordinates are used in m
if spherical coordinates are used in degrees (see command `COORD`)
`'fname'` name of the file containing the output locations.

4.6.2 Write or plot computed quantities

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

     [ref]                 (for output quantity TSEC)                  &

     [dur] SEC|MIN|HR|DAY  (for output quantities SETUP, HSIG, HRMS,
                            MVEL, MTKE and MSAL)                       &

     [depth]               (for output quantity HRUN)                  &

     [delrp]               (for output quantity RUNUP)                 &

     [xcom] [ycom] [zcom]  (for output quantities MOMX, MOMY, MOMZ)    &

     [alpobj]              (for output quantities FORCEX, FORCEY, FORCEZ,
                            MOMX, MOMY, MOMZ)                          &

      |-> PROBLEMcoord |
     <                 >   (for directions and vectors, e.g. VDIR and VEL)
      |   FRAME        |
```

With this command the user can influence or specify

- the naming of output quantities,
- the accuracy of writing output quantities,
- the definition of some output quantities,
- the duration over which wave parameters and mean current are calculated,
- the center of gravity and rotation angle of floating body for computing hydrodynamic loads, and

- reference direction for vectors.

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

the output parameters as given in commands `BLOCK` and `TABLE`.

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

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

`[lexp]` lowest expected value of the output quantity.

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

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

The following data are accepted only in combination with some specific output quantities.

`[ref]` reference time used for the quantity `TSEC`.
Default value: starting time of the first computation, except in cases where this is later than the time of the earliest input. In these cases, the time of the earliest input is used.

`[dur]` the time duration over which the wave parameters (e.g. wave height and setup), mean current or mean turbulence quantities are computed. This corresponds to the final stage of the simulation period, which should be long enough to establish steady-state conditions. The corresponding unit is indicated in the next option:

SEC	unit seconds
MIN	unit minutes
HR	unit hours
DAY	unit days

`[depth]` the total depth (in m) where inundation takes place. This can be used to compute the maximum runup height; see output quantity `HRUN`.

`[delrp]` the threshold depth (in m) for runup height calculation; see output quantity `RUNUP`.

`[xcom]` $x-$coordinate of center of mass of floating body (in m) with respect to the computational grid. To be used to compute the moments acting on the body; see output quantities `MOMX`, `MOMY` and `MOMZ`.
Default: `[xcom]` = 0.0

`[ycom]` $y-$coordinate of center of mass of floating body (in m).
Default: `[ycom]` = 0.0

`[zcom]` $z-$coordinate of center of mass of floating body (in m).
Default: `[zcom]` = 0.0

`[alpobj]` direction of the positive $x-$axis of the floating object (in degrees, Cartesian convention) with respect to the computational grid. To be used to compute the forces and moments acting on the rotated body.
Default: `[alpobj]` = 0.0

`PROBLEMCOORD`− vector components are relative to the $x-$ and $y-$axes of the problem coordinate system:

- directions are counterclockwise relative to the positive $x-$axis of the problem coordinate system if Cartesian direction convention is used (see command `SET`)
- directions are relative to North (clockwise) if Nautical direction convention is used (see command `SET`)

`FRAME` If output is requested on sets created by command `FRAME` or automatically (`COMPGRID` or `BOTTGRID`):

- vector components are relative to the $x-$ and $y-$axes of the frame coordinate system
- directions are counterclockwise relative to the positive $x-$axis of the frame coordinate system if Cartesian direction convention is used (see command `SET`)
- directions are relative to North (clockwise) if Nautical direction convention is used (see command `SET`)

Examples:

`QUANTITY Xp hexp=100.`	for simulations of lab. experiments
`QUANTITY WLEV VX excv=-9.`	to change the exception value for water level and $u-$component
`QUANTITY HSIG SETUP dur 30 min`	to compute wave height and setup by averaging surface elevation over the last 30 minutes of the simulation.
`QUANTITY MVEL MSALK dur 1 hr`	to compute depth-averaged mean velocity and mean salinity per layer by averaging these quantities over the last 1 hour of the simulation.
`QUANTITY HRUN depth 0.05`	the inundation depth is set to 5 cm.
`QUANTITY RUNUP delrp 0.1`	the minimum depth is set to 10 cm, above which the wave runup level is computed.
`QUANT MomX xcom=200.  ycom=0.  zcom=-1.`	the center of gravity of floating body is $(x_c, y_c, z_c) = (200, 0, -1)$ with respect

	to the origin of the computational grid meant to compute the roll moment.
`QUANTITY ForceX alpobj=30.`	the rotation angle of floating body is 30^o relative to the computational grid.
`QUANTITY VEL VDIR frame`	to obtain vector components and direction with respect to the frame.

```
OUTPut OPTIons 'comment' (TABle [field]) (BLOck [ndec] [len])
```

This command enables the user to influence the format of block and table output.

`comment` a comment character; is used in comment lines in the output
Default: `comment` = %

`field` length of one data field in a table. Minimum is 8 and maximum is 16.
Default: `field` = 12

`ndec` number of decimals in block (if appearing after keyword `BLOCK`).
Maximum is 9.
Default: `ndec` = 4

`len` number of data on one line of block output. Maximum is 9999.
Default: `len` = 6

```
                 | -> HEADer   |
BLOck  'sname'  <               > 'fname' (LAYout [idla])
                 |    NOHEADer |

           | DEP     |
           |         |
           | BOTL    |
           |         |
           | WATL    |
           |         |
           | DRAF    |
           |         |
           | VMAG    |
           |         |
           | VDIR    |
           |         |
           | VEL       |
           |           |
           | VKSI      |
           |           |
           | VETA      |
           |           |
           | PRESS     |
           |           |
           | NHPRES    |
           |           |
           | QMAG      |
           |           |
           | QDIR      |
           |           |
           | DISCH     |
           |           |
           | QKSI      |
           |           |
           | QETA      |
           |           |
           | VORT      |
           |           |
           | WMAG      |
           |           |
           | WDIR      |
           |           |
           | WIND      |
           |           |
           | FRC       |
           |           |
           | USTAR     |
           |           |
           | UFRIC     |
           |           |
           | SAL       |
           |           |
           | TEMP      |
           |           |
           | SED       |
           |           |
           | HRUN      |
           |           |
           | BRKP      |
           |         |
           | SETUP   |
           |         |
           | HS      |
           |         |
           | HRMS    |
           |         |
           | MVMAG   |
           |         |
           | MVDIR   |
           |         |
           | MVEL    |
           |         |
           | MVKSI   |
           |         |
           | MVETA   |
           |         |
           | MSAL    |
           |         |
           | MTEMP   |
           |         |
           | MSED    |
           |         |
           | ZK      |
           |         |
           | HK      |
           |         |
           | VMAGK   |
           |         |
           | VDIRK   |
           |         |
           | VELK    |
           |         |
           | VKSIK   |
           |         |                                          | -> Sec  |
     <    <  VETAK    >  [unit] > (OUTput [tbegblk] [deltblk]) <     MIn   >)
           |         |                                          |    HR   |
           | VZ      |                                          |    DAy  |
           |         |
           | VOMEGA  |
           |         |
           | SALK    |
           |         |
           | TEMPK   |
           |         |
           | SEDK    |
           |         |
           | TKE     |
           |         |
           | EPS     |
           |         |
           | VISC    |
           |         |
           | QMAGK   |
           |         |
           | QDIRK   |
           |         |
           | DISCHK  |
           |         |
           | QKSIK   |
           |         |
           | QETAK   |
           |         |
           | PRESSK  |
           |         |
           | NHPRSK  |
           |         |
           | MVMAGK  |
           |         |
           | MVDIRK  |
           |         |
           | MVELK   |
           |         |
           | MVKSIK  |
           |         |
           | MVETAK  |
           |         |
           | MSALK   |
           |         |
           | MTEMPK  |
           |         |
           | MSEDK   |
           |         |
           | MTKE    |
           |         |
           | MEPS    |
           |          |
           | MVISC    |
           |          |
           | TIME     |
           |          |
           | TSEC     |
           |          |
           | XP       |
           |          |
           | YP       |
           |          |
           | DIST     |
           |          |
           | FORCEX   |
           |          |
           | FORCEY   |
           |          |
           | FORCEZ   |
           |          |
           | MOMX     |
           |          |
           | MOMY     |
           |          |
           | MOMZ     |
           |          |
           | TRAX     |
           |          |
           | TRAY     |
           |          |
           | TRAZ     |
           |          |
           | ROTX     |
           |          |
           | ROTY     |
           |          |
           | ROTZ     |
           |          |
           | PTOP     |
           |          |
           | RUNUP    |
```

With this optional command the user indicates that one or more spatial distributions should be written to a file.

`'sname'` name of frame or group (see commands `FRAME` or `GROUP`)

`HEADER` with this option the user indicates that the output should be written to a file with header lines. The text of the header indicates run identification (see command `PROJECT`), time, frame name or group name (`'sname'`), variable and unit. The number of header lines is 8.
Note: the numerical values in the file are in the units indicated in the header.

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

`'fname'` name of the data file where the output is to be written to.
Default for option `HEADER` is the `PRINT` file. In case of `NOHEADER` the filename is required. Below a few remarks on the file formats.

Basically, the output files generated by SWASH are human-readable files that use ASCII character encoding. Such files can be open and edit in any text editor or can be viewed in Matlab or Excel. However, if the user specifies the extension of the output file as '**.mat**', a binary MATLAB file will be generated. This file requires less space on your computer and can be loaded in MATLAB much faster than an ASCII file. Also note that the output parameters are stored as single precision. (Hence, use the Matlab command `double` for conversion to double precision, if necessary.) Binary MATLAB files are particularly useful for the computation with unstructured grids. A number of MATLAB scripts are provided with the SWASH source code that can be used to plot wave parameters as maps in a simple way.

Since version 8.01, SWASH can generate VTK files that can be viewed in Paraview, an open-source, general-purpose visualization package, available for Windows, Mac and Linux (see https://www.paraview.org). The basic extension is '**.vtk**', but SWASH will generate various XML-based file formats depending on the grid types (***.vts** associated with structured grids and ***.vtu** containing unstructured mesh data). The VTK XML files are binary, self-descriptive (include plain text metadata) and portable (cross-platform). In addition, a key benefit is that there is no need to collect VTK files residing on separate processes of a distributed memory machine (after execution of SWASH in parallel). Paraview can simply visualize the whole domain that consists of several subdomains.

`LAY-OUT` with this option the user can prescribe the lay-out of the output to file with the value of `[idla]`.

`[idla]` see command `READINP` (options are: `[idla]`=1, 3, 4). Option 4 is recommended for postprocessing by MATLAB, however, in case of a generated binary MATLAB file option 3 is recommended.

Default: [idla] = 1

DEPTH water depth (in m) (not the still water depth!).

BOTLEV bottom level or still water depth (in m).
Output is in both active and non-active points.
Note: exception value for bottom levels must be given!
(See command INPGRID BOTTOM EXCEPTION).

WATLEV water level or surface elevation (in m).
Output is in both active and non-active points.
Note: exception value for water levels must be given!
(See command INPGRID WLEVEL EXCEPTION).

DRAFT draft of floating object (in m).
Output is in both active and non-active points.

VMAG velocity magnitude (in m/s).

VDIR velocity direction (in degrees; Cartesian or nautical; see command SET).

VEL flow velocity (vector; in m/s).

VKSI grid-oriented U-velocity (in m/s).
Note: not applicable to unstructured meshes.

VETA grid-oriented V-velocity (in m/s).
Note: not applicable to unstructured meshes.

PRESS pressure at bottom (in hPa).

NHPRES normalised (by density) non-hydrostatic pressure at bottom (in m^2/s^2).

QMAG magnitude of discharge per unit width (in m^2/s).

QDIR direction of discharge per unit width (in degrees; Cartesian or nautical).

DISCH discharge per unit width (vector; in m^2/s).

QKSI grid-oriented U-discharge per unit width (in m^2/s).
Note: not applicable to unstructured meshes.

QETA grid-oriented V-discharge per unit width (in m^2/s).
Note: not applicable to unstructured meshes.

VORT vorticity or rotation of the fluid in depth-averaged flow (in 1/s).

WMAG wind velocity at 10 m above sea level (in m/s).

WDIR wind direction at 10 m above sea level (in degrees; Cartesian or nautical).

WIND wind velocity at 10 m above sea level (vector; in m/s).

FRC bottom friction coefficient (see command FRICTION).

USTAR magnitude of friction velocity (in m/s).

UFRIC friction velocity (vector; in m/s).

SAL salinity (in ppt).

TEMP temperature (^{o}C).

SED suspended sediment (in kg/m^3).

HRUN represents the maximum horizontal runup by masking inundated points
as 1 and non-inundated points as 0.

BRKP represents area of wave breaking by masking breaking points as 1
and non-breaking points as 0.

SETUP wave-induced setup (in m).

HSIG	significant wave height (in m; for definition, see Appendix A).
HRMS	RMS wave height (in m; for definition, see Appendix A).
MVMAG	time-averaged or mean velocity magnitude (in m/s).
MVDIR	time-averaged or mean velocity direction (in degrees; Cartesian or nautical; see command SET).
MVEL	time-averaged or mean velocity (vector; in m/s).
MVKSI	time-averaged or mean grid-oriented U-velocity (in m/s). Note: not applicable to unstructured meshes.
MVETA	time-averaged or mean grid-oriented V-velocity (in m/s). Note: not applicable to unstructured meshes.
MSAL	time-averaged or mean salinity (in ppt).
MTEMP	time-averaged or mean temperature (^{o}C).
MSED	time-averaged or mean suspended sediment (in kg/m^3).
ZK	layer interface (in m).
HK	layer thickness (in m).
VMAGK	velocity magnitude per layer (in m/s).
VDIRK	velocity direction per layer (in degrees; Cartesian or nautical).
VELK	flow velocity per layer (vector; in m/s).
VKSIK	grid-oriented U-velocity per layer (in m/s). Note: not applicable to unstructured meshes.
VETAK	grid-oriented V-velocity per layer (in m/s). Note: not applicable to unstructured meshes.
VZ	velocity in $z-$direction (in m/s).
VOMEGA	vertical velocity relative to sigma plane (in m/s).
SALK	salinity per layer (in ppt).
TEMPK	temperature per layer (^{o}C).
SEDK	suspended sediment per layer (in kg/m^3).
TKE	turbulent kinetic energy per layer interface (in m^2/s^2).
EPS	dissipation rate of turbulent kinetic energy per layer interface (in m^2/s^3).
VISC	vertical eddy viscosity per layer interface (in m^2/s).
QMAGK	magnitude of discharge per unit width per layer (in m^2/s).
QDIRK	direction of discharge per unit width per layer (in degrees; Cartesian or nautical).
DISCHK	discharge per unit width per layer (vector; in m^2/s).
QKSIK	grid-oriented U-discharge per unit width per layer (in m^2/s). Note: not applicable to unstructured meshes.
QETAK	grid-oriented V-discharge per unit width per layer (in m^2/s). Note: not applicable to unstructured meshes.
PRESSK	pressure per layer (in hPa).
NHPRSK	normalised (by density) non-hydrostatic pressure per layer (in m^2/s^2).
MVMAGK	time-averaged or mean velocity magnitude per layer (in m/s).
MVDIRK	time-averaged or mean velocity direction per layer (in degrees; Cartesian or nautical; see command SET).
MVELK	time-averaged or mean velocity per layer (vector; in m/s).

`MVKSIK` time-averaged or mean grid-oriented U-velocity per layer (in m/s).
Note: not applicable to unstructured meshes.

`MVETAK` time-averaged or mean grid-oriented V-velocity per layer (in m/s).
Note: not applicable to unstructured meshes.

`MSALK` time-averaged or mean salinity per layer (in ppt).

`MTEMPK` time-averaged or mean temperature per layer (^{o}C).

`MSEDK` time-averaged or mean suspended sediment per layer (in kg/m^3).

`MTKE` time-averaged or mean turbulent kinetic energy per layer interface (in m^2/s^2).

`MEPS` time-averaged or mean dissipation rate of turbulent kinetic energy per layer interface (in m^2/s^3).

`MVISC` time-averaged or mean vertical eddy viscosity per layer interface (in m^2/s).

`TIME` full date-time string as part of line used in `TABLE` only.

`TSEC` time in seconds with respect to a reference time (see command `QUANTITY`).

`XP` user instructs SWASH to write the $x-$coordinate in the problem coordinate system of the output location.

`YP` user instructs SWASH to write the $y-$coordinate in the problem coordinate system of the output location.

`DIST` if output has been requested along a curve (see command `CURVE`) then the distance along the curve can be obtained with the command `TABLE`. `DIST` is the distance along the curve measured from the first point on the curve to the output location on the curve in meters (also in the case of spherical coordinates).

`FORCEX` $x-$component of the force acting on floating body (surge; in kN)
Note: to be used in `TABLE` only with the empty set `NOGRID`.

`FORCEY` $y-$component of the force acting on floating body (sway; in kN).
Note: to be used in `TABLE` only with the empty set `NOGRID`.

`FORCEZ` $z-$component of the force acting on floating body (heave; in kN).
Note: to be used in `TABLE` only with the empty set `NOGRID`.

`MOMX` $x-$component of the moment acting on floating body (roll; in kNm).
Note: to be used in `TABLE` only with the empty set `NOGRID`.

`MOMY` $y-$component of the moment acting on floating body (pitch; in kNm).
Note: to be used in `TABLE` only with the empty set `NOGRID`.

`MOMZ` $z-$component of the moment acting on floating body (yaw; in kNm).
Note: to be used in `TABLE` only with the empty set `NOGRID`.

`TRAX` translational displacement of rigid body in $x-$direction (surge; in m)
Note: to be used in `TABLE` only with the empty set `NOGRID`.

`TRAY` translational displacement of rigid body in $y-$direction (sway; in m).
Note: to be used in `TABLE` only with the empty set `NOGRID`.

`TRAZ` translational displacement of rigid body in $z-$direction (heave; in m).
Note: to be used in `TABLE` only with the empty set `NOGRID`.

`ROTX` rotational displacement of rigid body around $x-$axis (roll; in degrees).
Note: to be used in `TABLE` only with the empty set `NOGRID`.

`ROTY` rotational displacement of rigid body around $y-$axis (pitch; in degrees).
Note: to be used in `TABLE` only with the empty set `NOGRID`.

`ROTZ` rotational displacement of rigid body around $z-$axis (yaw; in degrees).
Note: to be used in `TABLE` only with the empty set `NOGRID`.

`PTOP` power of PTO (power take-off; in kW).
Note: to be used in `TABLE` only with the empty set `NOGRID`.

`RUNUP` (vertical) runup height (in m).
Note: to be used in `TABLE` only with the empty set `NOGRID`.
NOT FOR 2D-MODE.

`[unit]` this controls the scaling of output. The program divides computed values by `[unit]` before writing to file, so the user should multiply the written value by `[unit]` to obtain the proper value.
Default: if `HEADER` is selected, value is written as a 5 position integer. SWASH takes `[unit]` such that the largest number occurring in the block can be printed.
If `NOHEADER` is selected, values are printed in floating-point format, `[unit]` = 1.

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

`[tbegblk]` begin time of the first field of the variable, the format is:

1	: ISO-notation	19870530.153000
2	: (as in HP compiler)	'30−May−87 15:30:00'
3	: (as in Lahey compiler)	05/30/87.15:30:00
4	:	15:30:00
5	:	87/05/30 15:30:00'
6	: as in WAM	8705301530
7	:	153000.000

This format is installation dependent. See Implementation Manual or ask the person who installed SWASH on your computer. Default is option 7.

`[deltblk]` time interval between fields, the unit is indicated in the next option:

SEC	unit seconds
MIN	unit minutes
HR	unit hours
DAY	unit days

Notes:

- The $x-$ and $y-$components of the vectorial quantities `VEL`, `DISCH`, `UFRIC`, etc. are always given with respect to the problem coordinate system.

- For direction in Cartesian convention: relative to $x-$axis of the problem coordinate system (counterclockwise). Possible exception: in the case of output with `BLOCK` command in combination with command `FRAME`, see command `QUANTITY`.

- For a proper use of the quantity `HRUN`, take into account the following notes:

1. Quantity HRUN indicates the *maximum* horizontal runup that occur during the whole simulation. So, it should be plotted at the final time step.
2. The correct values for HRUN are 0 and 1 only. To prevent interpolation, use the command GROUP.
3. The horizontal runup is associated with the inundation depth. The default value is the threshold water depth; see command SET. Otherwise it is determined by the user using the command QUANTITY HRUN depth.
4. To determine the (vertical) runup height with respect to the still water level, also plot the quantity BOTLEV. At those locations where HRUN is for the first time zero, the corresponding value of $-1\times$ BOTLEV indicates the runup height.
5. To find and plot the location of inundation use the following Matlab script:

```
BWs=edge(Hrunup,'sobel');
hrp=zeros(size(BWs,1),size(BWs,2));
hrp(BWs==1)=1;
hrp(hrp==0)=NaN;
x=Xp(find(hrp==1));
y=Yp(find(hrp==1));
```

   Note the quantities XP and YP must be included as well.

- Quantity BRKP indicates areas where wave breaking occur. The correct values for BRKP are 0 (no breaking) and 1 (breaking) only. To prevent interpolation, use the command GROUP. In this way, white spots can be visualised.

- The wave parameters SETUP, HSIG and HRMS are determined by averaging the surface elevation over the duration [dur] as specified in command QUANTITY. Hence, they are regarded as stationary quantities and must be plotted at the final time step only.

- The mean currents MVEL, MVELK, etc. are determined by averaging the velocities over the duration [dur] as specified in command QUANTITY. Hence, they are regarded as stationary quantities and must be plotted at the final time step only.

- The mean constituents MSAL, MTEMP and MSED are determined by averaging the constituents over the duration [dur] as specified in command QUANTITY. Hence, they are regarded as stationary quantities and must be plotted at the final time step only.

- The mean turbulence quantities MTKE, MEPS and MVISC are determined by averaging the turbulence quantities over the duration [dur] as specified in command QUANTITY. Hence, they are regarded as stationary quantities and must be plotted at the final time step only.

- The hydrodynamic forces that act on the floating body (e.g. a moored ship or pontoon), FORCEX, FORCEY and FORCEZ, are found by integrating the total pressure over the wet surface of the body. The moments around the center of gravity of the body, MOMX, MOMY and MOMZ, are computed as the integral of the product of the pressure and the moment arm, which is the distance to the center of the body (see command QUANTITY), over the wet surface of the body. These output quantities are thus space independent, can be specified with the command TABLE only, and the empty set NOGRID must be chosen.
- The runup height RUNUP is computed by the intersection between free surface and bottom level. Three assumptions have been made for the calculation of this runup height.

  1. A minimum water depth larger than zero is assumed. The default value is the threshold water depth; see command SET. Otherwise it is determined by the user using the command QUANTITY RUNUP delrp.
  2. The simulation is carried out in 1D-mode.
  3. The wave condition is imposed on the west boundary of the computational domain (either regular or irregular), so that the wave propagation is pointing eastward.

  This output quantity is space independent, can be specified with the command TABLE only, and the empty set NOGRID must be chosen.

```
                 | -> HEADer   |
                 |             |
TABle  'sname'  <     NOHEADer  >  'fname'                                  &
                 |             |
                 |    SWASH    |

              | ... |                                   | -> Sec   |
       <    <       >   > (OUTput [tbegtbl] [delttbl]   <     MIn   >)
              | ... |                                   |    HR    |
                                                        |    DAy   |
```

With this optional command the user indicates that for each location of the output location set 'sname' (see commands POINTS, CURVE, FRAME or GROUP) one or more variables should be written to a file. Some quantities are space independent and vary in time only. These are e.g. the wave runup height and the hydrodynamic loads (forces and moments acting on a floating object). For these quantities an empty output set with the name 'sname' = 'NOGRID' must be chosen.

The keywords `HEADER` and `NOHEADER` determine the appearance of the table; the filename determines the destination of the data.

`'sname'` name of the set of `POINTS`, `CURVE`, `FRAME`, `GROUP` or the empty set `NOGRID`.

`HEADer` output is written in fixed format to file with headers giving name of variable and unit per column. A disadvantage of this option is that the data are written in fixed format; numbers too large to be written will be shown as: `****`. Number of header lines is 4.

`NOHEADer` output is written in floating point format to file and has no headers; it is intended primarily for processing by other programs. With some spreadsheet programs, however, the `HEADER` option works better.

`SWASH` a table on file is produced with a special fixed format appropriate for layer-dependent quantities. This file contains headers with useful information for a correct interpretation of the data.

`'fname'` name of the data file where the output is to be written to.
Default for option HEADER is output to the `PRINT` file.
In case of `NOHEADER` the filename is required.

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

the output parameters as given in command `BLOCK`.

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

`[tbegtbl]` begin time of the first field of the variable, the format is:

1 :	ISO-notation	19870530.153000
2 :	(as in HP compiler)	'30−May−87 15:30:00'
3 :	(as in Lahey compiler)	05/30/87.15:30:00
4 :		15:30:00
5 :		87/05/30 15:30:00'
6 :	as in WAM	8705301530
7 :		153000.000

This format is installation dependent. See Implementation Manual or ask the person who installed SWASH on your computer. Default is option 7.

`[delttbl]` time interval between fields, the unit is indicated in the next option:

SEC	unit seconds
MIN	unit minutes
HR	unit hours
DAY	unit days

Notes:

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

- It is advised to use the command HEAD or SWASH in case of outputting layer-dependent quantities (e.g. ZK, VELK, PRESSK, etc.). With the command SWASH, the quantity TIME is automatically included.

4.6.3 Write or plot intermediate results

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

      (FILE ‘fname’)
```

If SWASH produces unexpected results, this optional command can be used to instruct the program to produce intermediate results during a SWASH run (test output). A TEST command may change between commands in the file to change the level of test output during a SWASH run. This change occurs during the execution of the run. A TEST command controls the test output until the next TEST command. Such a next TEST command may have level 0, thus stopping test output.

[itest] the level of test output. For values under 100 the amount is usually reasonable, for values above 200 it can be very large. For values of [itest] up to 50 the test output can be interpreted by the user. For higher values of [itest] the test output can only be interpreted by those who have the program source listing at their disposal.
For instance, with [itest] = 30, one may check mass and energy conservation, if appropriate.
Default: [itest] = 1

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

POINTS if this option is used, the user instructs SWASH to produce detailed print output during the computational process for a grid point of the computational grid. Output at a maximum of 50 grid points is possible. This option can be used only after the bathymetry has been read (see command READINP BOTTOM).

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

[i], [j] grid indices of a test point. Values of [i] range from 1 to [mxc]+1 (see command CGRID), values of [j] from 1 to [myc]+1.
ONLY MEANT FOR STRUCTURED GRIDS.

[k] vertex index of a test point. This can be obtained in a grid generator file

(`.node` and `.n` files of Triangle and Easymesh, respectively).
ONLY MEANT FOR UNSTRUCTURED GRIDS.

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

`[x], [y]` coordinates of a test point (problem coordinates in meters in case of Cartesian coordinates, or longitude and latitude in degrees in case of spherical coordinates, see command `COORD`).

`FILE` some flow variables for test points are written.

`‘fname’` name of the file to which the output is written; default filename: `DIAGNOSTIC`.

4.7 Lock-up

```
                              | -> Sec  |
COMPute  [tbegc] [deltc] <        MIn    > [tendc]
                              |     HR   |
                              |     DAy  |
```

This command orders SWASH to start the computation.

`[tbegc]` the start (date and) time of the computation, the format is

1	: ISO-notation	19870530.153000
2	: (as in HP compiler)	’30−May−87 15:30:00’
3	: (as in Lahey compiler)	05/30/87.15:30:00
4	:	15:30:00
5	:	87/05/30 15:30:00’
6	: as in WAM	8705301530
7	:	153000.000

This format is installation dependent. See Implementation Manual or ask the person who installed SWASH on your computer. Default is option 7.

`[deltc]` the time step of the computation, the unit is indicated in the next option:

SEC	unit seconds
MIN	unit minutes
HR	unit hours
DAY	unit days

`[tendc]` the end time of the computation, format see `[tbegc]`.

Note that several commands `COMPUTE` can appear, where the wave state at the end of one computation is used as initial state for the next one, unless a command `INIT` appears in between the two COMPUTE commands. This enables the user to change the computational time step during the whole computation, to change a boundary condition, to modify numerical parameters etc.

```
STOP
```

This required command marks the end of the commands in the command file. Note that the command STOP may be the last command in the input file; any information in the input file beyond this command is ignored.

Chapter 5

Setting up your own command file

In this chapter some guidelines for setting up a command file is given. This file contains all the necessary commands and data required for defining a model and running the simulation. All the commands and keywords are listed in Appendix C. Not all the commands need to be specified. Most of them will be taken default. Generally, a command file consists of five parts which must be specified by the user:

1. a computational grid,
2. some input grids, most notably the bathymetry,
3. initial and boundary conditions,
4. some physical and numerical parameters, and
5. output quantities, output locations and output formats.

5.1 Computational grid

A distinction is made between the definition of grids in horizontal and vertical directions. In the horizontal direction, either rectilinear, curvilinear or unstructured mesh can be employed. The grid definition in vertical direction is defined by means of a fixed number of layers in a such a way that both the bottom topography and the free surface can be accurately represented; see also Figure 5.1.

First we need to define the size and direction of the computational domain in the horizontal plane. The area of interest should be kept at least two wave lengths away from the boundaries. Note that if a sponge layer needs to be included, the computational domain needs to be extended with the size of that sponge layer. It is always wise to choose the grid axes being aligned as much as possible with the dominant wave direction.

An important aspect of specifying a computational grid is the spatial resolution. In principle, most energetic wave components need to be resolved accurately on the grid. Basically, we take sufficient number of grid points per wave length associated with the peak wave energy. For low waves, i.e. $H/d \ll 1$ with H a characteristic wave height (either significant or RMS) and d the (still) water depth, it is sufficient to take 50 grid cells (or 51 grid points) per peak wave length. However, for relatively high waves, it is better to take at least 100 grid cells per peak wave length. So, at least, we need to know something about a typical water depth and a typical peak period. Based on the linear dispersion relation the corresponding peak wave length can be found.

An example. We have a domain of 1500 m alongshore and 1200 m cross-shore of which the deepest part is 20 m. We impose a wave spectrum at the entrance with a significant wave height of 2 m and a peak period of 10 s. According to the linear dispersion relation the peak wave length is about 120 m and $H_s/d = 0.1$. Hence, the waves are not too low but also not too high either. So, for safety, we choose 100 grid cells per peak wave length which implies a grid size of 1.2 m. Since the dominant wave direction is cross-shore, we may relax the grid size alongshore by choosing 3 m. So, the total number of grid cells for the computational domain would be 500 × 1000.

Keep in mind, however, that waves become shorter when the water depth decreases. Hence, for a desired number of grid cells per wave length the grid resolution must therefore be higher locally. In this respect, a rectangular, non-uniform grid is recommended. The desired number of grid cells for this case may be lesser than the one associated with the peak wave length. A rule of thumb is 15 to 20 cells.

However, to further enhance the flexibility with respect to local grid refinements at places where needed, either a boundary-fitted, orthogonal curvilinear grid or an unstructured triangular mesh may be applied. Such meshes must be generated externally. The user is, however, strongly advised to define the computational domain as a rectangle with one of the axis being aligned with the dominant wave direction.

Another issue is the choice of the number of layers. This choice mainly depends on two types of application:

- vertical flow structures, and
- wave transformation.

since flows and waves have drastically different physical features. If one is interested in the vertical flow structures, e.g. undertow and stratified flows, then at least 10 or perhaps more vertical layers should be adopted. If necessary, a non-equidistant layer distribution can be specified; see Figure 5.1. The layer thickness h_k, which is the distance between two consecutive layer interfaces, may be defined in a relative way, i.e. a percentage of the water depth similar to the $\sigma-$coordinates, or in an absolute way, i.e. a constant or fixed layer thickness as expressed in meters. To make sure that the sum of the layer thicknesses equals the water depth, at least one layer must be defined in a relative way. The choice for fixed layers may be useful in the case of a bathymetry exhibiting strong variation of the depth, such as a lake with some pits, in order to keep the vertical resolution relatively high along the bottom.

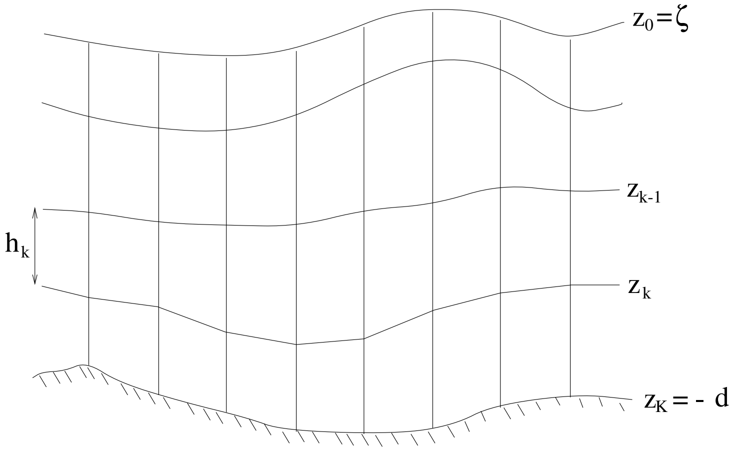

Figure 5.1: Vertical grid definition with K layers and K+1 layer interfaces.

Concerning the wave transformation, the number of layers is determined by the linear frequency dispersion. In particular, the dimensionless depth, kd with k the wave number, decides the number of layers. The higher the value of kd, the more vertical layers needed. In addition, the accuracy with which the phase velocity of the wave components, $c = \omega/k$ with ω the angular frequency, is obtained depends on the discretization of the vertical pressure gradient in the momentum equations. For wave transformation usually the Keller-box scheme is adopted; see Section 5.4.3.

The range of applicability of the SWASH model to values of kd indicating the relative importance of linear wave dispersion for *primary waves* is given in Table 5.1. This range is determined by requiring a relative error in the normalized wave celerity ($= c/\sqrt{gd}$) of

Table 5.1: Range of dimensionless depth as function of number of layers K in SWASH.

K	range	error
1	$kd \leq 0.5$	1%
1	$kd \leq 2.9$	3%
2	$kd \leq 7.7$	1%
3	$kd \leq 16.4$	1%

at most 1%. An exception is the use of one vertical layer where the relative error is 3%, which is acceptable for many applications. Here, at most three layers may be considered enough for typical wave simulations. Moreover, the layers are assumed to have variable thicknesses and be equally distributed, which is the usual choice for wave simulations. So, do not use fixed layers!

It is noted that SWASH uses its own dispersion relation, which is an approximate one of the exact linear dispersion relation, given by

$$\omega^2 = gk\,\tanh(kd)$$

This approximate relation is derived using the Keller-box scheme (see Section 5.4.3) and depends on the number of equidistant layers employed in the model. The linear dispersion relation of SWASH using one vertical layer, i.e. depth-averaged, is given by

$$\omega^2 = gk\,\frac{kd}{1+\frac{1}{4}k^2d^2}$$

while for two and three equidistant layers, it is given by

$$\omega^2 = gk\,\frac{kd+\frac{1}{16}k^3d^3}{1+\frac{3}{8}k^2d^2+\frac{1}{256}k^4d^4}$$

and

$$\omega^2 = gk\,\frac{kd+\frac{5}{54}k^3d^3+\frac{1}{1296}k^5d^5}{1+\frac{5}{12}k^2d^2+\frac{5}{432}k^4d^4+\frac{1}{46656}k^6d^6}$$

respectively. Thus the approximate dispersion relation is consistent with the model, particularly for relatively high wave frequencies. This will lead to more accurate results. The approximate dispersion relation in SWASH is only available for one, two and three equidistant layers with variable thickness (i.e. sigma planes, not fixed layers). SWASH shall indicate this in the PRINT file.

So, for primary waves with $kd \leq 2.9$, use of one layer is sufficient, while at least two equidistant layers need to be chosen if $kd \leq 7.7$. For most typical nearshore wave simulations, two layers may be enough. In the example above, $kd = 1.04$, so one vertical layer would be enough.

However, one must realized that, for a given number of layers, relatively high harmonics may propagate too slow at a given depth. For example, using one layer, SWASH is accurate up to a kd value of 2.9 for primary waves. For a depth of 20 m (deepest part in the example above), the shortest wave that can accurately modelled at this depth has a frequency of 0.18 Hz (minimum wave period of 5.6 s; derived from the approximate dispersion relation above). In other words, for a given number of layers and a water depth, there is a maximum frequency above which a wave component has an incorrect celerity;

Table 5.2: Maximum frequency (in Hz) as function of still water depth (in m) and number of layers.

d (m)	K=1	K=2	K=3
1	0.82	1.37	2.00
5	0.37	0.61	0.89
10	0.26	0.43	0.63
15	0.21	0.35	0.52
20	0.18	0.31	0.45
25	0.16	0.27	0.40
30	0.15	0.25	0.36
35	0.14	0.23	0.34
40	0.13	0.22	0.32
45	0.12	0.20	0.30
50	0.12	0.19	0.28
100	0.08	0.14	0.20

see Table 5.2. This means, for instance, that the phase differences between the concerning harmonics are wrong. This is particularly important when nonlinear effects are dominant[1].

Ideally, the maximum frequency is about 1.5 to 2 times the peak frequency at a given depth. It is then assumed that all components above this maximum frequency have a little bit amount of energy (here the presuming spectrum shape is a Jonswap one). As a consequence, the phase differences between the *representative* wave components, including the relatively short waves, are thus well controlled in the model.

Referred to the example above, the peak frequency at the wavemaker boundary is 0.1 Hz, while the depth is 20 m. The required maximum frequency should be at least 0.15 Hz or preferably higher. So, the use of one layer would be critical when propagation of short waves with frequencies higher than 0.18 Hz needs to be modelled accurately.

5.2 Input grids

The most important input grid is the bathymetric grid. This grid represents the bottom level at each grid point with land points defined as negative while wet points are defined as positive. The resolution of bathymetric grid is not necessarily the same as that of the computational grid. It is advised to avoid extremely steep bottom slopes or sharp obstacles

[1] If one is interested in propagation of very short waves in the surf zone ($d \sim 0.5$ m) with $f < 4$ Hz, while these waves exchange energy through nonlinear interactions, the user is advised to choose six equidistant layers. By using these number of layers, the maximum relative error for $kd < 40$ remains below 0.1%.

as much as possible. Some kind of smoothing or re-interpolation is therefore recommended.

Jetties, piers and quays, or other impermeable walls, may be schematized either

- by including their slopes in the bathymetry, or
- by means of porosity layers with a small porosity ($n < 0.1$), or
- a combination of the first two options.

From a stability point of view, the second option is the best one. However, the first option may be a better choice when, for instance, wave diffraction around the berm of the quay need to be simulated accurately. To avoid unrealistically high surface elevation around the quays or possible instabilities due to steep slopes, this first option may be combine with a larger threshold of the water depth; see Section 5.4.6 and command `SET DEPMIN`. The default value of this threshold is 0.05 mm. Depending on the horizontal grid sizes (Δx, Δy), and thereby the actual slope, this threshold may be altered in order to get a stable solution. The larger the grid sizes, the higher this threshold should be set (e.g. 0.1 mm or even 1 mm). But be careful as this higher threshold may negatively influence mass conservation.

Instead, however, a combination is also possible, i.e. to place the porosity layer on top of the quay walls, and having a volumetric porosity of 20% and a grain size of 0.1 m.

A rubble mound breakwater must be schematized by means of porosity layers. These layers must be placed inside the computational domain. Rubble mound breakwaters have a typical porosity value of (n=) 0.4, while the stone size of the armour layer is typically 0.5 m. The berm of the breakwater can be specified by means of the structure heights (relative to the bottom). In case of two or more breakwaters in the domain, both porosity and structure height are thus spatially varied, and so they need to be inputted by means of input grids. Also stone diameters need to be specified as well.

Alternatively, the porosity layers may be placed on top of the impermeable core of the breakwater which, in turn, is schematized by adapting the bottom level. In addition, the berm of the breakwater is schematized by including its slope in the adapted bathymetry.

The width of the breakwater should be at least four times the grid size of the computational grid. So, the grid resolution should be high enough. When choosing a too coarse grid size, it may lead to an overestimation of the transmission and an underestimation of the reflection.

This way of schematization permits to simulate partial reflection and transmission of the waves through breakwaters. Wave reflection at a breakwater is typically determined by wave energy dissipation on the slope and wave penetration into the breakwater. Both processes are equally important, and thus both slope angle and porosity are important governing parameters for the wave reflection.

Using the command `INPGRID BOTTOM EXCEPTION`, one can introduce permanently dry points in the computational grid. This provides a means to make a line of dams or screens through the computational domain, separating the flow on both sides. This line of thin dams may represent a small obstacle with subgrid dimensions that possibly influence the local flow. It must be noted that for parallel runs using MPI the user must indicate an exception value for bottom levels, if appropriate, in order to obtain good load balancing.

The water depth should be uniform along the wavemaker boundary where incident waves are imposed.

If tidal currents are significant over the computational domain, the spatial distribution of the currents $u(x, y)$ should be specified as an input grid.

5.3 Initial and boundary conditions

To solve the continuity and momentum equations, appropriate boundary conditions need to be imposed at the boundaries of the computational grid.

In general, initial conditions are more important for relative short simulations (e.g. a few minutes in case of short waves or a few days in case of tidal waves). Boundary conditions are by far more important for longer simulations. Moreover, often no information is available in order to start the simulation. Therefore, the simulation will usually start with zero velocities and a spatially constant water level, and the simulation will be long enough to get a steady-state solution; see also Section 5.4.1.

Waves may be generated along one or two boundaries. These are called wavemaker boundaries. First, it is assumed that the boundaries are not curved. Thus, the use of curvilinear grids is restricted to *rectangular* domains with non-uniform grids. Second, it is assumed that the variation of the depth along these wavemaker boundaries is slowly or (preferably) constant. Third, it is advised to place these wavemaker boundaries away from the area of interest, and away from steep topography. At the wavemaker boundary, we may imposed either regular or irregular waves. For one-dimensional cases they are by definition long-crested or uni-directional. For a two-dimensional case short-crested or multi-directional waves can also be specified. Usually, a time series need to be given for incident waves. This may either be synthesized from parametric information (wave height, period, etc.) or derived from a surface elevation time series.

It is important to note that at a wavemaker boundary only the horizontal velocities (u and v) are prescribed whereas a homogeneous Neumann condition is applied to the vertical velocity w. This, however, may induce an error in the generating waves near the boundary, especially when the waves are high and the advection terms of the $w-$momentum equation are involved. Therefore, it is advised not to include (both horizontal and vertical) advection terms in the $w-$momentum equation. (The balance between the local acceleration term $\partial w/\partial t$ and the vertical pressure gradient $\partial q/\partial z$ usually suffices.) See Section 5.4.5 for further details.

For regular waves, at least the wave height and the wave period must be specified. Optionally, the wave direction can be specified as well. Alternatively, a time series can be imposed.

For irregular waves, either a spectrum or time series can be enforced. In the case of a spectrum, both the shape and wave characteristics need to be specified. The usual shape is either Jonswap or Pierson-Moskowitz. Sometimes a TMA shape is desired. The wave characteristics are determined by the following parameters: the significant or RMS wave height, peak or first order mean period, peak wave direction, and directional spreading (only in case of short-crested waves). The frequency range $[f_{\rm min}, f_{\rm max}]$ is such that the highest frequency, $f_{\rm max}$, equals 3 times the peak frequency (or mean frequency), while the lowest one, $f_{\rm min}$, equals half of the peak/mean frequency.

Alternatively, a spectrum file may be given. There are two types of files:

- A file containing 1D or non-directional wave spectrum (usually from measurements).
- A file containing 2D or directional wave spectrum (possibly from another SWAN run).

Using a spectrum a time series of surface elevation will be synthesized. At least, the length of this series should correspond to the time period over which surface elevation and velocities are outputted after steady-state condition has been established. This time period should be long enough to provide statistically reliable wave data. After this time period the time series repeats itself. This duration of the time series is called the cycle period (see command `BOUnd ... SPECTrum ... [cycle]`). The recommended range is from 100 to 300 wave periods. If the cycle period is denoted as $T_{\rm cycle}$, then the frequency step Δf to be used for the evaluation of the parametric spectrum (e.g. Jonswap) equals

$$\Delta f = \frac{1}{T_{\rm cycle}}$$

Thus the spectrum is divided into N frequency bins with uniform spacing Δf,

$$N = \frac{f_{\rm max} - f_{\rm min}}{\Delta f}$$

Referring to the above example, we impose a Jonswap spectrum at the wavemaker boundary. The peak period is 10 s, so that the frequencies are in between 0.05 Hz and 0.3 Hz.

The duration of the time series of surface elevation to be synthesized is set to 30 minutes, which is supposed to be accurate enough to get sufficient statistics like wave height and mean period. Hence, in total, 450 wave components will be generated at the entrance of the computational domain.

Please be careful in choosing the cycle period. The larger this period the more wave components will be involved at the wavemaker boundary. Based on these components, SWASH will synthesize time series for the orbital velocities in each grid point and each vertical layer along the boundary. That would enhance the computing time significantly.

When imposing a spectrum at the boundary, one has to realize that some so-called evanescent modes might be included as well. These modes show exponential decay with distance from the boundary at which the spectrum is imposed. As such, they can not be "seen" by the model. Evanescent waves are a general property of the underlying model equations. The frequency at which the evanescent modes are generated is the cut-off frequency and is determined by the dispersive properties of the model equations. It is given by

$$\omega_{\mathrm{cf}} = 2\,K\sqrt{\frac{g}{d}}$$

with K the number of layers used in the model. Hence, the lowest wave period to be considered in the model simulation equals $2\pi/\omega_{\mathrm{cf}}$. So, given the depth at the boundary and the number of layers used, the cut-off frequency is determined above which the evanescent waves are generated at the wavemaker boundary where the spectrum is imposed; see Table 5.3.

Table 5.3: Cut-off frequency (in Hz) as function of still water depth (in m) and number of layers.

d (m)	K=1	K=2	K=3
1	1.00	1.99	2.99
5	0.45	0.89	1.34
10	0.32	0.63	0.95
15	0.26	0.51	0.77
20	0.22	0.45	0.67
25	0.20	0.40	0.60
30	0.18	0.36	0.55
35	0.17	0.34	0.51
40	0.16	0.32	0.47
45	0.15	0.30	0.45
50	0.14	0.28	0.42
100	0.10	0.20	0.30

These evanescent modes will be removed by SWASH. Note that these modes carry a little bit energy and thus negligible. SWASH will give a warning when at least 10% of the total wave components are the evanescent modes that have been removed. If there are too much evanescent modes on the boundary, i.e. these modes together contain a significant amount of energy of the wave spectrum, the user is advised either to enlarge the number of layers (see also Table 5.2) or to truncate the imposed spectrum (e.g. SWAN spectrum), i.e. the highest frequency of the spectrum is not larger than the given cut-off frequency.

In the above example, one layer ($K = 1$) has been chosen. We assume that along the wavemaker boundary we have a uniform depth of 20 m. So the cut-off frequency is 0.22 Hz (see Table 5.3; the lowest wave period is thus 4.5 s). However, the highest frequency is 0.3 Hz. So there are 144 evanescent modes on the boundary, which is about 30%, thus reasonably. They will be removed from the boundary.

For high waves, sub- and super-harmonics are generated due to nonlinearity. These waves are called bound waves as they are attached to the primary wave and travel at its phase speed instead of that of a free wave at the same frequency. If linear wave conditions are enforced at the boundaries, the model will generate spurious free wave components with the same magnitude but 180^o out of phase with the bound waves at the wavemaker in order to satisfy the linear wave boundary condition. The presence of spurious waves that travel at different speeds will lead to a spatially nonhomogeneous wave field with the wave height changing continuously over the domain.

For this unwanted situation, it is recommended to add second order bound waves at the wavemaker boundary so to avoid the presence of spurious waves. When a monochromatic wave is to be imposed at the open boundary, then bound super-harmonic wave components derived from the second order Stokes pertubation expansions (Vasarmidis et al., 2024) can be added, while for a bichromatic wave and a spectral wave also the bound sum- and difference-frequency components are included at the wavemaker boundary. This can be achieved by the command `ADDBoundwave`.

Like the dispersion relation, the solution to the second order Stokes wave theory is derived by means of the Keller-box scheme (see Section 5.4.3) and also depends on the number of equidistant layers employed in the model. It is limited up to four layers. For details, we refer to Vasarmidis et al. (2024).

The second order Stokes corrections are derived with the assumption of weak nonlinearity. This implies that the proposed boundary condition cannot be used in the surf zone ($a/d \sim 1$) and in deep water ($kd > 1$). This is the case when the Ursell number $a/d\,/\,(kd)^3$ exceeds 0.2. In that case, SWASH will give a warning. Nevertheless, for most practical applications, the boundary will be located in intermediate water depths where these limitations are not met. Furthermore, in deep water the second order response is small and can – to a good approximation – be neglected. In such case, a boundary condition based on linear wave theory is likely sufficient.

To simulate entering waves without some reflections at the wavemaker boundary, a weakly reflective condition allowing outgoing waves must be adopted (command `BTYPE WEAK`). This type of radiation condition has been shown to lead to good results within the surf zone.

Waves propagating out of the computational domain are absorbed by means of a sponge layer placed behind an output boundary. It is recommended to take the width of the sponge layer at least 3 times the typical wave length. However, for long waves a Sommerfeld radiation condition might be a good alternative.

When no boundary conditions are specified at a boundary, this boundary is considered to be a closed one. This boundary is fully reflective. Alternatively, periodic boundary conditions can be applied at two opposite boundaries. This means that wave energy leaving at one end of the domain enters at the other side. In this case no reflections at these boundaries occur. This is recommended in the case of a simulation of a field case where longshore bottom variations are negligible. In such a case the computational domain is made repeated in a representative direction (see command `CGRID ... REPeating X|Y`).

5.4 Numerical parameters

5.4.1 Duration of simulation

It is recommended to take into account both the spin up time and the duration of the time series at the wavemaker boundary. In this way, a steady-state condition will be obtained. We assume that the spin up time takes at most 10 to 15% of the total time of the simulation. For a suitable simulation time at least 500 to 1000 waves are needed.

In the above example the cycle period equals 30 minutes, which is supposed to be at least 85% of the total simulation time. So, the duration of the intended simulation would be 35 minutes, or more safely, 40 minutes.

5.4.2 Time step

The time integration is of explicit type and thus requires strict confirmity of stability criteria for a stable solution. The well-known CFL condition for 1D problems is given by

$$\mathrm{Cr} = \frac{\Delta t \left(\sqrt{gd} + |u| \right)}{\Delta x} \leq 1 \tag{5.1}$$

with Δx the mesh width, Δt the time step, u the flow velocity, and Cr the Courant number. For a 2D problem, however, the following CFL condition is employed

$$\mathrm{Cr} = \Delta t \left(\sqrt{gd} + \sqrt{u^2 + v^2} \right) \sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}} \leq 1$$

A dynamically adjusted time step controlled by the Courant number in a user prescribed range is implemented in SWASH as follows. The actual maximum of the Courant number over all wet grid points is determined. The time step is halved when this number becomes larger than a preset constant $\mathrm{Cr}_{\max} < 1$, and the time step is doubled when this number is smaller than another constant $\mathrm{Cr}_{\min}$, which is small enough to be sure the time step can be doubled. Usually, $\mathrm{Cr}_{\min}$ is set to 0.2, while the maximum Courant number $\mathrm{Cr}_{\max}$ is specified in the range of 0.5 to 0.8. It is advised not to choose a value higher than 0.8 since nonlinear processes, e.g. wave breaking and wave-wave interactions, can affect the stability condition. For high, nonlinear waves, or wave interaction with structures with steep slopes (e.g. jetties, quays), a Courant number of 0.5 is advised.

5.4.3 Vertical pressure gradient

Spatial discretization of the governing equations is carried out in a finite volume/finite difference fashion. A staggered grid arrangement is used in which the velocity components are located at the center of the cell faces (see Figure 5.2). The water level is located at cell center. Concerning the non-hydrostatic pressure, two layouts to assign this unknown to grid points are employed. This variable can be given either at the cell center or at the layer interface. The former is called the standard layout, while the latter one is called the box layout; see Figure 5.2.

The choice depends on the discretization of the vertical pressure gradient, namely, explicit central differences referring as the classical case and the implicit Keller-box or compact scheme, respectively. This compact scheme allows straightforward implementation of the zero pressure boundary condition at the free surface without the need for special attention at interior points near that surface. Moreover, the discretization error is four to six times smaller than the error of classical central differences of the same order and involving the same number of vertical grid points. Hence, use of the compact scheme allows a very few number of vertical grid points with relative low numerical dispersion and dissipation, thereby enhancing the accuracy of the frequency dispersion for relative short waves up to an acceptable level, see Table 5.1.

At very low vertical resolution (one or two layers), the Keller-box scheme gives good dispersive properties. At high vertical resolutions, however, the standard layout is preferable because it appears to be more robust while its dispersion characteristics are then usually sufficiently accurate.

To summarize, for wave simulations with 5 layers or less, the Keller-box scheme using the box layout is recommended, while for simulations with typically 10$-$20 layers, the

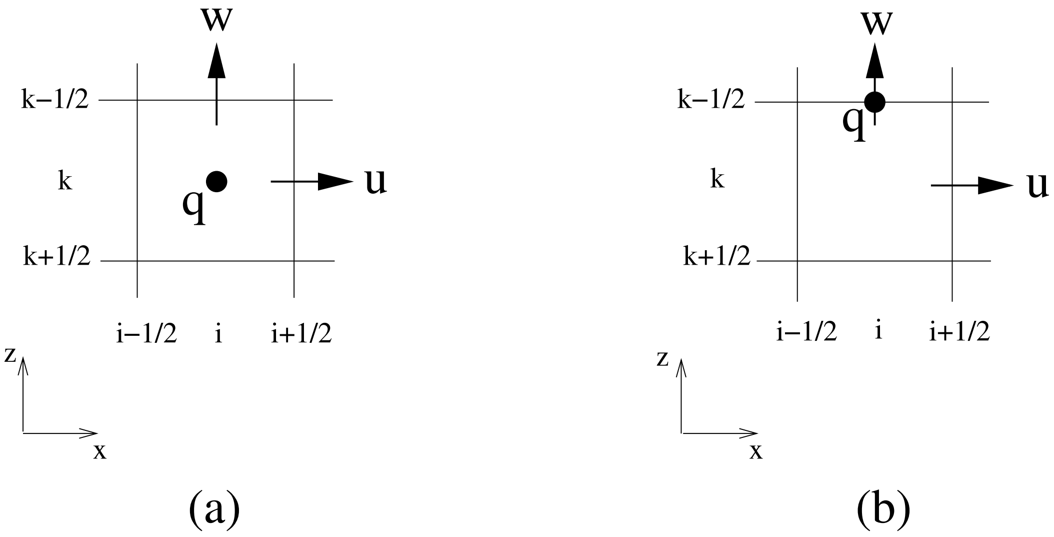

Figure 5.2: Applied arrangements of the unknowns in a staggered grid: (a) standard layout and (b) box layout. Meaning of the unknowns: u is the horizontal velocity, w is the vertical velocity and q is the non-hydrostatic pressure.

classical central differencing employing the standard layout is preferred. See command `NONHYDrostatic STANdard|BOX`.

Related to this choice, it might be useful to specify the preconditioner for solving the Poisson pressure equation. Two options are available: ILU and ILUD. For a robust solution, the ILU preconditioner is preferred. This choice might be a good one for applications where high and short waves are involved, or irregular beds with steep slopes (e.g. weir, breakwater, quay, jetty), or when relatively large number of layers (> 30) are involved. On the other hand, the ILUD preconditioner is a better choice to get an efficient solution (e.g. parallel computing). See command `NONHYDrostatic ... PREConditioner ILUD|ILU`.

5.4.4 Momentum conservation

For simulation of breaking waves, hydraulic jumps and bores, momentum must be conserved. Preference should then be given to a numerical method that conserves momentum, which ensures that the wave properties under breaking waves are modelled correctly. See command `DISCRET UPW MOM`.

5.4.5 Discretization of advection terms in the momentum equations

We make a distinction between horizontal and vertical advection terms of the momentum equations. Moreover, we consider the momentum equations separately, i.e. the $u-$momentum equation and the $w-$momentum equation. Note that the $v-$momentum equation will be treated as the $u-$momentum equation in exactly the same way. So, we have four different commands:

- `DISCRET UPW UMOM ...  H`
- `DISCRET UPW WMOM H`
- `DISCRET UPW UMOM ...  V`
- `DISCRET UPW WMOM V`

Below, they will be outlined, respectively.

Horizontal advection terms of $u-$momentum equation

We consider terms like

$$u\frac{\partial u}{\partial x} \quad \text{and} \quad v\frac{\partial u}{\partial y}$$

There are many schemes to approximate these terms. Some of these schemes are accurate but are prone to generate wiggles − typically space-centred schemes. Other schemes generate a certain amount of numerical diffusion and thus may affect the wave amplitude or wave energy of particularly short waves − typically upwind schemes. Higher order upwind schemes still generate small wiggles. If this is not desired, a flux-limiting scheme may be employed instead. Upwind schemes are known to be more stable than space-centred approximations.

The default scheme for the considered terms is the well-known second order BDF scheme (or sometimes called the LUDS scheme). For many applications this is a good choice. However, in some cases central differences (CDS) are preferred. This is especially the case when the higher harmonics are involved or when wave breaking is present (the amount of dissipation of higher harmonics is then important). Note that when the command `BREAK` is employed, SWASH will automatically apply central differences to the horizontal advection terms. If, for some reason, SWASH becomes unstable, possibly due to the growth of wiggles, the user is then advised to use the BDF scheme.

Other higher upwind schemes (e.g. QUICK) may be used as well, but we did not experience much differences compared to the BDF scheme. In any case, never apply the first order upwind scheme to any horizontal advection term, which is usually too numerically diffusive.

Horizontal advection terms of $w-$momentum equation

The horizontal advection terms of the $w-$momentum equation are given by

$$u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y}$$

These terms are usually ignored. For some applications they are negligible small compared to the vertical pressure gradient. However, they will be automatically taken into account in the simulation for

- applications where vertical flow structures are present (the standard layout for non-hydrostatic pressure is employed), or
- applications when wave breaking is important (command `BREAK` is employed).

If they are included, then the second order BDF scheme will be employed. Sometimes, central differences are preferred, for instance, when the higher harmonics are involved.

Vertical advection term of $u-$momentum equation

The vertical advection term of the $u-$momentum equation reads

$$w\frac{\partial u}{\partial z}$$

This term is only included in the computation when more than one layer is chosen ($K > 1$). The default scheme for this advection term is the MUSCL scheme. However, when many layers are involved or higher harmonics are present, then central differences might be a better choice.

Vertical advection term of $w-$momentum equation

This term is given by

$$w\frac{\partial w}{\partial z}$$

and is usually ignored even when $K > 1$ (for an explanation, see Section 5.3). It will be included automatically if the vertical flow structure is present (the standard layout for non-hydrostatic pressure is employed). This term is by default approximated with the MUSCL scheme.

5.4.6 Moving shorelines

For the calculation of wave runup and rundown on the beach, use of a moving boundary condition is required. The method used in SWASH to track the moving shoreline amounts to ensure non-negative water depths. For a one-dimensional case, one can show that if

$$\frac{|u|\Delta t}{\Delta x} \leq 1$$

and if a first order upwind scheme is applied to the global continuity equation, we shall have non-negative water depths at every time step; see [2] for a proof. Hence, flooding never happens faster than one grid size per time step, which is physically correct. This implies that the calculation of the dry areas does not need any special feature. For this reason, no complicated drying and flooding procedures are required. Additionally, the shoreline motion in the swash zone can be simulated in a natural manner.

For computational efficiency, the model equations are not solved and the velocities are set to zero when the water depth is below a threshold value (see command `SET DEPMIN`). Its default value is 0.05 mm. However, a higher threshold value may be chosen for scaling reasons. For instance, at the scale of a field site, a value of 1 mm is an appropriate choice. (As a matter of fact, the value of 0.05 mm is a suitable one under laboratory conditions.) This will also relax the time step to some extent in case of explicit time stepping. For a large-scale ocean simulation, a threshold value of 1 cm is probably more effective than 0.05 mm. Be careful when choosing a too high value as this may negatively influence mass conservation.

To achieve second order accuracy, the so-called MUSCL limiter may be employed (see command `DISCRET CORRDEP`).

Since the CFL condition, Eq. (5.1), holds this implies that ensuring non-negative water depths does not lead to a new time step restriction.

For some two-dimensional cases, however, ensuring non-negative water depths might lead to a time step restriction which appears to be more restrictive than the usual CFL condition. An example is the case where locally all velocities are directed outward of a grid cell. Nevertheless, such a case is rarely encountered and usually the time step is restricted by the Courant number based on the stability criterion.

5.5 Physical parameters

5.5.1 Depth-induced wave breaking

Neither Boussinesq-type wave models nor non-hydrostatic wave-flow models can be directly applied to details of breaking waves, since in both models essential processes such as overturning, air-entrainment and wave generated turbulence, are absent. But, if only the macro-scale effects of wave breaking are of interest, such as the effect on the statistics of wave heights, details of the breaking process can be ignored. By observing that both spilling and plunging breakers eventually evolve into a quasi-steady bore, where the entire front-face of the wave is turbulent, a breaking wave becomes analogous to a hydraulic jump. Consequently, its integral properties (rate of energy dissipation, jump height) are approximately captured by regarding the breaking wave as a discontinuity in the flow variables (free surface, velocities). Proper treatment of such a discontinuity in a non-hydrostatic model (conservation of mass and momentum) can therefore be used to determine the energy dissipation of waves in the surf zone; see Section 5.4.4.

Though a vertical coarse resolution (1−3 layers) is sufficient to describe the wave physics outside the surf zone (e.g. refraction, shoaling, diffraction, nonlinear wave-wave interactions), dissipation due to wave breaking requires a disproportional high vertical resolution (∼10−20). A coarse resolution will result in an underestimation of the horizontal velocities near the wave crest, and thus an underestimation of the amplitude dispersion. This underestimation implies that at low vertical resolution the influence of the non-hydrostatic pressure gradient is overestimated. Consequently, the stabilizing dispersive effects (i.e. the non-hydrostatic pressures) postpone the transition into the characteristic saw-tooth shape and therefore also the onset of dissipation.

By enforcing a hydrostatic pressure distribution at the front of a wave, we can locally reduce a non-hydrostatic wave-flow model to the nonlinear shallow water equations. The wave then rapidly transitions into the characteristic saw-tooth shape and, consistent with the high resolution approach, dissipation is captured by ensuring momentum conservation over the resulting discontinuity.

The subsequent dissipation is well described by assuming depth uniform velocities and a hydrostatic pressure distribution. In fact, these assumptions often form the basis to derive dissipation formulations to account for depth-induced breaking in energy balance type models, e.g. Battjes and Janssen (1978) among many others. Hence, prescribing a hydrostatic pressure distribution in the model around the discontinuity should result in the correct bulk dissipation.

There is no need to assume a hydrostatic pressure distribution if the vertical resolution is sufficient (i.e. 10 to 20 layers). However, imposing a hydrostatic distribution resolutions at low resolutions (1−3 layers) will ensure that, due the absence of dispersive effects, the front quickly transitions into a bore like shape. Hence, it can be used to initiate the onset of wave breaking, thus allowing for the use of low-vertical resolutions throughout the domain. In practice this means that once a grid point is in the front of a breaking wave, vertical accelerations are no longer resolved, and the non-hydrostatic pressure is set to zero.

A grid point is therefore labelled for hydrostatic computation if the local surface steepness $\partial\zeta/\partial x$ exceeds a predetermined value α. Equivalenty, $\partial\zeta/\partial t > \alpha\sqrt{gd}$. Once labelled, a point only becomes non-hydrostatic again if the crest of the wave has passed. This is assumed to occur when $\partial\zeta/\partial t < 0$. Furthermore, because grid points only become active again when the crest passes (where $w \approx 0$), vertical velocities w are set to zero on the front. To represent persistence of wave breaking, we locally reduce the criterion α to β if a neighbouring grid point (in $x-$ or $y-$direction) has been labelled for hydrostatic computation. In this case a point is thus also labelled for hydrostatic computation if $\partial\zeta/\partial t > \beta\sqrt{gd}$, with $\beta < \alpha$. In all other grid points, the computations are non-hydrostatic. Based on calibration, the default value for the maximum steepness parameter α is 0.6, while the persistence parameter β is set to 0.3.

To summarize, in case of a few layers (1−3) we must apply the command `BREAK` with optionally different values for α and β. In case of a sufficient number of layers (>10) nothing needs to be specified with respect to wave breaking.

5.5.2 Subgrid turbulent mixing

In case of the lateral mixing of momentum, e.g. around the tips of breakwaters and dams, it is recommended to employ the well-known Smagorinsky subgrid model in which the mixing length is assumed to be proportional to the typical grid spacing. The default value for the Smagorinsky constant is 0.2.

5.5.3 Vertical turbulent mixing

If the user is interested in the vertical flow structure, it is advised to apply the standard $k-\varepsilon$ turbulence model in order to take into account the vertical mixing. For stability reason, a background viscosity of 0.0001 m^2/s is recommended (see command `SET [backvisc]`).

5.5.4 Bottom friction

When waves are travelling over a relatively long distance of order of several kilometres, the influence of bottom friction becomes more pronounced. Moreover, it may affect long waves close to the shoreline, e.g. infragravity waves, and nearshore circulations. For wave simulations, a Manning coefficient of 0.019 is recommended.

5.6 Output quantities, locations and formats

SWASH calculates the time-dependent evolution of the surface elevation, velocities (both horizontal and vertical), pressure and possibly some turbulence quantities. It would require an excessive amount of disk space to the store these quantities at every grid point for every time step. Therefore, the user has to make some decisions on what need to be outputted by SWASH. One can make some tables of time series of surface elevation, velocities, discharges, pressures, turbulent kinetic energy, etc. or time-averaged velocities and turbulence quantities or wave height and wave-induced setup at specified location points. Alternatively, one may also output several quantities over the entire domain or a part of the domain at certain times (”snapshots”). They are stored as blocks in the Matlab binary files. Note that the corresponding file size is limited to 8 GB.

The user must determine the time duration over which the wave parameters, e.g. wave height and setup, mean current or mean turbulence quantities are computed; see command

QUANT. This corresponds to the final stage of the simulation period, which should be long enough to establish steady-state conditions. This time duration, which should be long enough to provide statistically reliable data, equals the duration of the simulation minus the spin up time of the simulation; see Section 5.4.1.

5.7 The importance of parallel computing

Suppose one wants to simulate a harbour with a typical domain size of 2 × 2 km^2 with SWASH. In addition, we assume the following typical values:

- an offshore water depth of 20 m,
- an offshore wave height of 1.5 m,
- a wave peak period of 8 s, and
- a simulation period of 60 minutes to get reliable wave statistics.

According to the dispersion relation, $kd = 1.4$ or the wave length of the primary wave is about 90 m. We choose a grid size of 2 m, being 1/45 of the wave length. Requiring a wave Courant number of at most 0.5, the associated time step is 0.03 s. To take into account the higher harmonics in the simulation accurately, two layers will be chosen (see Table 5.2; the minimum wave period is 3.2 s, which is 2.5 times smaller than the peak period of 8 s). On a present-day computer (2.0 GHz Intel Core 2 processor) SWASH requires about 6 μs per grid point and per time step for a two-layer simulation. So, the simulation of our harbour takes about 17 days on a single processor to complete the run of 60 minutes real time. This clearly shows the need for parallel computing.

Different parallelization strategies can be considered of which the most popular are:

- data parallel programming,
- shared memory programming, and
- message passing.

Data parallel programming uses automatic parallelizing compilers which enables loop-level parallelization. Generally, this approach often will not yield high efficiency. The main reason for this is that a large portion of the existing code is in most cases inherently sequential.

On shared memory platforms with all processors using a single memory, parallelization is usually done by multithreading with the help of OpenMP compiler directives. A drawback of this approach is that forcing good parallel performance limits the number of processors only to about 16.

Obtaining good scalability for relatively large number of processors is usually achieved through distributed memory parallel machines with each processor having its own private memory. A popular example of distributed memory architecture is a cluster of Linux PCs connected via fast networks, since it is very powerful, relatively cheap and nearly available to all end-users. The conventional methodology for parallelization on distributed computing systems is domain decomposition, which not only achieves benefit from carrying out the task simultaneously on many processors but also enables a large amount of memory required. It gives efficient parallel algorithms and is easy to program within message passing environment such as MPICH2.

A parallel version of SWASH with the distributed memory parallelization paradigm using MPI standard has been developed. The message passings are implemented by a high level communication library MPICH2. Only simple point-to-point and collective communications have been employed. No other libraries or software are required. For a full three-dimensional simulation with a high resolution we expect a good scalable performance.

Refer to the example above, numerical computations have been carried out for the full simulation period of our harbour on 1 through 32 computational cores of our Linux cluster. The results show a super linear speedup of up to a factor 8.6 on 8 cores, but then it levels off to a factor of 26 on 32 cores. As such, the computing time has been reduced to 15 hours per 60 minutes to be simulated.

Appendix A

Definitions of variables

In SWASH a number of variables are used in input and output. The definitions of these variables are mostly conventional.

H_s Significant wave height, in meters, defined as

$$H_s = 4\sqrt{\int E(f)df}$$

where $E(f)$ is the variance density spectrum and f is the frequency.

With respect to output, this quantity is computed as four times the standard deviation of the surface elevation over the duration `[dur]` (see command `QUANTITY`) in the final stage of the simulation, which should be long enough to provide statistically reliable wave height. The simulation period should therefore be long enough to establish steady-state conditions.

$H_{\rm rms}$ RMS wave height, in meters, defined as

$$H_{\rm rms} = \frac{1}{2}\sqrt{2}H_s$$

T_p Peak period, in seconds, defined as inverse of the frequency at which the variance density spectrum is a maximum.

T_{m01} Mean wave period, in seconds, defined as

$$T_{m01} = \frac{\int E(f)df}{\int fE(f)df}$$

Directional spread The directional distribution of incident wave energy is given by $D(\theta) = A\cos^m(\theta)$ for all frequencies. The power m is related to the one-sided directional spread as given in Table A.1.

Cr

Courant number defined as follows:

$$\mathrm{Cr} = \Delta t \left(\sqrt{gd} + \sqrt{u^2 + v^2} \right) \sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}}$$

with Δt the time step, Δx and Δy the grid sizes in $x-$ and $y-$direction, respectively, and u and v the velocity components in $x-$ and $y-$direction.

VEL

Current velocity components in $x-$ and $y-$direction of the problem coordinate system, except in the case of output with `BLOCK` command in combination with command `FRAME`, where x and y relate to the $x-$axis and $y-$axis of the output frame.

MVEL

Mean or time-averaged velocity components in $x-$ and $y-$direction of the problem coordinate system, except in the case of output with `BLOCK` command in combination with command `FRAME`, where x and y relate to the $x-$axis and $y-$axis of the output frame. Time-averaging is carried out over the duration `[dur]` (see command `QUANTITY`) in the final stage of the simulation.

WIND

Wind velocity components in $x-$ and $y-$direction of the problem coordinate sytem, except in the case of output with `BLOCK` command in combination with command `FRAME`, where x and y relate to the $x-$axis and $y-$axis of the output frame.

TIME

Full date-time string.

TSEC

Time in seconds with respect to a reference time (see command `QUANTITY`).

Cartesian convention

The direction is the angle between the vector and the positive $x-$axis, measured counterclockwise. In other words: the direction where the waves are going to or where the wind is blowing to.

Nautical convention

The direction of the vector from geographic North measured clockwise. In other words: the direction where the waves are coming from or where the wind is blowing from.

Table A.1: Directional distribution.

m	directional spread (in ^{o})
1.	37.5
2.	31.5
3.	27.6
4.	24.9
5.	22.9
6.	21.2
7.	19.9
8.	18.8
9.	17.9
10.	17.1
15.	14.2
20.	12.4
30.	10.2
40.	8.9
50.	8.0
60.	7.3
70.	6.8
80.	6.4
90.	6.0
100.	5.7
200.	4.0
400.	2.9
800.	2.0

Appendix B

Command syntax

B.1 Commands and command schemes

The actual commands of the user to SWASH must be given in one file containing all commands. This file is called the command file. It must be presented to SWASH in ASCII. It is important to make a distinction between the description of the commands in this User Manual and the actual commands in the command file. The descriptions of the commands in this User Manual are called command schemes. Each such command scheme includes a diagram and a description explaining the structure of the command and the meaning of the keyword(s) and of the data in the command. The proper sequence of the commands is given in Section 4.2.

B.2 Command

B.2.1 Keywords

Each command instructs SWASH to carry out a certain action which SWASH executes before it reads the next command. A command must always start with a keyword (which is also the name of the command) which indicates the primary function of that command; see list in Section 4.1). A simple command may appear in its command scheme as:

```
  KEYword data
```

A command may contain more than one keyword (which refines the instructions to SWASH), e.g.,

```
  KEY1word KEY2word data
```

where KEY2word is the second keyword.

Spelling of keywords

In every command scheme, keywords appear as words in both lower- and upper-case letters. When typing the command or keyword in the command file, the user **must at least** copy literally the part with upper-case letters. SWASH reads only this part. SWASH is case insensitive except for one instance (character strings), see below. When typing the keyword in the command file, any extension of the part with upper-case letters is at the users discretion as long as the extension is limited to letters or digits, as well as the characters − and _. So, in the first command outlined above one may write: `KEY` or `KEYW` or `KEY−word` or `keyhole`, etc., whereas with the abovementioned second command scheme, `key1 KEY2 data` may appear in the command file.

In the command file

- a keyword is closed by a blank or one of the following characters = or :
- a keyword is not enclosed by square brackets or quotes,
- a keyword followed by a comma (,) is interpreted as a keyword followed by an empty data field (see below).

Required and optional keywords

All keywords in a command are required except when an option is available.

Optional keywords are indicated in the command scheme with the following signs enclosing the keywords concerned:

```
  |  KEY1word ...... data ....... |
 <                                  >
  |  KEY2word ...... data ....... |
```

For the above example it may appear as:

```
           |  KEY2word data |
 KEY1word  <                  >
           |  KEY3word data |
```

In case the user does not indicate an option in a command, SWASH chooses the alternative indicated with an arrow (->) appearing in the command scheme (the default option). In the above example, it may appear as:

```
           |    KEY2word data |
 KEY1word  <                    >
           | -> KEY3word data |
```

where `KEY3WORD` is the default option.

Repetitions of keywords and/or other data

The use of keywords is sometimes repetitive, e.g. in a sequence of data and keywords containing many locations in $x, y-$space. In such a case, the command scheme indicates this repetitive nature by placing the keywords (and data) concerned between angle brackets `< >`. For instance,

```
 KEY1word <data KEY2word data>
```

In the actual command in the command file the user must give such a sequence. It ends with either

- end of line
- a keyword other than the ones mentioned in the repetition group
- the character / or ;

If more than one line is required for a command, the user may continue on the next line as described in Section B.4. The repetition may consist of one instance (in fact, no repetition at all).

B.2.2 Data

Most commands contain data, either character data or numerical data.

Character data and numerical data

Character data (character strings) are represented in the command schemes by names, enclosed in quotes (' ').

Numerical data are represented in the command schemes by names enclosed in square brackets ([ ]).

As a rule, an error message will result if numerical data is given where character data should be given.

Spelling of data

Character data are represented as character strings (sequence of characters and blanks) between quotes (in the command scheme and in the command file). SWASH interprets an end of line as an end quote (a character data field can therefore never extend over more than one line).

In a command scheme the character string is always a name (which is placed between quotes as indicated). In the command file such a name can be entered in two ways:

- Replace the name by another character string at the users discretion (between quotes; this is the only occurrence where SWASH is case sensitive; e.g. for text to appear in a plot.
  Example:
  command scheme: `KEYword 'City' data`
  command file: `KEY 'Amsterdam' data`

- Copy the name of the variable (without the quotes) literally followed by an = sign and a name at the users discretion (between quotes). SWASH interprets the copied name in the command file as a keyword with all the characteristics of a keyword such as ending a sequence of optional data (see below). As with other keywords the name of the variable is case-insensitive.
  Example:
  command scheme: `KEYword 'City' data`
  command file: `KEY city='Amsterdam' data`

As a rule, an error message will result if numerical data is given where character data should be given.

Numerical data are simple numbers, e.g. 15 or −7 (integer data), or 13.7 or 0.8E−4 (real data). Whether or not integer number or real number should be given by the user is indicated in the description of the command scheme.

Note that a decimal point is not permitted in an integer number. On the other hand, an integer number is accepted by SWASH where a real number should be given.

In a command scheme, the number is always indicated with a name (which is placed between square brackets). In the command file such a name can be entered in two ways:

- Replace the name by a number (not between square brackets).
  Example:
  command scheme: `KEYword [nnn]`
  command file: `KEY 314`

- Copy the name of the variable (without the quotes) literally followed by an = sign and the number (not between square brackets). SWASH interprets the copied name in the command file as a keyword with all the characteristics of a keyword such as ending a sequence of optional data (see below). As with other keywords the name of the variable is case-insensitive.
  Example:
  command scheme: `KEYword [nnn]`
  command file: `KEY nnn=314`

Required data and optional data

All data must be given by the user in the command file in the same order as they appear in the command scheme. They are separated by blanks or comma's.

Required data (indicated in the description of each individual command) must be given explicitly as character string or numbers.

Optional data are indicated

(a) in the text of each individual command or

(b) for sets of data: in parenthesis around the data concerned

```
( data )
```

For example:

```
KEY1word KEY2word 'name' ([nnn] [mmm]) [zzz]
```

or

(c) some optional data are indicate in the same way as optional keywords are indicated:

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

Optional data of the kind (a) or (b) may be omitted by giving blanks between comma's (SWASH then substitutes reasonable default values). If after a required datum all data is optional (till the next keyword or the next end-of-line), then the comma's may be omitted too. Optional data of the kind (c) are to be treated in the same way as optional keywords.

B.3 Command file and comments

All text after one $ or between two $ signs on one line in the command file is ignored by SWASH as comment. Such comments may be important to the user e.g., to clarify the meaning of the commands used. In fact, this option has been used to create the edit file `swash.edt` (see Appendix C). Anything appearing after two $ signs is not interpreted as comment, but again as data to be processed (possibly interrupted again by $ or two $ signs). Alternatively, the exclamation mark '!' can be used as comment sign. Everthing behind a ! is interpreted as comment, also if ! or $ are in that part of the input line.

B.4 End of line or continuation

A command in the command file may be continued on the next line if the previous line terminates with a continuation mark & or _ (underscore).

Appendix C

File swash.edt

Below the file `swash.edt` is presented in which all the commands that can be used with SWASH are specified.

```
!   PROJECT  'name'  'nr'
!            'title1'
!            'title2'
!            'title3'
!
!   SET [level] [nor] [depmin] [maxmes] [maxerr] [seed]                        &
!       [grav] [rhowat] [temp] [salinity] [dynvis] [rhoair] [rhosed]           &
!       [cdcap] [prmean] [backvisc] [kappa] [latitude]                         &
!       CARTesian|NAUTical  [epshu]  CC                                        &
!       [printf]  [prtest]  [outlev]
!
!   MODE  DYNamic  / -> TWODimensional \  (SKIPMOMentum)  (LINear)
!                  \    ONEDimensional /
!
!   COORDinates  /  -> CARTesian
!                \ SPHErical      CCM|QC
!
!          | REGular [xpc] [ypc] [alpc] [xlenc] [ylenc] [mxc] [myc]  |
!   CGRID <  CURVilinear [mxc] [myc]  (EXC [xexc] [yexc])             >        &
!          | UNSTRUCtured                                            |
!
!          REPeating X|Y
!
!   VERTical [kmax] < [thickness] M|PERC >
!
!   INPgrid  BOT|WLEV|CUR|VX|VY|FRic|WInd|WX|WY|PRessure|CORIolis|POROsity|    &
!            PSIZ|HSTRUC|NPLAnts|DRAFt|LABel|SALinity|TEMPerature|SEDiment|    &
!            MWL|ACUR|AVX|AVY                                                   &
!
!      | REG [xpinp] [ypinp] [alpinp]  [mxinp] [myinp]  [dxinp] [dyinp] |
!      |                                                                |
!     <  CURVilinear STAGgered                                           >     &
!      |                                                                |
!      | UNSTRUCtured                                                   |
!
!      (EXCeption  [excval])                                                    &
!
!      (NONSTATionary [tbeginp] [deltinp] SEC|MIN|HR|DAY [tendinp])            &
!
!      (NONUNIForm  [kmax])
!
!   READgrid UNSTRUCtured / -> TRIAngle \
!                         \    EASYmesh / 'fname'
!
!   READinp   BOTtom|WLEVel|CURrent|FRic|WInd|PRessure|COOR|CORIolis|POROsity| &
!             PSIZ|HSTRUC|NPLAnts|DRAFt|LABel|SALinity|TEMPerature|SEDiment|   &
!             MWL|ACURrent                                                      &
!
!                | 'fname1'        |
!        [fac]  <  SERIes 'fname2'  >  [idla] [nhedf] ([nhedt]) (nhedvec])     &
!                | LAYers 'fname3' |
!
!        FREE | FORMAT 'form' | [idfm] | UNFORMATTED
!
!             | -> CONstant [wlev] [vx] [vy] [tke] [epsilon]
!             |
!   INITial  <     ZERO
!             |
!             |    STEAdy
!
!
!                    |    PM              |
!   BOUnd SHAPespec <  -> JONswap [gamma]  > SIG|RMS  PEAK|MEAN  DSPR POW|DEGR
!                    |    TMA             |
!
!              / -> SIDE  N|NW|W|SW|S|SE|E|NE | [k]  CCW|CLOCKWise  \
!   BOUndcond <                                                      >         &
!              \    SEGment  / -> XY  < [x]  [y] >            \     /
!                            \    IJ  < [i]  [j] > | < [k] >  /
```

```
!
!        BTYPe WLEV|VEL|DISCH|RIEMann|LRIEmann|WEAKrefl|SOMMerfeld|OUTFlow     &
!
!        LAYer [k] | HYPerbolic | LOGarithmic                                  &
!
!        SMOOthing [period] SEC|MIN|HR|DAY                                     &
!
!        ADDBoundwave | ADDIG                                                  &
!
!                | -> FOURier     [azero] < [ampl] [omega] [phase] >
!                |    REGular     [h] [per] [dir]
!                |    BICHromatic [h1] [h2] [per1] [per2] [dir1] [dir2]
!    | UNIForm  <     SPECTrum    [h] [per] [dir] [dd] [cycle] SEC|MIN|HR|DAY
!    |           |    SERIes      ’fname’ [itmopt]
!    |           |    SPECFile    ’fname’ [cycle] SEC|MIN|HR|DAY
!   <                                                                          &
!    |           | -> FOURier  < [len] [azero] < [ampl] [omega] [phase] > >
!    |           |    REGular  < [len] [h] [per] [dir] >
!    |           |    BICHrom  < [len] [h1] [h2] [per1] [per2] [dir1] [dir2] >
!    | VARiable <     SPECTrum < [len] [h] [per] [dir] [dd] [cycle] S|MI|HR|DA >
!                |    SERIes   < [len] ’fname’ [itmopt] >
!                |    SPECFile < [len] ’fname’ [cycle] SEC|MIN|HR|DAY >
!                |    SPECSwan ’fname’ [cycle] SEC|MIN|HR|DAY
!
!
!   SOURce   X|Y | [k]   [centre] [width] [depth] [delta]                      &
!
!             / REGular  [h] [per] [dir]                             \
!             \ SPECTrum [h] [per] [dir] [dd] [cycle] SEC|MIN|HR|DAY /         &
!
!            SMOOthing [period] SEC|MIN|HR|DAY
!
!
!   SPONgelayer  N|NW|W|SW|S|SE|E|NE [width] | < [k]  [width] >
!
!
!   FLOAT  [alpha] [theta]
!
!
!   BODY  DIMension [l] [mass] [Ix] [Iy] [Iz] [cogx] [cogy] [cogz]             &
!
!         DOF SUrge SWay HEave ROll PItch YAw                                  &
!
```

```
!         ( MLIne < [K] [B] [apbx] [apby] [apbz] [apfx] [apfy] [apfz] [elen] > &
!
!                 PRETension )                                                 &
!
!         ( FENder < [K] [apfx] [apfy] [apfz] > )
!
!
!                 |    NEWmark [beta] [gamma] |
!                 |                           |
!                 | -> CH [rho]               |
!   BODY  SOLVer <                             > COUPling [tol] [maxiter] [relax] &
!                 |    HHT [rho]              |
!                 |                           |
!                 |    WBZ [rho]              |
!
!         KBC [theta]
!
!
!                     | -> CONstant [cd]                       |
!                     |                                        |
!                     |    CHARNock [beta] [height]            |
!                     |                                        |
!                     |    LINear [a1] [a2] [b] [wlow] [whigh] |
!                     |                                        |
!                     |    WU                                  |   | REL  [alpha]
!   WIND [vel] [dir] <                                          > <
!                     |    GARRatt                             |   | RELW [crest]
!                     |                                        |
!                     |    SMIthbanke                          |
!                     |                                        |
!                     |    CHEn                                |
!                     |                                        |
!                     |    FIT                                 |
!
!
!              |    LINear [k]
!              |
!              |    CONstant [cf]
!              |
!              |    CHEZy [cf]
!              |
!   FRICtion  <  -> MANNing [cf]
!              |
!              |    COLEbrook [h]
!              |
!              |            | -> SMOOTH
!              |    LOGlaw <
!                           |    ROUGHness [h]
!
!
!
!                               | -> CONstant [visc]
!                               |
!              | -> Horizontal <     SMAGorinsky [cs]
!              |                |
!              |                |    MIXing [lm]
!              |
!              |
!   VISCosity <     Vertical  KEPS [cfk] [cfe]
!              |
!              |
!              |                | -> LINear
!              |    FULL  KEPS <
!              |                |    NONLinear
!
!
!
!   POROsity  [size] [height] [alpha0] [beta0] [wper]
!
!
!   VEGEtation < [height] [diamtr] [nstems] [drag] > MASS [cm] POROsity Vertical
!
!
!   CORIolis  [fpar] [epsab2]
!
!
!                          | -> Sec |    | -> NONCohesive [size]            |
!   TRANSP [diff] [retur] <     MIn  >  <                                    > &
!                          |    HR  |    | COHesive [tauce] [taucd] [erate] |
!                          |    DAy |
!
!          [fall] [snum] [ak] DENSity Y|N   [alfa] [crsn] [cp] [ek]            &
!
!          ANTICreep STAndard|SVK|None
!
!
```

```
!   BREaking  [alpha] [beta] [nufac]
!
!
!   AMBient  [U] [V] [eta]  Cell|Stagg  [theta]
!
!
!                    |    STAndard      |
!                    |                  |
!   NONHYDrostatic  <  -> BOX            > [theta]                             &
!                    |                  |
!                    |    DEPthaveraged |
!
!                   SUBGrid [pmax]  REDuced [qlay]                             &
!
!                   SOLVer [rhsaccur] [initaccur] [maxiter] [relax] [precfq]   &
!
!                   PREConditioner  ILUDS|ILUD|ILU|NONE                        &
!
!                   PROJection  ITERative [tol] [maxiter]
!
!
!                   |         | UMOM  MOMentum|HEAD  / -> Horizontal   |
!                   |         |                      \    Vertical     |
!                   | UPWind <                                         |
!                   |         | WMOM  / -> Horizontal                  |
!                   |         |       \    Vertical                    |
!                   |                                                  |
!                   | CORRdep                                          |
!   DISCRETization <                                                    >      &
!                   | TRANSPort  / -> Horizontal                       |
!                   |            \    Vertical                         |
!                   |                                                  |
!                   | MIMEtic                                          |
!                   |                                                  |
!                   |           | -> Umom                              |
!                   | ACURrent <                                       |
!                   |           | Wmom                                 |
!
!
!                      | NONe                                 |
!                      |                                      |
!                      | FIRstorder                           |
!                      |                                      |
!                      | HIGherorder [kappa]                  |
!                      |                                      |
!                      |          | -> SWEBy [phi]            |
!                      |          |                           |
!                      | LIMiter <  RKAPpa [kappa]            |
!                      |          |                           |
!                      |          | PLKAPpa [kappa] [mbound]  |
!                      | FROmm                                |
!                      |                                      |
!                      | -> BDF | LUDs                        |
!                      |                                      |
!                     <  QUIck                                 >
!                      |                                      |
!                      | CUI                                  |
!                      |                                      |
!                      | MINMod                               |
!                      |                                      |
!                      | SUPerbee                             |
!                      |                                      |
!                      | VANLeer                              |
!                      |                                      |
!                      | MUScl                                |
!                      |                                      |
!                      | KORen                                |
!                      |                                      |
!                      | SMArt                                |
!
!
!   DPSopt  MIN|MEAN|MAX|SHIFt
!
!
!               | EXPL [cfllow] [cflhig]  (HANCock|EULer)       |
!               |                                               |
!   TIMEI METH <  IMPL [thetac] [thetas] &                       >             &
!               |                                               |
!               |      SOLVer [tol] [maxiter] [weight]  NEWTon  |
!
!         VERTical [thetau] [thetaw] [thetat]
!
!
!   FRAME   'sname'  [xpfr] [ypfr] [alpfr] [xlenfr] [ylenfr] [mxfr] [myfr]
!
!   GROUP   'sname'  SUBGRID [ix1] [ix2] [iy1] [iy2]
```

```
!
!   CURVE   'sname'  [xp1] [yp1]   < [int]  [xp]  [yp] >
!
!   RAY     'rname'  [xp1] [yp1] [xq1] [yq1]                                   &
!                    <  [int]  [xp]  [yp]  [xq]  [yq]  >
!
!   ISOLINE 'sname'  'rname'  DEPTH|BOTTOM  [dep]
!
!   POINTS  'sname'  < [xp]  [yp]  >     |    FILE 'fname'
!
!
!              |...|
!   QUANTity  <     >   'short'  'long'  [lexp]  [hexp]  [excv]  [ref]         &
!              |...|
!
!                                        [dur] SEC|MIN|HR|DAY [depth] [delrp]  &
!
!                                        [xcom] [ycom] [zcom] [alpobj]         &
!
!             / -> PROBLEMcoord \
!             \ FRAME           /
!
!   OUTPut OPTions  'comment'  (TABle [field])  (BLOck  [ndec]  [len])
!
!   BLOCK   'sname'  HEAD | NOHEAD  'fname' (LAY-OUT [idla])                   &
!           < TSEC|XP|YP|DEP|BOTL|WATL|DRAF|VMAG|VDIR|VEL|VKSI|VETA|           &
!             PRESS|NHPRES|QMAG|QDIR|DISCH|QKSI|QETA|VORT|WMAG|WDIR|WIND|      &
!             FRC|USTAR|UFRIC|HRUN|BRKP|ZK|HK|VMAGK|VDIRK|VELK|VKSIK|          &
!             VETAK|VZ|VOMEGA|QMAGK|QDIRK|DISCHK|QKSIK|QETAK|PRESSK|           &
!             NHPRSK|TKE|EPS|VISC|HS|HRMS|SETUP|MVMAG|MVDIR|MVEL|MVKSI|        &
!             MVETA|MVMAGK|MVDIRK|MVELK|MVKSIK|MVETAK|MTKE|MEPS|MVISC|         &
!             SAL|TEMP|SED|MSAL|MTEMP|MSED|SALK|TEMPK|SEDK|MSALK|MTEMPK|       &
!             MSEDK [unit] >                                                   &
!           (OUTPUT [tbegblk] [deltblk] SEC|MIN|HR|DAY)
!
!   TABLE   'sname'  HEAD | NOHEAD | STAB | SWASH | IND  'fname'               &
!           < TIME|TSEC|XP|YP|DIST|DEP|BOTL|WATL|DRAF|VMAG|VDIR|VEL|VKSI|VETA| &
!             PRESS|NHPRES|QMAG|QDIR|DISCH|QKSI|QETA|VORT|WMAG|WDIR|WIND|FRC|  &
!             USTAR|UFRIC|HRUN|BRKP|ZK|HK|VMAGK|VDIRK|VELK|VKSIK|VETAK|VZ|     &
!             VOMEGA|QMAGK|QDIRK|DISCHK|QKSIK|QETAK|PRESSK|NHPRSK|TKE|EPS|     &
!             VISC|HS|HRMS|SETUP|MVMAG|MVDIR|MVEL|MVKSI|MVETA|MVMAGK|MVDIRK|   &
!             MVELK|MVKSIK|MVETAK|MTKE|MEPS|MVISC|SAL|TEMP|SED|MSAL|MTEMP|     &
!             MSED|SALK|TEMPK|SEDK|MSALK|MTEMPK|MSEDK|FORCEX|FORCEY|FORCEZ|    &
!            MOMX|MOMY|MOMZ|TRAX|TRAY|TRAZ|ROTX|ROTY|ROTZ|PTOP|RUNUP >        &
!           (OUTPUT [tbegtbl] [delttbl] SEC|MIN|HR|DAY)
!
!
!
!                               / -> IJ < [i] [j] > | < [k] > \
!   TEST [itest] [itrace] POINTS <                               > 'fname'
!                               \    XY < [x] [y] >           /
!
!
!                            | -> Sec  |
!   COMPute   [tbegc] [deltc] <     MIn   > [tendc]
!                            |    HR   |
!                            |    DAy  |
!
!   STOP
```

Bibliography

[1] P. Smit, M. Zijlema, and G. S. Stelling. Depth-induced wave breaking in a non-hydrostatic, near-shore wave model. *Coast. Engng.*, 76:1–16, 2013.

[2] G. S. Stelling and S. P. A. Duinmeijer. A staggered conservative scheme for every Froude number in rapidly varied shallow water flows. *Int. J. Numer. Meth. Fluids*, 43:1329–1354, 2003.

[3] G. S. Stelling and J. A. Th. M. Van Kester. On the approximation of horizontal gradients in sigma co-ordinates for bathymetry with steep slopes. *Int. J. Numer. Meth. Fluids*, 18:915–935, 1994.

[4] G. S. Stelling and M. Zijlema. An accurate and efficient finite difference algorithm for non-hydrostatic free-surface flow with application to wave propagation. *Int. J. Numer. Meth. Fluids*, 43:1–23, 2003.

[5] M. Zijlema. Computation of free surface waves in coastal waters with SWASH on unstructured grids. *Comput. Fluids*, 213, 2020. Article 104751.

[6] M. Zijlema and G. S. Stelling. Further experiences with computing non-hydrostatic free-surface flows involving water waves. *Int. J. Numer. Meth. Fluids*, 48:169–197, 2005.

[7] M. Zijlema and G. S. Stelling. Efficient computation of surf zone waves using the nonlinear shallow water equations with non-hydrostatic pressure. *Coast. Engng.*, 55:780–790, 2008.

[8] M. Zijlema, G. S. Stelling, and P. B. Smit. SWASH: an operational public domain code for simulating wave fields and rapidly varied flows in coastal waters. *Coast. Engng.*, 58:992–1012, 2011.

Index