U. S. Department of Commerce
National Oceanic and Atmospheric Administration
National Weather Service
National Centers for Environmental Prediction
4700 Silver Hill Road, Mail Stop 9910
Washington, DC 20233-9910

Technical Note

User manual and system documentation of WAVEWATCH-III version 2.22 †

Hendrik L. Tolman ‡
SAIC-GSO at the
Environmental Modeling Center
Marine Modeling and Analysis Branch

September 2002

---

† MMAB Contribution No. 222.
‡ e-mail: Hendrik.Tolman@NOAA.gov

This page is intentionally left blank.

This release of WAVEWATCH III
is dedicated to the memory of my son

Daniel Lieuwe Tolman
April 5, 2002 - April 8, 2002

Contents

This page is intentionally left blank.

1 Introduction

1.1 About this manual

This is the user manual and system documentation of version 2.22 of the full-spectral third-generation wind-wave model WAVEWATCH III (henceforth denoted as WWATCH). WWATCH has been developed at the Marine Modeling and Analysis Branch (MMAB) of the Environmental Modeling Center (EMC) of the National Centers for Environmental Prediction (NCEP). It is based on WAVEWATCH I and WAVEWATCH II as developed at Delft University of Technology, and NASA Goddard Space Flight Center, respectively. WWATCH differs from its predecessors in all major aspects; i.e., governing equations, program structure, numerical and physical approaches.

This manual describes the governing equations, numerical approaches, compilation, and running of WWATCH. The format of a combined user manual and system documentation has been chosen, to give users the necessary background to upgrade the model according to their own specifications. Whereas this document is intended to be complete and self-contained, this is not the case for all elements in the system documentation. For additional system details, reference is made to the source code, which is fully documented.

The governing equations and numerical approaches used in WWATCH are described in chapters 2 and 3. Running the model is described in chapter 4. Installing WWATCH is described in chapter 5. Finally, a short system documentation is given in chapter 6. A thorough knowledge of WWATCH can be obtained by following chapters 2 through chapter 5. A shortcut is to first install the model (chapter 5), and then successively modify input files in example runs (chapter 4).

The previously released version of WWATCH is version 1.18. Since this release major code changes have been introduced. Although transparent for the physics and numerics, a major change was the conversion to FORTRAN 90. With this change, the code has been entirely reorganized in modules, all COMMON blocks have been removed, and all relevant arrays have been made allocatable. The latter removes the necessity to recompile models for new applications. The grid preprocessor automatically produces default model

setting, unless these values are overwritten with optional namelist input. Default setting for numerical schemes and physics parameterizations are now also identified. Cray shared memory parallelization directives have been replaced by more generally applicable OpenMP directives. With these changes, the GRIB post-processor[1] has been made part of the general distribution, as well as graphical postprocessing using GrADS [2] software. The NCAR graphical post-processors have been discontinued. This publication thus replaces Tolman (1999a,b), whereas Tolman (1999c) has become obsolete.

Additional changes to the model include new physics and numerical options and approaches. New physics parameterizations include of exact nonlinear interactions using the Webb-Resio-Tracy method as provided by Gerbrant Ph. van Vledder (Van Vledder, 2002b). New numerical approaches include the change of a central to forward time integration scheme for the source terms, and additional options for numerical propagation schemes. Also new are new Garden Sprinkler Effect alleviation methods, as described in Tolman (2002a), and a method to deal with unresolved islands and ice as described in Tolman (2002e). Finally, the model can now be run on either a spherical (longitude-latitude) grid, or on a Cartesian grid. A review of the impact of all these changes on NCEP's operational wave models can be found in Tolman (2002d).

Finally, the new model is prepared for data assimilation, with a general data ingest procedure, and an interface routine to include data assimilation software. Actual data assimilation software is not provided at this time.

Two final notes of caution. First, the model definition files for the new version of WWATCH are not downward compatible. It is therefore prudent to set up the new version of WWATCH in its own separate directory structure. Secondly, WAVEWATCH III includes wave-current interactions. The implementation of these interactions has only been tested in idealized test cases. Tests with realistic conditions have not yet been performed.

Up to date information on this model can be found (including bugs and bug fixes) on the WAVEWATCH III web page, and comments, questions and suggestions should be directed to the corresponding E-mail address

[1] Requires user-supplied GRIB packing routines.

[2] See http://www.iges.org/grads for source code and description.

http://polar.ncep.noaa.gov/waves/wavewatch
NCEP.EMC.wavewatch@NOAA.gov

Questions regarding the WRT nonlinear interaction routines should be directed to Gerbrant Ph. van Vledder

vledder@alkyon.nl

1.2 Disclaimers

The National Weather Service (NWS) supplies the source code of WAVEWATCH III and additional utilities as **public domain** software, which may be used freely by the public. For any use, proper reference should be made to the origin of the software, and all modifications by the user should be properly documented.

The user assumes the entire risk related to the use of this software. NWS is providing the software 'as is', and NWS disclaims any and all warranties, whether express or implied, including (without limitation) any implied warranties of merchantability or fitness for a particular purpose. In no event will NWS be liable to you or any third party for any direct, indirect, incidental, consequential, special or exemplary damages or lost profit resulting from use or misuse of this software.

The Marine Modeling and Analysis Branch (MMAB) of the Environmental Modeling Center (EMC) of the National Centers for Environmental Prediction (NCEP) of NWS, will not provide support for implementation or execution of WAVEWATCH III.

All above disclaimers of NWS also apply to individual authors of the WAVEWATCH III source code and corresponding publications.

1.3 Contributors and acknowledgments

Even in its original development, when I was working on the WWATCH code mostly on my own, many have contributed to the success of the model. With the expansion of physical and numerical parameterizations available, the list of contributors to this model is growing. I would like to recognize the following contributors (in alphabetic order) :

Nico Booij (Delft University of Technology, The Netherlands)
Original design of source code pre-processor (`w3adc`), basic method of documentation and other programming habits. Spatially varying wavenumber grid.

Dmitry V. Chalikov (UCAR - NOAA/NCEP, USA)
Co-author of the Tolman and Chalikov (1996) input and dissipation parameterizations and source code.

Gerbrant Ph. van Vledder (Alkyon Hydraulic Consultancy & Research, NL)
Provided the routines and documentation for the Webb-Resio-Tracy exact nonlinear interaction routines, as well as some of the original service routines.

The development of WAVEWATCH III has been an ongoing process for well over a decade. The development of WAVEWATCH I was entirely funded through my Ph.D. work at Delft University. The development of WAVEWATCH II has been funded entirely through my position as a National Research Council Resident Research Associate at NASA, Goddard Space Flight Center. The initial development of WAVEWATCH III version 1.18 was entirely funded by NOAA/NCEP, with most funding provided by the NOAA High Performance Computing and Communication (HPCC) office. Developments of the present release of WAVEWATCH III then have been funded similarly through NOAA/NCEP.

I would finally like to thank all users who have reported errors and glitches, or have made suggestions for improvements, particularly those who have beta-tested this release (Rique Alves, Fabrice Ardhuin, Erick Rogers, Gerbrant van Vledder and Paul Wittmann).

2 Governing equations

2.1 Introduction

Waves or spectral wave components in water with limited depth and non-zero mean currents are generally described using several phase and amplitude parameters. Phase parameters are the wavenumber vector $\mathbf{k}$, the wavenumber k, the direction θ and several frequencies. If effects of mean currents on waves are to be considered, a distinction is made between the relative or intrinsic (radian) frequency σ $(= 2\pi f_r)$, which is observed in a frame of reference moving with the mean current, and the absolute (radian) frequency ω $(= 2\pi f_a)$, which is observed in a fixed frame of reference. The direction θ is by definition perpendicular to the crest of the wave (or spectral component), and equals the direction of $\mathbf{k}$. Generally, scales of variation of depths and currents are assumed to be much larger than those of an individual wave. The quasi-uniform (linear) wave theory then can be applied locally, giving the following dispersion relation and Doppler type equation to interrelate the phase parameters

$$\sigma^2 = gk \tanh kd \,, \tag{2.1}$$

$$\omega = \sigma + \mathbf{k} \cdot \mathbf{U} \,, \tag{2.2}$$

where d is the mean water depth and $\mathbf{U}$ is the (depth- and time- averaged) current velocity. The assumption of slowly varying depths and currents implies a large-scale bathymetry, for which wave diffraction can generally be ignored. The usual definition of $\mathbf{k}$ and ω from the phase function of a wave or wave component implies that the number of wave crests is conserved (see, e.g., Phillips, 1977; Mei, 1983)

$$\frac{\partial \mathbf{k}}{\partial t} + \nabla\omega = 0 \,. \tag{2.3}$$

From Eqs. (2.1) through (2.3) the rates of change of the phase parameters can be calculated (e.g., Christoffersen, 1982; Mei, 1983; Tolman, 1990, equations not reproduced here).

For monochromatic waves, the amplitude is described as the amplitude, the wave height, or the wave energy. For irregular wind waves, the (random)

variance of the sea surface is described using variance density spectra (in the wave modeling community usually denoted as energy spectra). The variance spectrum F is a function of all independent phase parameters, i.e., $F(\mathbf{k},\sigma,\omega)$, and furthermore varies in space and time, e.g., $F(\mathbf{k},\sigma,\omega;\mathbf{x},t)$. However, it is usually assumed that the individual spectral components satisfy the linear wave theory (locally), so that Eqs. (2.1) and (2.2) interrelate $\mathbf{k}$, σ and ω. Consequently only two independent phase parameters exist, and the local and instantaneous spectrum becomes two-dimensional. Within WWATCH the basic spectrum is the wavenumber-direction spectrum $F(k,\theta)$, which has been selected because of its invariance characteristics with respect to physics of wave growth and decay for variable water depths. The output of WWATCH, however, consists of the more traditional frequency-direction spectrum $F(f_r,\theta)$. The different spectra can be calculated from $F(k,\theta)$ using straightforward Jacobian transformations

$$F(f_r,\theta) = \frac{\partial k}{\partial f_r} F(k,\theta) = \frac{2\pi}{c_g} F(k,\theta) \;, \tag{2.4}$$

$$F(f_a,\theta) = \frac{\partial k}{\partial f_a} F(k,\theta) = \frac{2\pi}{c_g} \left(1 + \frac{\mathbf{k}\cdot\mathbf{U}}{k c_g} \right)^{-1} F(k,\theta) \;, \tag{2.5}$$

$$c_g = \frac{\partial \sigma}{\partial k} = n \frac{\sigma}{k} \;,\; n = \frac{1}{2} + \frac{kd}{\sinh 2kd} \;, \tag{2.6}$$

where c_g is the so-called group velocity. From any of these spectra one-dimensional spectra can be generated by integration over directions, whereas integration over the entire spectrum by definition gives the total variance E (in the wave modeling community usually denoted as the wave energy).

In cases without currents, the variance (energy) of a wave package is a conserved quantity. In cases with currents the energy or variance of a spectral component is no longer conserved, due to the work done by current on the mean momentum transfer of waves (Longuet-Higgins and Stewart, 1961, 1962). In a general sense, however, wave action $A \equiv E/\sigma$ is conserved (e.g., Whitham, 1965; Bretherthon and Garrett, 1968). This makes the wave action density spectrum $N(k,\theta) \equiv F(k,\theta)/\sigma$ the spectrum of choice within the model. Wave propagation then is described by

$$\frac{DN}{Dt} = \frac{S}{\sigma} \;, \tag{2.7}$$

where D/Dt represents the total derivative (moving with a wave component) and S represents the net effect of sources and sinks for the spectrum F. Because the left side of Eq. (2.7) generally considers linear propagation, effects of nonlinear wave propagation (i.e., wave-wave interactions) arise in S. Propagation and source terms will be discussed separately in the following sections.

2.2 Propagation

In a numerical model, a Eulerian form of the balance equation (2.7) is needed. This balance equation can either be written in the form of a transport equation (with velocities outside the derivatives), or in a conservation form (with velocities inside the derivatives). The former form is valid for the vector wavenumber spectrum $N(\mathbf{k};\mathbf{x},t)$ only, whereas valid equations of the latter form can be derived for arbitrary spectral formulations, as long as the corresponding Jacobian transformation as described above is well behaved (e.g., Tolman and Booij, 1998). Furthermore, the conservation equation conserves total wave energy/action, unlike the transport equation. This is an important feature of an equation when applied in a numerical model. The balance equation for the spectrum $N(k,\theta;\mathbf{x},t)$ as used in WWATCH is given as (for convenience of notation, the spectrum is henceforth denoted simply as N)

$$\frac{\partial N}{\partial t} + \nabla_x \cdot \dot{\mathbf{x}} N + \frac{\partial}{\partial k}\dot{k} N + \frac{\partial}{\partial \theta}\dot{\theta} N = \frac{S}{\sigma} \,, \tag{2.8}$$

$$\dot{\mathbf{x}} = \mathbf{c}_g + \mathbf{U} \,, \tag{2.9}$$

$$\dot{k} = -\frac{\partial \sigma}{\partial d}\frac{\partial d}{\partial s} - \mathbf{k} \cdot \frac{\partial \mathbf{U}}{\partial s} \,, \tag{2.10}$$

$$\dot{\theta} = -\frac{1}{k}\left[\frac{\partial \sigma}{\partial d}\frac{\partial d}{\partial m} - \mathbf{k} \cdot \frac{\partial \mathbf{U}}{\partial m}\right] \,, \tag{2.11}$$

where $\mathbf{c}_g$ is given by c_g and θ, s is a coordinate in the direction θ and m is a coordinate perpendicular to s. Equation (2.8) is valid for a Cartesian grid. For large-scale applications, this equation is usually transferred to a spherical grid, defined by longitude λ and latitude ϕ, but maintaining the definition of the local variance (i.e., per unit surface, as in WAMDIG, 1988)

$$\frac{\partial N}{\partial t} + \frac{1}{\cos\phi}\frac{\partial}{\partial\phi}\dot{\phi}N\cos\theta + \frac{\partial}{\partial\lambda}\dot{\lambda}N + \frac{\partial}{\partial k}\dot{k}N + \frac{\partial}{\partial\theta}\dot{\theta}_g N = \frac{S}{\sigma}\,, \tag{2.12}$$

$$\dot{\phi} = \frac{c_g\cos\theta + U_\phi}{R}\,, \tag{2.13}$$

$$\dot{\lambda} = \frac{c_g\sin\theta + U_\lambda}{R\cos\phi}\,, \tag{2.14}$$

$$\dot{\theta}_g = \dot{\theta} - \frac{c_g\tan\phi\cos\theta}{R}\,, \tag{2.15}$$

where R is the radius of the earth and U_ϕ and U_λ are current components. Equation (2.15) includes a correction term for propagation along great circles, using a Cartesian definition of θ where $\theta = 0$ corresponds to waves travelling from west to east. WWATCH can be run on either a Cartesian of spherical grid. Note that unresolved obstacles such as islands can be included in the equations. In WWATCH this is done at the level of the numerical scheme, as is discussed in section 3.3.3.

2.3 Source terms

2.3.1 General concepts

The net source term S is generally considered to consist of three parts, a wind-wave interaction term S_{in}, a nonlinear wave-wave interactions term S_{nl} and a dissipation ('whitecapping') term S_{ds}. In shallow water additional processes have to be considered, most notably wave-bottom interactions S_{bot} (e.g., Shemdin et al., 1978). This defines the general source terms used in WWATCH as

$$S = S_{in} + S_{nl} + S_{ds} + S_{bot}\,. \tag{2.16}$$

Other source terms are easily added. These source terms are defined for the *energy* spectra. In the model, however, most source terms are directly calculated for the action spectrum. The latter source terms are denoted as $\mathcal{S} \equiv S/\sigma$.

The treatment of the nonlinear interactions defines a third-generation wave model. Therefore, the options for the calculation of S_{nl} will be discussed first in section 2.3.2 through 2.3.3. S_{in} and S_{ds} represent separate processes, but should be considered as interrelated, because the balance of these two source terms governs the integral growth characteristics of the wave model. Two combinations of these basic source terms are available, those of WAM cycles 1 through 3 (section 2.3.4) and the parameterizations of Tolman and Chalikov (1996) (section 2.3.5). Shallow water source terms or source terms describing special physical processes are considered to be "additional" source terms. Presently only the JONSWAP formulation for bottom friction (section 2.3.6) is available. Note that all default model settings, including the selected source terms, are discussed in section 5.4.3.

A third-generation wave model effectively integrates the spectrum only up to a cut-off frequency f_{hf} (or wavenumber k_{hf}). Above this frequency a parametric tail is applied (e.g., WAMDIG, 1988)

$$F(f_r,\theta) = F(f_{r,hf},\theta)\left(\frac{f_r}{f_{r,hf}}\right)^{-m} \tag{2.17}$$

which is easily transformed to any other spectrum using the Jacobian transformations as discussed above. For instance, for the present action spectrum, the parametric tail can be expressed as (assuming deep water for the wave components in the tail)

$$N(k,\theta) = N(k_{hf},\theta)\left(\frac{f_r}{f_{r,hf}}\right)^{-m-2} \tag{2.18}$$

The actual values of m and the expressions for $f_{r,hf}$ depend on the source term parameterization used, and will be given below.

Before actual source term parameterizations are described, the definition of the wind requires some attention. In cases with currents, one can either consider the wind to be defined in a fixed frame of reference, or in a frame of reference moving with the current. Both definitions are available in WWATCH, and can be selected during compilation. The output of the program, however, will always be the wind speed which is not in any way corrected for the current.

2.3.2 Nonlinear interactions (DIA)

Nonlinear wave-wave interactions can be modeled using the discrete interaction approximation (DIA, Hasselmann et al., 1985). This parameterization was originally developed for the spectrum $F(f_r,\theta)$. To assure the conservative nature of S_{nl} for this spectrum (which can be considered as the "final product" of the model), this source term is calculated for $F(f_r,\theta)$ instead of $N(k,\theta)$, using the conversion (2.4).

Resonant nonlinear interactions occur between four wave components (quadruplets) with wavenumber vector $\mathbf{k}_1$ through $\mathbf{k}_4$. In the DIA, it is assumed that $\mathbf{k}_1 = \mathbf{k}_2$. Resonance conditions then require that

$$\left.\begin{array}{rcl} \mathbf{k}_1 + \mathbf{k}_2 & = & \mathbf{k}_3 + \mathbf{k}_4 \\ \sigma_2 & = & \sigma_1 \\ \sigma_3 & = & (1+\lambda_{nl})\sigma_1 \\ \sigma_4 & = & (1-\lambda_{nl})\sigma_1 \end{array}\right\} \quad , \tag{2.19}$$

where λ_{nl} is a constant. For these quadruplets, the contribution δS_{nl} to the interaction for each discrete (f_r,θ) combination of the spectrum corresponding to $\mathbf{k}_1$ is calculated as

$$\begin{pmatrix} \delta S_{nl,1} \\ \delta S_{nl,3} \\ \delta S_{nl,4} \end{pmatrix} = D \begin{pmatrix} -2 \\ 1 \\ 1 \end{pmatrix} C g^{-4} f_{r,1}^{11} \times \left[F_1^2 \left(\frac{F_3}{(1+\lambda_{nl})^4} + \frac{F_4}{(1-\lambda_{nl})^4} \right) - \frac{2F_1F_3F_4}{(1-\lambda_{nl}^2)^4} \right] , \tag{2.20}$$

where $F_1 = F(f_{r,1},\theta_1)$ etc. and $\delta S_{nl,1} = \delta S_{nl}(f_{r,1},\theta_1)$ etc., C is a proportionality constant. The nonlinear interactions are calculated by considering a limited number of combinations (λ_{nl}, C). In practice, only one combination is used. Default values for different source term packages are presented in Table 2.1.

This source term is developed for deep water, using the appropriate dispersion relation in the resonance conditions. For shallow water the expression is scaled by the factor D (still using the deep-water dispersion relation, however)

$$D = 1 + \frac{c_1}{\bar{k}d} \left[1 - c_2 \bar{k} d \right] e^{-c_3 \bar{k} d} \, . \tag{2.21}$$

	λ_{nl}	C
WAM-3	0.25	$2.78\ 10^7$
Tolman and Chalikov	0.25	$1.00\ 10^7$

Table 2.1: Default constants in DIA for input-dissipation packages.

Recommended (default) values for the constants are (Hasselmann and Hasselmann, 1985) $c_1 = 5.5$, $c_2 = 5/6$ and $c_3 = 1.25$. The overbar notation denotes straightforward averaging over the spectrum. For an arbitrary parameter z the spectral average is given as

$$\bar{z} = E^{-1} \int_0^{2\pi} \int_0^{\infty} zF(f_r,\theta)\, df_r\, d\theta\,, \tag{2.22}$$

$$E = \int_0^{2\pi} \int_0^{\infty} F(f_r,\theta)\, df_r\, d\theta\,, \tag{2.23}$$

For numerical reasons, however, the mean relative depth is estimated as

$$\bar{k}d = 0.75\hat{k}d\,, \tag{2.24}$$

where $\hat{k}$ is defined as

$$\hat{k} = \left(\overline{1/\sqrt{k}}\right)^{-2}\,. \tag{2.25}$$

The shallow water correction of Eq. (2.21) is valid for intermediate depths only. For this reason the mean relative depth $\bar{k}d$ is not allowed to become smaller than 0.5 (as in WAM). All above constants can be reset by the user in the input files of the model (see chapter 4).

2.3.3 Nonlinear interactions (WRT) (G. Ph. van Vledder)

The second method for calculating the nonlinear interactions in WWATCH is the so-called Webb-Resio-Tracy method (WRT), which is based on the original six-dimensional Boltzmann integral formulation of Hasselmann (1962, 1963a,b), and additional considerations by Webb (1978), Tracy and Resio (1982) and Resio and Perrie (1991).

The Boltzmann integral describes the rate of change of action density of a particular wavenumber due to resonant interactions between pairs of four wavenumbers. To interact these wavenumbers must satisfy the following resonance conditions

$$\left.\begin{array}{rcl} \mathbf{k}_1 + \mathbf{k}_2 & = & \mathbf{k}_3 + \mathbf{k}_4 \\ \sigma_1 + \sigma_2 & = & \sigma_3 + \sigma_4 \end{array}\right\} \quad , \tag{2.26}$$

which is a more general version of the resonance conditions (2.19). The rate of change of action density N_1 at wave number $\mathbf{k}_1$ due to all quadruplet interactions involving $\mathbf{k}_1$ is given by

$$\begin{aligned} \frac{\partial N_1}{\partial t} &= \iiint G\left(\mathbf{k}_1,\mathbf{k}_2,\mathbf{k}_3,\mathbf{k}_4\right)\, \delta\left(\mathbf{k}_1+\mathbf{k}_2-\mathbf{k}_3-\mathbf{k}_4\right)\, \delta\left(\sigma_1+\sigma_2-\sigma_3-\sigma_4\right) \\ &\times \left[N_1 N_3\left(N_4-N_2\right)+N_2 N_4\left(N_3-N_1\right)\right]\, d\mathbf{k}_2\, d\mathbf{k}_3\, d\mathbf{k}_4 \quad , \end{aligned} \tag{2.27}$$

where the action density N is defined in terms of the wavenumber vector $\mathbf{k}$, $N = N(\mathbf{k})$. The term G is a complicated coupling coefficients for which expressions have been given by Herterich and Hasselmann (1980). In the WRT method a number of transformations are made to remove the delta functions. A key element in the WRT method is to consider the integration space for each $(\mathbf{k}_1, \mathbf{k}_3)$ combination (see Resio and Perrie, 1991)

$$\frac{\partial N_1}{\partial t} = 2 \int T\left(\mathbf{k}_1,\mathbf{k}_3\right)\, d\mathbf{k}_3 \quad , \tag{2.28}$$

in which the function T is given by

$$\begin{aligned} T\left(\mathbf{k}_1,\mathbf{k}_3\right) &= \iint G\left(\mathbf{k}_1,\mathbf{k}_2,\mathbf{k}_3,\mathbf{k}_4\right)\, \delta\left(\mathbf{k}_1+\mathbf{k}_2-\mathbf{k}_3-\mathbf{k}_4\right) \\ &\times \delta\left(\sigma_1+\sigma_2-\sigma_3-\sigma_4\right)\, \theta\left(\mathbf{k}_1,\mathbf{k}_3,\mathbf{k}_4\right) \\ &\times \left[N_1 N_3\left(N_4-N_2\right)+N_2 N_4\left(N_3-N_1\right)\right]\, d\mathbf{k}_2\, d\mathbf{k}_4 \quad , \end{aligned} \tag{2.29}$$

in which

$$\theta\left(\mathbf{k}_1,\mathbf{k}_3,\mathbf{k}_4\right) = \left\{\begin{array}{lll} 1 & \text{when} & \left|\mathbf{k}_1-\mathbf{k}_3\right| \le \left|\mathbf{k}_1-\mathbf{k}_4\right| \\ 0 & \text{when} & \left|\mathbf{k}_1-\mathbf{k}_3\right| > \left|\mathbf{k}_1-\mathbf{k}_4\right| \end{array}\right. \tag{2.30}$$

The delta functions in Eq. (2.29) determine a region in wavenumber space along which the integration should be carried out. The function θ determines

a section of the integral which is not defined due to the assumption that $\mathbf{k}_1$ is closer to $\mathbf{k}_3$ than $\mathbf{k}_2$. The crux of the Webb method consists of using a local coordinate system along a so-named locus, that is, the path in $\mathbf{k}$ space given by the resonance conditions for a given combination of $\mathbf{k}_1$ and $\mathbf{k}_3$. To that end the (k_x, k_y) coordinate system is replaced by a (s, n) coordinate system, where s (n) is the tangential (normal) direction along the locus. After some transformations the transfer integral can then be written as a closed line integral along the closed locus

$$\begin{aligned} T\left(\mathbf{k}_1,\mathbf{k}_3\right) &= \oint G \left|\frac{\partial W(s,n)}{\partial n}\right|^{-1} \theta(\mathbf{k}_1,\mathbf{k}_3,\mathbf{k}_4) \\ &\times \left[N_1 N_3 \left(N_4 - N_2\right) + N_2 N_4 \left(N_3 - N_1\right)\right] \, ds \quad , \end{aligned} \tag{2.31}$$

In which G is the coupling coefficient and $|\partial W/\partial n|$ is the gradient term of a function representing the resonance conditions (see Van Vledder, 2000). Numerically, the Boltzmann integral is computed as the finite sum of many line integrals T for all discrete combinations of $\mathbf{k}_1$ and $\mathbf{k}_3$. The line integral (2.31) is solved by dividing the locus in typically 30 pieces, such that the discretized version is given as:

$$T\left(\mathbf{k}_1,\mathbf{k}_3\right) \approx \sum_{i=1}^{n_s} G(s_i) W(s_i) P(s_i) \, \Delta s_i \quad , \tag{2.32}$$

in which $P(s_i)$ is the product term for a given point on the locus, n_s is the number of segments, and s_i is the discrete coordinate along the locus. Finally, the rate of change for a given wavenumber $\mathbf{k}_1$ is given by

$$\frac{\partial N(\mathbf{k}_1)}{\partial t} \approx \sum_{i_{k3}=1}^{n_k} \sum_{i_{\theta 3}=1}^{n_\theta} k_3 T(\mathbf{k}_1,\mathbf{k}_3) \, \Delta k_{i_{k3}} \, \Delta\theta_{i_{\theta 3}} \tag{2.33}$$

where n_k and n_θ are the discrete number of wavenumbers and directions in the calculational grid, respectively. Note that although the spectrum is defined in terms of the vector wavenumber $\mathbf{k}$, the calculational grid in a wave model is more conveniently defined in terms of the absolute wavenumber and wave direction (k, θ) to assure directional isotropy of the calculations. Taking all wave numbers $\mathbf{k}_1$ into account produces the complete source term due to nonlinear quadruplet wave-wave interactions. Details of the efficient

computation of a locus for a given combination of the wave numbers $\mathbf{k}_1$ and $\mathbf{k}_3$ can be found in Van Vledder (2000, 2002a,b).

It should be noted that these exact interaction calculations are extremely expensive, typically requiring 10^3 to 10^4 times more computational effort than the DIA. Presently, these calculations can therefore only be made for highly idealized test cases involving a limited spatial grid.

The nonlinear interactions according to the WRT method have been implemented in WWATCH using the portable subroutines developed by Van Vledder (2002b). In this implementation, the computational grid of the WRT method is taken identical to the discrete spectral grid of WWATCH. In addition, the WRT routines inherit the power of the parametric spectral tail as in the DIA. Choosing a higher resolution than the computational grid of WWATCH for computing the nonlinear interactions is possible in theory, but this does not improve the results and is therefore not implemented.

Because nonlinear quadruplet wave-wave interactions at high frequencies are important, it is recommended to choose the maximum frequency of the wave model about five times the peak frequency of the spectra that are expected to occur in a wave model run. Note that this is important as the spectral grid determines the range of integration in Eq. (2.33). The recommended number of frequencies is about 40, with a frequency increment factor 1.07. The recommended directional resolution for computing the nonlinear interactions is about 10°. For specific purposes other resolutions may be used, and some testing with other resolutions may be needed.

An important feature of most algorithms for the evaluation of the Boltzmann integral is that the integration space can be pre-computed. This is also the case for the subroutine version of the WRT method used in WWATCH. In the initialization phase of the wave model the integration space, consisting of the discretized paths of all loci, together with the interaction coefficients and gradient terms, are computed and stored in a binary data file. For each water depth such a data file is generated and stored in the current directory. The names of these data files consist of a keyword, "quad", followed by the keyword "*xxxx*", with *xxxx* the water depth in meters, or 9999 for deep water. The extension of the binary data file is "bqf" (of Binary Quadruplet File). If a BQF file exists, the program checks if this BQF file has been generated with the proper spectral grid. If this is not the case, the existing BQF file is overwritten with the correct BQF file. During a wave model run with various depths, the optimal BQF is used, by looking at the nearest water depths for which a valid BQF file has been generated. In addition, the result is rescaled

using the ratio of the depth scaling factors (2.21) for the target depth and the depth corresponding to the BQF file.

2.3.4 Input and dissipation (WAM-3)

The input and dissipation source terms of WAM cycles 1 through 3 are based on Snyder et al. (1981) and Komen et al. (1984) (see also WAMDIG, 1988). The input source term is given as

$$\mathcal{S}_{in}(k,\theta) = C_{in}\frac{\rho_a}{\rho_w}\max\left[0,\left(\frac{28\,u_*}{c}\cos(\theta-\theta_w)-1\right)\right]\,\sigma\,N(k,\theta)\;, \tag{2.34}$$

$$u_* = u_{10}\sqrt{(0.8+0.065u_{10})10^{-3}}\,, \tag{2.35}$$

where C_{in} is a constant ($C_{in}=0.25$), ρ_a (ρ_w) is the density of air (water), u_* is the wind friction velocity (Charnock, 1955; Wu, 1982), c is the phase velocity σ/k, u_{10} is the wind speed at 10 m above the mean sea level and θ_w is the mean wind direction. The corresponding dissipation term is given as

$$\mathcal{S}_{ds}(k,\theta) = C_{ds}\,\hat{\sigma}\,\frac{k}{\hat{k}}\left(\frac{\hat{\alpha}}{\hat{\alpha}_{PM}}\right)^2 N(k,\theta)\;, \tag{2.36}$$

$$\hat{\sigma} = \left(\overline{\sigma^{-1}}\right)^{-1}\;, \tag{2.37}$$

$$\hat{\alpha} = E\,\hat{k}^2\,g^{-2}\;, \tag{2.38}$$

where C_{ds} is a constant ($C_{ds}=-2.36\,10^{-5}$), $\hat{\alpha}_{PM}$ is the value of $\hat{\alpha}$ for a PM spectrum ($\hat{\alpha}_{PM}=3.02\,10^{-3}$) and where $\hat{k}$ is given by Eq. (2.25).

The parametric tail [Eqs. (2.17) and (2.18)] corresponding to these source terms is given by[3] $m=4.5$ and by

$$f_{hf} = \max\left[\,2.5\,\hat{f}_r\;,\;4\,f_{r,PM}\,\right]\;, \tag{2.39}$$

$$f_{r,PM} = \frac{g}{28\,u_*}\,, \tag{2.40}$$

[3] originally, WAM used $m=5$, present setting used for consistent limit behavior (e.g., Tolman, 1992).

where $f_{r,PM}$ is the Pierson and Moskowitz (1964) frequency, estimated from the wind friction velocity u_*. The shape and attachment point of this tail is hardwired to the present model. The tunable parameters C_{in}, C_{ds} and α_{PM} are preset to their default values, but can be redefined by the user in the input files of the model.

2.3.5 Input and dissipation (Tolman and Chalikov)

The source term package of Tolman and Chalikov (1996) consists of the input source term of Chalikov and Belevich (1993) and Chalikov (1995), and two dissipation constituents. The input source term is given as

$$\mathcal{S}_{in}(k,\theta) = \sigma\,\beta\,N(k,\theta)\,, \tag{2.41}$$

where β is a nondimensional wind-wave interaction parameter, which is approximated as

$$10^4\beta = \begin{cases} -a_1\tilde{\sigma}_a^2 - a_2 & , \quad & \tilde{\sigma}_a & \leq -1 \\ a_3\tilde{\sigma}_a(a_4\tilde{\sigma}_a - a_5) - a_6 & , \quad -1 < & \tilde{\sigma}_a & < \Omega_1/2 \\ (a_4\tilde{\sigma}_a - a_5)\tilde{\sigma}_a & , \quad \Omega_1/2 < & \tilde{\sigma}_a & < \Omega_1 \\ a_7\tilde{\sigma}_a - a_8 & , \quad \Omega_1 < & \tilde{\sigma}_a & < \Omega_2 \\ a_9(\tilde{\sigma}_a - 1)^2 + a_{10} & , \quad \Omega_2 < & \tilde{\sigma}_a & \end{cases} \tag{2.42}$$

where

$$\tilde{\sigma}_a = \frac{\sigma\,u_\lambda}{g}\cos(\theta - \theta_w) \tag{2.43}$$

is the non-dimensional frequency of a spectral component, θ_w is the wind direction and u_λ is the wind velocity at a height equal to the 'apparent' wave length

$$\lambda_a = \frac{2\pi}{k|\cos(\theta - \theta_w)|}\,. \tag{2.44}$$

The parameters $a_1 - a_{10}$ and Ω_1, Ω_2 in Eq. (2.42) depend on the drag coefficient C_λ at the height $z = \lambda_a$:

$$
\begin{array}{ll}
\Omega_1 = 1.075 + 75C_\lambda & \Omega_2 = \ 1.2 + 300C_\lambda \\
a_1 = 0.25 + 395C_\lambda, & a_3 = \ (a_0 - a_2 - a_1)/(a_0 - a_4 + a_5) \\
a_2 = 0.35 + 150C_\lambda, & a_5 = \ a_4\Omega_1 \\
a_4 = 0.30 + 300C_\lambda, & a_6 = \ a_0(1 - a_3) \\
a_9 = 0.35 + 240C_\lambda, & a_7 = \ (a_9(\Omega_2 - 1)^2 + a_{10})/(\Omega_2 - \Omega_1) \\
a_{10} = -0.05 + 470C_\lambda, & a_8 = \ a_7\Omega_1 \\
 & a_0 = \ 0.25a_5^2/a_4
\end{array}
\tag{2.45}
$$

The wave model takes the wind u_r at a given reference height z_r as its input, so that u_λ and C_λ need to be derived as part of the parameterization. Excluding a thin surface layer adjusting to the water surface, the mean wind profile is close to logarithmic

$$u_z = \frac{v_*}{\kappa} \ln\left(\frac{z}{z_0}\right) \ , \tag{2.46}$$

where $\kappa = 0.4$ is the Von Kàrmàn constant, and z_0 is the roughness parameter. This equation can be rewritten in terms of the drag coefficient C_r at the reference height z_r as (Chalikov, 1995)

$$C_r = \kappa^2 \left[R - \ln(C)\right]^2 \ , \tag{2.47}$$

where

$$R = \ln\left(\frac{z_r g}{\chi\sqrt{\alpha}u_r^2}\right) \ , \tag{2.48}$$

where $\chi = 0.2$ is a constant, and where α is the conventional nondimensional energy level at high frequencies. An accurate explicit approximation to these implicit relations is given as

$$C_r = 10^{-3}\left(0.021 + \frac{10.4}{R^{1.23} + 1.85}\right) \ . \tag{2.49}$$

The estimation of the drag coefficient thus requires an estimate of the high-frequency energy level α, which could be estimated directly from the wave model. However, the corresponding part of the spectrum is generally not well resolved, tends to be noisy, and is tainted by errors in several source terms. Therefore, α is estimated parametrically as (Janssen, 1989)

$$\alpha = 0.57\left(\frac{u_*}{c_p}\right)^{3/2} \ . \tag{2.50}$$

As the latter equation depends on the drag coefficient, Eqs. (2.48) through (2.50) formally need to be solved iteratively. Such iterations are performed during the model initialization, but are not necessary during the actual model run, as u_* generally changes slowly. Note that Eq. (2.50) can be considered as an internal relation to the parameterization of C_r, and can therefore deviate from actual model behavior without loss of generality. In Tolman and Chalikov (1996), C_r is therefore expressed directly in terms of c_p.

Using the definition of the drag coefficient and Eq. (2.46) the roughness parameter z_0 becomes

$$z_0 = z_r \ \exp\left(-\kappa C_r^{-1/2}\right) \ , \tag{2.51}$$

and the wind velocity and drag coefficient at height λ become

$$u_\lambda = u_r \frac{ln(\lambda_a/z_0)}{ln(z_r/z_0)} \tag{2.52}$$

$$C_\lambda = C_r \left(\frac{u_a}{u_\lambda}\right)^2 \tag{2.53}$$

Finally, Eq. (2.50) requires an estimate for the peak frequency f_p. To obtain a consistent estimate of the peak frequency of actively generated waves, even in complex multimodal spectra, this frequency is estimated from the equivalent peak frequency of the positive part of the input source term (see Tolman and Chalikov, 1996)

$$f_{p,i} = \frac{\int\int \ f^{-2} \, c_g^{-1} \ \max\left[\, 0 \, , \ \mathcal{S}_{wind}(k,\theta) \,\right] \, df \ d\theta}{\int\int \ f^{-3} \, c_g^{-1} \ \max\left[\, 0 \, , \ \mathcal{S}_{wind}(k,\theta) \,\right] \, df \ d\theta} \ , \tag{2.54}$$

from which the actual peak frequency is estimated as (the tilde identifies nondimensional parameter based on u_* and g)

$$\tilde{f}_p \ = \ 3.6 \, 10^{-4} \ + \ 0.92 \tilde{f}_{p,i} \ - \ 6.3 \, 10^{-10} \ \tilde{f}_{p,i}^{-3} \ . \tag{2.55}$$

All constants in the above equations are defined within the model. The user only defines the reference wind height z_r.

During testing of a global implementation of WWATCH including this source term (Tolman, 2002f), it was found that its swell dissipation due to opposing or weak winds was severely overestimated. To correct this deficiency, a filtered input source term is defined as

$$\mathcal{S}_{i,m} = \left\{ \begin{array}{lllll} \mathcal{S}_i & \text{for} & \beta \geq 0 & \text{or} & f > 0.8 f_p \\ X_s \mathcal{S}_i & \text{for} & \beta < 0 & \text{and} & f < 0.6 f_p \\ \mathcal{X}_s \mathcal{S}_i & \text{for} & \beta < 0 & \text{and} & 06 f_p < f < 0.8 f_p \end{array} \right. , \tag{2.56}$$

where f is the frequency, f_p is the peak frequency of the wind sea as computed from the input source term, $\mathcal{S}_i$ is the input source term (2.41), and $0 < X_s < 1$ is a reduction factor for $\mathcal{S}_i$, which is applied to swell with negative β only (defined by the user). $\mathcal{X}_s$ represents a linear reduction of X_s with f_p providing a smooth transition between the original and reduced input.

The corresponding dissipation source term consists of two constituents. The (dominant) low-frequency constituent is based on an analogy with energy dissipation due to turbulence,

$$\mathcal{S}_{ds,l}(k,\theta) = -2\, u_*\, h\, k^2 \phi\, N(k,\theta) \,, \tag{2.57}$$

$$h = 4 \left(\int_0^{2\pi} \int_{f_h}^{\infty} F(f,\theta)\, df\, d\theta \right)^{1/2} . \tag{2.58}$$

$$\phi = b_0 + b_1 \tilde{f}_{p,i} + b_2 \tilde{f}_{p,i}^{-b_3} \,. \tag{2.59}$$

where h is a mixing scale determined from the high-frequency energy content of the wave field and where ϕ is an empirical function accounting for the development stage of the wave field. The linear part of Eq. (2.59) describes dissipation for growing waves. The nonlinear term has been added to allow for some control over fully grown conditions by defining a minimum value for ϕ ($\phi_{\min}$) for a minimum value of $f_{p,i}$ ($f_{p,i,\min}$). If $\phi_{\min}$ is below the linear curve, b_2 and b_3 are given as

$$b_2 = \tilde{f}_{p,i,\min}^{b_3} \left(\phi_{\min} - b_0 - b_1 \tilde{f}_{p,i,\min} \right) , \tag{2.60}$$

$$b_3 = 8 \,. \tag{2.61}$$

If $\phi_{\min}$ is above the linear curve, b_2 and b_3 are given as

$$\tilde{f}_a = \frac{\phi_{\min} - b_0}{b_1} \quad , \quad \tilde{f}_b = \max \left\{ \tilde{f}_a - 0.0025 \,,\, \tilde{f}_{p,i,\min} \right\} \quad , \tag{2.62}$$

$$b_2 = \tilde{f}_b^{b_3} \left[\phi_{\min} - b_0 - b_1 \tilde{f}_b \right] , \tag{2.63}$$

$$b_3 = \frac{b_1 \tilde{f}_b}{\phi_{\min} - b_0 - b_1 \tilde{f}_b} \, . \tag{2.64}$$

The above estimate of b_3 results in $\partial\phi/\partial\tilde{f}_{p,i} = 0$ for $\tilde{f}_{p,i} = \tilde{f}_b$. For $\tilde{f}_{p,i} < \tilde{f}_b$, ϕ is kept constant ($\phi = \phi_{\min}$).

The empirical high-frequency dissipation is defined as

$$\mathcal{S}_{ds,h}(k,\theta) = -a_0 \left(\frac{u_*}{g}\right)^2 f^3 \, \alpha_n^B \, N(k,\theta) \quad , \tag{2.65}$$

$$B = a_1 \left(\frac{f u_*}{g}\right)^{-a_2} \quad ,$$

$$\alpha_n = \frac{\sigma^6}{c_g \, g^2 \alpha_r} \int_0^{2\pi} N(k,\theta) \, d\theta \quad , \tag{2.66}$$

where α_n is Phillips' nondimensional high-frequency energy level normalized with α_r, and where a_0 through a_2 and α_r are empirical constants. This parameterization implies that $m = 5$ in the parametric tail, which has been preset in the model. Note that in the model Eq. (2.66) is solved assuming a deep water dispersion relation, in which case α_n is evaluated as

$$\alpha_n = \frac{2 \, k^3}{\alpha_r} F(k) \quad . \tag{2.67}$$

The two constituents of the dissipation source term are combined using a simple linear combination, defined by the frequencies f_1 and f_2.

$$\mathcal{S}_{ds}(k,\theta) = \mathcal{A}\mathcal{S}_{ds,l} + (1-\mathcal{A}) \, \mathcal{S}_{ds,h} \, , \tag{2.68}$$

$$\mathcal{A} = \begin{cases} 1 & \text{for} \quad f < f_l \quad , \\ \frac{f-f_2}{f_1-f_2} & \text{for} \quad f_1 \le f < f_2 \quad , \\ 0 & \text{for} \quad f_2 \le f \quad , \end{cases} \tag{2.69}$$

To enhance the smoothness of the model behavior for frequencies near the parametric cut-off f_{hf}, a similar transition zone is used between the prognostic spectrum and the parametric high-frequency tail as in Eq. (2.18)

$$N(k_i,\theta) = (1-\mathcal{B}) \, N(k_i,\theta) + \mathcal{B} N(k_{i-1},\theta) \left(\frac{f_i}{f_{i-1}}\right)^{-m-2} \quad , \tag{2.70}$$

Tuned to :	a_0	a_1	a_2	b_0	b_1	$\phi_{\min}$
KC stable	4.8	$1.7\,10^{-4}$	2.0	$0.3\,10^{-3}$	0.47	0.003
KC unstable	4.5	$2.3\,10^{-3}$	1.5	$-5.8\,10^{-3}$	0.60	0.003

Table 2.2: Suggested constants in the source term package of Tolman and Chalikov. KC denotes Kahma and Calkoen (1992, 1994). First line represents default model settings.

---

where i is a discrete wavenumber counter, and where $\mathcal{B}$ is defined similarly to $\mathcal{A}$, ranging from 0 to 1 between f_2 and f_{hf}.

The frequencies defining the transitions and the length scale h are predefined in the model as

$$\left.\begin{array}{rcl} f_{hf} & = & 3.00\, f_{p,i} \\ f_1 & = & 1.75\, f_{p,i} \\ f_2 & = & 2.50\, f_{p,i} \\ f_h & = & 2.00\, f_{p,i} \end{array}\right\} \quad . \tag{2.71}$$

Furthermore, $f_{p,i,\min} = 0.009$ and $\alpha_r = 0.002$ are preset in the model. All other tunable parameters have to be provided by the user. Suggested and default values are given in Table 2.2.

Test results of these source terms in a global model implementation (Tolman, 2002f) suggested that (i) the model tuned in the classical way to fetch-limited growth for stable conditions underestimates deep-ocean wave growth (a deficiency apparently shared by the WAM model) and that (ii) effects of stability on the growth rate of waves as identified by Kahma and Calkoen (1992, 1994) should be included explicitly in the parameterization of the source terms. Ideally, both problems would be dealt with by theoretical investigation of the source terms. Alternatively, the wind speed u can be replaced by an effective wind speed u_e. In Tolman (2002f) the following effective wind speed is used :

$$\frac{u_e}{u} = \left(\frac{c_o}{1 + C_1 + C_2} \right)^{-1/2} , \tag{2.72}$$

$$C_1 = c_1 \tanh \left[\max(0, f_1\{\mathcal{ST} - \mathcal{ST}_o\}) \right] , \tag{2.73}$$

$$C_2 = c_2 \tanh \left[\max(0, f_2\{\mathcal{ST} - \mathcal{ST}_o\}) \right] , \tag{2.74}$$

$$\mathcal{ST} = \frac{hg}{u_h^2} \frac{T_a - T_s}{T_0} \,, \tag{2.75}$$

where $\mathcal{ST}$ is a bulk stability parameter, and T_a, T_s and T_0 are the air, sea and reference temperature, respectively. Furthermore, $f_1 \leq 0$, c_1 and c_2 have opposite signs and $f_2 = f_1 c_1 / c_2$. Following Tolman (2002f), default settings of $c_0 = 1.4$, $c_1 = -0.1$, $c_2 = 0.1$, $f_1 = -150$ and $\mathcal{ST}_o = -0.01$ in combination with the tuning to stable stratification wave growth data ('KC stable' parameter values in Table 2.2) are used. Note that this effective wind speed was derived for winds at 10 m height. The wind correction can be switched off by the user during compilation of the model, and default parameter settings can be redefined by the user in the program input files.

2.3.6 Bottom friction (JONSWAP)

A simple parameterization of bottom friction is the empirical, linear JONSWAP parameterization (Hasselmann et al., 1973), as used in the WAM model (WAMDIG, 1988). Using the notation of Tolman (1991), this source term can be written as

$$\mathcal{S}_{bot}(k,\theta) = 2\Gamma \frac{n - 0.5}{gd} N(k,\theta) \,, \tag{2.76}$$

where Γ is an empirical constant, which is estimated as $\Gamma = -0.038\ \mathrm{m^2 s^{-3}}$ for swell (Hasselmann et al., 1973), and as $\Gamma = -0.067\ \mathrm{m^2 s^{-3}}$ for wind seas (Bouws and Komen, 1983). n is the ratio of phase velocity to group velocity given by (2.6). The default value for $\Gamma = -0.067$ can be redefined by the user in the model input files.

2.4 Output parameters

The wave model provides output of the following gridded fields of mean wave parameters. Some of these parameters can also be found in the output for selected points. For activation of the output see section 4.4.5

1) The mean water depth (m).
2) The mean current velocity (vector, m/s).

3) The mean wind speed (vector, m/s). This wind speed is alway the speed as input to the model, i.e., is not corrected for the current speed.
4) The air-sea temperature difference (°C).
5) The friction velocity u_* (scalar). Definition depends on selected source term parameterization (m/s).
6) Significant wave height (m) [see Eq. (2.23)]

$$H_s = 4\sqrt{E} \,. \tag{2.77}$$

7) Mean wave length (m) [see Eq. (2.22)]

$$L_m = 2\pi\overline{k^{-1}} \,. \tag{2.78}$$

8) Mean wave period (s)

$$T_m = 2\pi\overline{\sigma^{-1}} \,. \tag{2.79}$$

9) Mean wave direction (degr., meteorological convention)

$$\theta_m = \mathrm{atan}\left(\frac{b}{a}\right) \,, \tag{2.80}$$

$$a = \int_0^{2\pi}\int_0^{\infty} \cos(\theta) F(\sigma,\theta)\, d\sigma\, d\theta \,, \tag{2.81}$$

$$b = \int_0^{2\pi}\int_0^{\infty} \sin(\theta) F(\sigma,\theta)\, d\sigma\, d\theta \,. \tag{2.82}$$

10) Mean directional spread (degr.; Kuik et al., 1988)

$$\sigma_\theta = \left[2\left\{1 - \left(\frac{a^2+b^2}{E^2}\right)^{1/2}\right\}\right]^{1/2} \,, \tag{2.83}$$

11) Peak frequency (Hz), calculated from the one-dimensional frequency spectrum using a parabolic fit around the discrete peak.
12) Peak direction (degr.), defined like the mean direction, using the frequency/wavenumber bin containing of the spectrum $F(k)$ that contains the peak frequency only.

13) Peak frequency of the wind sea part of the spectrum. For WAM-3 input, this is the highest local peak in the one-dimensional spectrum, if this frequency is higher than half the PM frequency and smaller than 0.75 times the maximum discrete frequency (otherwise undefined). For the Tolman and Chalikov input, it is calculated using Eqs. (2.54) and (2.55).
14) Wind sea direction (degr.), defined like the mean direction, using the frequency/wavenumber bin containing the peak frequency of the wind sea only.
15) Average time step in the source term integration (s).
16) Cut-off frequency f_c (Hz, depends on parameterization of input and dissipation).
17) Ice concentration.
18) Water level.

The first 16 output parameter were available in the first WWATCH release. Output parameters 17 and higher have been added since. All new output parameters are simply added to the end of the list to make conversion of input files to the new system as easy as possible.

3 Numerical approaches

3.1 Basic concepts

Equation (2.8) or (2.12) represents the basic equation of WWATCH. However, modified versions of these equations are used in the model, where (a) they are solved on a variable wavenumber grid (see below), where (b) a modified versions of these equations are used to properly described dispersion for discretized equations in selected numerical schemes (see section 3.3), and where (c) subgrid obstacles such as islands are considered (see section 3.3).

If (2.8) or (2.12) is solved directly, an effective reduction of spectral resolution occurs in shallow water (see Tolman and Booij, 1998). This loss of resolution can be avoided if the equation is solved on a variable wavenumber grid, which implicitly incorporates the kinematic wavenumber changes due to shoaling. Such a wavenumber grid corresponds to a spatially and temporally invariant frequency grid (Tolman and Booij, 1998). The corresponding local wavenumber grid can be calculated directly from the invariant frequency grid and the dispersion relation (2.1), and hence becomes a function of the local depth d.To accommodate economical calculations of S_{nl}, a logarithmic frequency grid is adopted,

$$\sigma_{m+1} = X_\sigma \, \sigma_m \, , \tag{3.1}$$

where m is a discrete grid counter in k-space. X_σ is defined by the user in the input files of the program. In most applications of third-generation models $X_\sigma = 1.1$ is used (WWATCH default)).

The effects of a spatially varying grid will be discussed for the Cartesian equation (2.8) only. Adaptation to the spherical grid is trivial. Denoting the variable wavenumber grid with κ, the balance equation becomes

$$\frac{\partial}{\partial t}\frac{N}{c_g} + \frac{\partial}{\partial x}\frac{\dot{x}N}{c_g} + \frac{\partial}{\partial y}\frac{\dot{y}N}{c_g} + \frac{\partial}{\partial \kappa}\frac{\dot{\kappa}N}{c_g} + \frac{\partial}{\partial \theta}\frac{\dot{\theta}N}{c_g} = 0 \, , \tag{3.2}$$

$$\dot{\kappa}\,\frac{\partial k}{\partial \kappa} = c_g^{-1}\frac{\partial \sigma}{\partial d}\left(\frac{\partial d}{\partial t} + \mathbf{U}\cdot\nabla_x d\right) - \mathbf{k}\cdot\frac{\partial \mathbf{U}}{\partial s} \, . \tag{3.3}$$

Equation (3.2) is solved using a fractional step method, as is commonplace in wave modeling. The first step considers temporal variations of the depth,

and corresponding changes in the wavenumber grid. As is discussed by Tolman and Booij (1998), this step can be invoked sparsely. By splitting of effects of (temporal) water level variations, the grid becomes invariant, and the depth becomes quasi-steady for the remaining fractional steps. Other fractional steps consider spatial propagation, intra-spectral propagation and source terms.

The multiple splitting technique results in a model that can efficiently be vectorized and parallelized at the same time. The time splitting furthermore allows for the use of separate partial or dynamically adjusted time steps in the different fractional steps of the model. WWATCH makes a distinction between 4 different time steps.

1) The 'global' time step Δt_g, by which the entire solution is propagated in time, and at which intervals input winds and currents are interpolated. This time step is provided by the user, but can be reduced within the model to reach a requested input or output time.
2) The second time step is the time step for spatial propagation. The user supplies the maximum propagation time step for the lowest model frequency $\Delta t_{p,1}$. For the frequency with counter m, the maximum time step $\Delta t_{p,m}$ is calculated within the model as

$$\Delta t_{p,m} = \frac{f_m}{f_1} \Delta t_{p,1} \; . \tag{3.4}$$

   If the propagation time step is smaller than the global time step, the propagation effects are calculated with a number of successive smaller time steps. This generally implies that several partial time steps are used for the lowest frequency, but that the highest frequencies are propagated over the interval Δt_g with a single calculation. The latter results in a significantly more efficient model, particularly if higher-order accurate propagation schemes are used.
3) The third time step is the time step for intra-spectral propagation. For large-scale and deep-water grids this time step can generally be taken equal to the global time step Δt_g. For shallow water grids, smaller intra-spectral propagation time steps allow for larger effects of refraction within the stability constraints of the scheme. Note that the order of invoking spatial and intra-spectral propagation is alternated to enhance numerical accuracy.

4) The final time step is the time step for the integration of the source terms, which is dynamically adjusted for each separate grid point and global time step Δt_g (see section 3.5). This results in more accurate calculations for rapidly changing wind and wave conditions, and a more economical integration for slowly varying conditions.

The following sections deal with the separate steps in the fractional step method, model input, ice treatment and boundary data transfer between separate model runs.

3.2 Depth variations in time

Temporal depth variations result in a change of the local wavenumber grid. Because the wavenumber spectrum is invariant with respect to temporal changes of the depth, this corresponds to a simple interpolation of the spectrum from the old grid to the new grid, without changes in the spectral shape. As discussed above, the new grid simply follows from the globally invariant frequency grid, the new water depth d and the dispersion relation (2.1). The time step of updating the water level is generally dictated by physical time scales of water level variations, but not by numerical considerations (Tolman and Booij, 1998).

The interpolation to the new wavenumber grid is performed with a simple conservative interpolation method. In this interpolation the old spectrum is first converted to discrete action densities by multiplication with the spectral bin widths. This discrete action then is redistributed over the new grid cf. a regular linear interpolation. The new discrete actions then are converted into a spectrum by division by the (new) spectral bin widths. The conversion requires a parametric extension of the original spectrum at high and low frequencies because the old grid generally will not completely cover the new grid. Energy/action in the old spectrum at low wavenumbers that are not resolved by the new grid is simply removed. At low wavenumbers in the new grid that are not resolved by the old grid zero energy/action is assumed. At high wavenumbers in the new grid the usual parametric tail is applied if necessary. The latter correction is rare, as the highest wavenumbers usually correspond to deep water.

In practical applications the grid modification is usually relevant for a small fraction of the grid points only. To avoid unnecessary calculations, the grid is transformed only if the smallest relative depth kd in the discrete spectrum is smaller than 4. Furthermore, the spectrum is interpolated only if the spatial grid point is not covered by ice, and if the largest change of wavenumber is at least $0.05\Delta k$.

3.3 Spatial propagation

Spatial propagation is described by the first terms of Eq. (3.2). For the spherical grid [Eq. (2.12)], the corresponding spatial propagation step becomes

$$\frac{\partial \mathcal{N}}{\partial t} + \frac{\partial}{\partial \phi}\,\dot{\phi}\mathcal{N} + \frac{\partial}{\partial \lambda}\,\dot{\lambda}\mathcal{N} = 0\,, \tag{3.5}$$

where the propagated quantity $\mathcal{N}$ is defined as $\mathcal{N} \equiv N\,c_g^{-1}\cos\phi$. For the Cartesian grid, a similar equation is found propagating $\mathcal{N} \equiv N\,c_g^{-1}$. In this section equations for the more complicated spherical grid are presented only. Conversion to a Cartesian grid is generally a simplification and is trivial.

Equation (3.5) in form is identical to the conventional deep-water propagation equation, but includes effects of both limited depths and currents. At the land-sea boundaries, wave action propagating towards the land is assumed to be absorbed without reflection, and waves propagating away from the coast are assumed to have no energy at the coastline. For so-called 'active boundary points' where boundary conditions are prescribed, a similar approach is used. Action traveling towards such points is absorbed, whereas action at the boundary points is used to estimate action fluxes for components traveling into the model.

Three separate issues arise regarding the spatial propagation. The first is the actual propagation scheme used (section 3.3.1). The second is the occurrence and alleviation of the Garden Sprinkler Effect (GSE) as discussed in section 3.3.2. The third is the inclusion of the effects of unresolved obstacles on wave propagation. Subgrid treatment of such obstacles is addressed in WWATCH as part of the numerical propagation scheme, and is discussed in section 3.3.3.

3.3.1 Propagation schemes

A simple and cheap first order upwind scheme has been included, mainly for testing during development of WWATCH. To assure numerical conservation of action, a flux or control volume formulation is used. The flux between grid points with counters i and $i-1$ in ϕ-space ($\mathcal{F}_{i,-}$) is calculated as

$$\mathcal{F}_{i,-} = \left[\dot{\phi}_b \, \mathcal{N}_u \right]^n_{j,l,m} \; , \tag{3.6}$$

$$\dot{\phi}_b = 0.5 \, \left(\dot{\phi}_{i-1} + \dot{\phi}_i \right)_{j,l,m} \; , \tag{3.7}$$

$$\mathcal{N}_u = \left\{ \begin{array}{ll} \mathcal{N}_{i-1} & \mathrm{for} \quad \dot{\phi}_b \geq 0 \\ \mathcal{N}_i & \mathrm{for} \quad \dot{\phi}_b < 0 \end{array} \right. \; , \tag{3.8}$$

where j, l and m are discrete grid counters in λ-, θ- and k-spaces, respectively, and n is a discrete time step counter. $\dot{\phi}_b$ represents the propagation velocity at the 'cell boundary' between points i and $i-1$, and the subscript u denotes the 'upstream' grid point. At land-sea boundaries, $\dot{\phi}_b$ is replaced by $\dot{\phi}$ at the sea point. Fluxes between points i and $i+1$ ($\mathcal{F}_{i,+}$) are obtained by replacing $i-1$ with i and i with $i+1$. Fluxes in λ-space are calculated similarly, changing the appropriate grid counters and increments. The 'action density' ($\mathcal{N}^{n+1}$) at time $n+1$ is estimated as

$$\mathcal{N}^{n+1}_{i,j,l,m} = \mathcal{N}^n_{i,j,l,m} + \frac{\Delta t}{\Delta \phi} \left[\mathcal{F}_{i,-} - \mathcal{F}_{i,+} \right] + \frac{\Delta t}{\Delta \lambda} \left[\mathcal{F}_{j,-} - \mathcal{F}_{j,+} \right] \; , \tag{3.9}$$

where Δt is the propagation time step, and $\Delta\phi$ and $\Delta\lambda$ are the latitude and longitude increments, respectively. Equations (3.6) through (3.8) with $\mathcal{N} = 0$ on land and applying Eq. (3.9) on sea points only automatically invokes the required boundary conditions.

Also available is the QUICKEST scheme (Leonard, 1979; Davis and More, 1982) combined with the ULTIMATE TVD (total variance diminishing) limiter (Leonard, 1991). This is the default propagation scheme for WWATCH. This scheme is third-order accurate in both space and time, and has been selected based on the extensive intercomparison of higher order finite difference schemes for water quality models performed by (see Cahyono, 1994; Falconer and Cayhono, 1993; Tolman, 1995). This scheme is applied to propagation

in longitudinal and latitudinal directions separately, alternating the direction to be treated first.

In the QUICKEST scheme the flux between grid points with counters i and $i-1$ in ϕ-space ($\mathcal{F}_{i,-}$) is calculated as[4]

$$\mathcal{F}_{i,-} = \left[\dot{\phi}_b \, \mathcal{N}_b \right]^n_{j,l,m} \, , \tag{3.10}$$

$$\dot{\phi}_b = 0.5 \, \left(\dot{\phi}_{i-1} + \dot{\phi}_i \right) \, , \tag{3.11}$$

$$\mathcal{N}_b = \frac{1}{2} \left[\, (1+C)\mathcal{N}_{i-1} + (1-C)\mathcal{N}_i \, \right] - \left(\frac{1-C^2}{6} \right) \mathcal{CU} \, \Delta\phi^2, \tag{3.12}$$

$$\mathcal{CU} = \left\{ \begin{array}{ll} (\, \mathcal{N}_{i-2} - 2\mathcal{N}_{i-1} + \mathcal{N}_i \,) \, \Delta\phi^{-2} & \text{for} \quad \dot{\phi}_b \geq 0 \\ (\, \mathcal{N}_{i-1} - 2\mathcal{N}_i + \mathcal{N}_{i+1} \,) \, \Delta\phi^{-2} & \text{for} \quad \dot{\phi}_b < 0 \end{array} \right. , \tag{3.13}$$

$$C = \frac{\dot{\phi}_b \, \Delta t}{\Delta\phi} \, , \tag{3.14}$$

where $\mathcal{CU}$ is the (upstream) curvature of the action density distribution, and where C is a CFL number including a sign to identify the propagation direction. Like the first order scheme, this scheme gives stable solutions for $|C| \leq 1$. To assure that this scheme does not generate aphysical extrema, it is used in combination with the ULTIMATE limiter. This limiter uses the central, upstream and downstream action density (suffices c, u and d, respectively), which are defined as

$$\begin{array}{lll} \mathcal{N}_c = \mathcal{N}_{i-1} \, , & \mathcal{N}_u = \mathcal{N}_i \, , & \mathcal{N}_d = \mathcal{N}_{i-2} \quad \text{for} \quad \dot{\phi}_b \geq 0 \\ \mathcal{N}_c = \mathcal{N}_i \, , & \mathcal{N}_u = \mathcal{N}_{i-1} \, , & \mathcal{N}_d = \mathcal{N}_{i+1} \quad \text{for} \quad \dot{\phi}_b < 0 \end{array} \, . \tag{3.15}$$

To assess if the initial state and the solution show similar monotonic or non-monotonic behavior, the normalized action $\tilde{\mathcal{N}}$ is defined

$$\tilde{\mathcal{N}} = \frac{\mathcal{N} - \mathcal{N}_u}{\mathcal{N}_d - \mathcal{N}_u} \, . \tag{3.16}$$

If the initial state is monotonic (i.e., $0 \leq \tilde{\mathcal{N}}_c \leq 1$), the (normalized) action at the cell boundary $\mathcal{N}_b$ is limited to

[4] Fluxes ($\mathcal{F}_{i,+}$) between grid points with counters $i+1$ and i again are obtained by substituting the appropriate indices.

$$\tilde{\mathcal{N}}_c \leq \tilde{\mathcal{N}}_b \leq 1 \quad , \quad \tilde{\mathcal{N}}_b \leq \tilde{\mathcal{N}}_c \, C^{-1} \quad . \tag{3.17}$$

otherwise

$$\tilde{\mathcal{N}}_b = \tilde{\mathcal{N}}_c \, . \tag{3.18}$$

An alternative scheme is necessary if one of the two grid points adjacent to the cell boundary is on land or represents an active boundary point. In such cases, Eqs. (3.7) and (3.12) are replaced by

$$\dot{\phi}_b = \dot{\phi}_s \, , \tag{3.19}$$

$$\mathcal{N}_b = \mathcal{N}_u \, , \tag{3.20}$$

where the suffix s indicates the (average of) the sea point(s). This boundary condition represents a simple first order upwind scheme, which does not require the limiter (3.15) through (3.18).

The final propagation scheme, similar to Eq. (3.9), becomes

$$\mathcal{N}^{n+1}_{i,j,l,m} = \mathcal{N}^{n}_{i,j,l,m} + \frac{\Delta t}{\Delta \phi} \left[\mathcal{F}_{i,-} - \mathcal{F}_{i,+} \right] \, . \tag{3.21}$$

The scheme for propagation in λ-space is simply obtained by rotating indices and increments in the above equations. For two-dimensional propagation, two versions of the above boundary treatment are available. In the 'hard' approach (default for WWATCH), the boundary conditions are applied in each space separately. This results in small islands and headlands to completely block swell propagation, but it also results in an (erroneous) absorption of swell energy for coast lines under an angle with the grid. The latter deficiency can be partially removed, if swell energy is allowed to propagate onto land between the longitude and latitude propagation steps (denoted as the 'soft' approach). Small islands and headlands then become partially transparent with respect to swell propagation.

3.3.2 Garden Sprinkler Effect

The ULTIMATE QUICKEST scheme is sufficiently free of numerical diffusion for the so-called 'Garden Sprinkler Effect' (GSE) to occur, i.e., a continuous swell field disintegrates into a set of discrete swell fields due to the discrete description of the spectrum (Booij and Holthuijsen, 1987, Fig. 3c). Several GSE alleviation methods are available in WWATCH.

The classical GSE alleviation method is given by Booij and Holthuijsen (1987), who derived an alternative propagation equation for the discrete spectrum, including a diffusive correction to account for continuous dispersion in spite of the discrete spectral description. This correction influences spatial propagation only, which for general spatial coordinates (x, y) becomes

$$\frac{\partial \mathcal{N}}{\partial t} + \frac{\partial}{\partial x}\left[\dot{x}\mathcal{N} - D_{xx}\frac{\partial \mathcal{N}}{\partial x}\right] + \frac{\partial}{\partial y}\left[\dot{y}\mathcal{N} - D_{yy}\frac{\partial \mathcal{N}}{\partial y}\right] - 2D_{xy}\frac{\partial^2 \mathcal{N}}{\partial x \partial y} = 0\,, \quad (3.22)$$

$$D_{xx} = D_{ss}\,\cos^2\theta + D_{nn}\,\sin^2\theta\,, \quad (3.23)$$

$$D_{yy} = D_{ss}\,\sin^2\theta + D_{nn}\,\cos^2\theta\,, \quad (3.24)$$

$$D_{xy} = (D_{ss} - D_{nn})\,\cos\theta\,\sin\theta\,, \quad (3.25)$$

$$D_{ss} = (\Delta c_g)^2\,T_s/12\,, \quad (3.26)$$

$$D_{nn} = (c_g \Delta\theta)^2\,T_s/12\,, \quad (3.27)$$

where D_{ss} is the diffusion coefficient in the propagation direction of the discrete wave component, D_{nn} is the diffusion coefficient along the crest of the discrete wave component and T_s is the time elapsed since the generation of the swell. In the present fractional step method the diffusion can be added as a separate step

$$\frac{\partial \mathcal{N}}{\partial t} = \frac{\partial}{\partial x}\left[D_{xx}\frac{\partial \mathcal{N}}{\partial x}\right] + \frac{\partial}{\partial y}\left[D_{yy}\frac{\partial \mathcal{N}}{\partial y}\right] + 2D_{xy}\frac{\partial^2 \mathcal{N}}{\partial x \partial y}\,. \quad (3.28)$$

This equation is incorporated with two simplifications, the justification of which is discussed in Tolman (1995). First, the swell 'age' T_s is kept constant throughout the model (defined by the user, no default value available). Secondly, the diffusion coefficients D_{ss} and D_{nn} are calculated assuming deep water

$$D_{ss} = \left((X_\sigma - 1)\,\frac{\sigma_m}{2k_m}\right)^2 \frac{T_s}{12}\,, \quad (3.29)$$

$$D_{nn} = \left(\frac{\sigma_m}{2k_m}\Delta\theta\right)^2 \frac{T_s}{12}\,, \quad (3.30)$$

where X_σ is defined as in Eq. (3.1). With these two assumptions, the diffusion tensor becomes constant throughout the spatial domain for each separate spectral component.

Equation (3.28) is solved using a forward-time central-space scheme. At the cell interface between points i and $i-1$ in ϕ (x) space, the term in brackets in the first term on the right side of Eq. (3.28) (denoted as $\mathcal{D}_{i,-}$) is estimated as

$$D_{xx}\frac{\partial \mathcal{N}}{\partial x} \approx \mathcal{D}_{i,-} = D_{xx}\ \left(\frac{\mathcal{N}_i - \mathcal{N}_{i-1}}{\Delta x}\right)\Bigg|_{j,l,m} . \tag{3.31}$$

Corresponding values for counters i and $i+1$, and for gradients in λ (y) space again are obtained by rotating indices and increments. If one of the two grid points in located on land, Eq. (3.31) is set to zero. The mixed derivative at the right side of Eq. (3.28) (denoted as $\mathcal{D}_{ij,--}$) is estimated for the grid point i and $i-1$ in x-space and j and $j-1$ in y-space is estimated as

$$\mathcal{D}_{ij,--} = \ D_{xy}\left(\frac{-\mathcal{N}_{i,j} + \mathcal{N}_{i-1,j} + \mathcal{N}_{i,j-1} - \mathcal{N}_{i-1,j-1}}{0.5(\Delta x_j + \Delta x_{j-1})\,\Delta y}\right)\Bigg|_{l,m} . \tag{3.32}$$

Note that the increment Δx is a function of y due to the use of the spherical grid. This term is evaluated only if all four grid points considered are sea points, otherwise it is set to zero. Using a forward in time discretization of the first term in Eq. (3.28), and central in space discretizations for the remainder of the first and second term on the right side, the final algorithm becomes

$$\begin{aligned}\mathcal{N}^{n+1}_{i,j,l,m} = \mathcal{N}^{n}_{i,j,l,m} &+ \frac{\Delta t}{\Delta x}\left(\mathcal{D}_{i,+} - \mathcal{D}_{i,-}\right) + \frac{\Delta t}{\Delta y}\left(\mathcal{D}_{j,+} - \mathcal{D}_{j,-}\right) \\ &+ \frac{\Delta t}{4}\left(\mathcal{D}_{ij,--} + \mathcal{D}_{ij,-+} + \mathcal{D}_{ij,+-} + \mathcal{D}_{ij,++}\right) .\end{aligned} \tag{3.33}$$

Stable solutions are obtained for (e.g., Fletcher, 1988, Part I section 7.1.1)

$$\frac{D_{\max}\,\Delta t}{\min(\Delta x, \Delta y)^2} \leq 0.5\,, \tag{3.34}$$

where $D_{\max}$ is the maximum value of the diffusion coefficient (typically $D_{\max} = D_{nn}$). Because this stability criterion is a quadratic function of the grid increment, stability can become a serious problem at high latitudes for large scale applications. To avoid that this puts undue constraints on the time step of a model, a corrected swell age $T_{s,c}$ is used

$$T_{s,c} = T_s \min \left\{ 1 \, , \, \left(\frac{\cos(\phi)}{\cos(\phi_c)} \right)^2 \right\} \, , \tag{3.35}$$

where ϕ_c is a cut-off latitude defined by the user.

The above diffusion is needed for swell propagation only, but is not realistic for growing wind seas. In the latter conditions, the ULTIMATE QUICKEST scheme without the dispersion correction is sufficiently smooth to render stable fetch-limited growth curves (Tolman, 1995). To remove minor oscillations, a small isotropic diffusion is used for growing wave components. To assure that this diffusion is small and equivalent for all spectral components, it is calculated from a preset cell Reynolds (or cell Peclet) number $\mathcal{R} = c_g \Delta x D_g^{-1} = 10$, where D_g is the isotropic diffusion for growing components

$$D_g = \frac{c_g \, \min(\Delta x, \Delta y)}{\mathcal{R}} \, , \tag{3.36}$$

The diffusions for swell and for wind seas are combined using a linear combination depending on the nondimensional wind speed or inverse wave age $u_{10} c^{-1} = u_{10} k \sigma^{-1}$ as

$$X_g = \min \left\{ 1 \, , \, \max \left[0 \, , \, 3.3 \left(\frac{k \, u_{10}}{\sigma} \right) - 2.3 \right] \right\} \, , \tag{3.37}$$

$$D_{ss} = X_g D_g + (1 - X_g) D_{ss,p} \, , \tag{3.38}$$

$$D_{nn} = X_g D_g + (1 - X_g) D_{nn,p} \, , \tag{3.39}$$

where the suffix p denotes propagation diffusions as defined in Eqs. (3.29) and (3.30). The constants in Eqs (3.36) and (3.37) are preset in the model.

The major drawback of the above GSE alleviation method is its potential impact on model economy as discussed in relation to Eq. (3.34) and in Tolman (2001, 2002a). For this reason, two additional GSE alleviation methods have been developed for WWATCH.

The first of these two methods, which represents the default for WWATCH, replaces the additional diffusion step (3.28) with a separate fractional step in which direct averaging of the field of energy densities for a given spectral component is considered. The area around each grid point over which

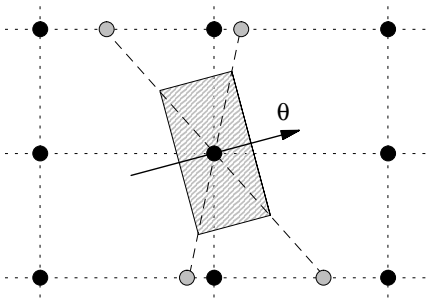

Figure 3.1: Graphical depiction of spatial averaging GSE alleviation technique used here. Solid circles and dotted lines represent the spatial grid. Hatched area represent averaging area to be considered. Corner point values are obtained from the central grid point and the grey points. The latter values are obtained by interpolation from adjacent grid points (from Tolman, 2002a).

---

the averaging is performed extends in the propagation (**s**) and normal (**n**) directions as

$$\pm\gamma_{a,s}\ \Delta c_g\ \Delta t\ \mathbf{s}\quad ,\pm\gamma_{a,n}\ c_g\Delta\theta\ \Delta t\ \mathbf{n}\quad , \tag{3.40}$$

where $\gamma_{a,s}$ and $\gamma_{a,n}$ are tunable constants, the default value of which is set to 1.5. This averaging is graphically depicted in Fig. 3.1. Note that these values may require some retuning for practical applications, as discussed in Tolman (2002a). Appendix A of the latter paper presents details of the averaging scheme, including conservation considerations.

Note that this kind of averaging with dominant directions **s** and **n** is similar to the Booij and Holthuijsen (1987) diffusion method, that uses the same main directions. The averaging method, however, never influences the time step, because it is completely separated from the actual propagation. Moreover, if explicit schemes are used with typically $c_g\Delta t/\Delta x < 1$, it is obvious that the averaging over the area as defined in (3.40) will generally require information at directly neighboring spatial grid points only, as in Fig. 3.1. Furthermore, this method does not require high-latitude filtering.

As is illustrated in Tolman (2002a,d), this method gives virtually identical results as the previous method, but does so at slightly lower costs. For high

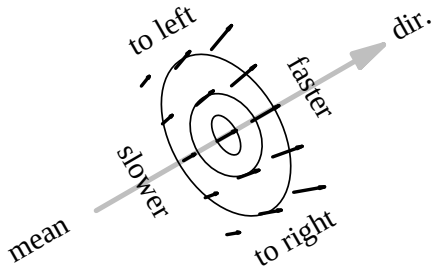

Figure 3.2: Graphical depiction of actual dispersion pattern of discrete wave field (from Tolman, 2002a).

---

resolution applications, the averaging method may become dramatically more economical.

The third and final GSE alleviation method available considers that the advection for a give discrete spectral bin is not unidirectional, but in fact divergent as illustrated in Fig. 3.2.

A linear (in physical space) correction algorithm for the advection speed and direction has been implemented. First, the center of a discrete swell field for the spectral bin (k_m, θ_l) is defined by the location of the maximum energy density $F(k_m, \theta_l)_{\max}$. Secondly, the extent of the swell field in the propagation (θ) direction, $r_{s,\max}$, and in the normal direction, $r_{n,\max}$ are estimated by evaluating whether

$$F(k_m, \theta_l; x, y) < \gamma_{d,0} F(k_m, \theta_l)_{\max} \quad . \tag{3.41}$$

Ideally, $\gamma_{d,0} = 0$, but numerical considerations require that $\gamma_{d,0} > 0$. The default setting is $\gamma_{d,0} = 0.05$. Assuming a linear correction of advection speed and direction with the actual propagation (r_s) and normal (r_n) distances to the center of the discrete swell field, the advection speed c_g and direction θ, relative to their original bin-mean values $c_{g,0}$ and θ_0 become

$$c_g = c_{g,0} + \gamma_{d,s}\, \tilde{r}_s \Delta c_g \quad , \tag{3.42}$$

$$\theta = \theta_{g,0} + \gamma_{d,n}\, \tilde{r}_n \Delta\theta \quad , \tag{3.43}$$

where $\Delta c_g = 0.5(X_\sigma - X_\sigma^{-1}) c_{g,0}$ and $\tilde{r}_s$ and $\tilde{r}_n$ are distances, normalized with $r_{s,\max}$ and $r_{n,\max}$, respectively. To avoid aphysical corrections for trace

energy, the normalized distances have been limited to $|\tilde{r}_s| < 2$ and $|\tilde{r}_n| < 3$. The constants $\gamma_{d,s}$ and $\gamma_{d,n}$ are tunable parameters with a default setting of 0.6.

Although this method has given promising results, it will require further development to become economically viable as discussed in Tolman (2002a).

3.3.3 Unresolved obstacles

Even with the original tuning of WWATCH version 1.15 (Tolman, 2002f), it was clear that unresolved islands groups are a major source of local wave model errors. This was illustrated in some more detail in Tolman (2001), Fig. 3 and Tolman et al. (2002), Fig. 8. In WWATCH, a methodology from the SWAN model (Booij et al., 1999; Holthuijsen et al., 2001) was adopted to apply the effects of unresolved obstacles at the cell boundaries of the spatial grid within the numerical scheme. In this approach, the numerical fluxes between cells through their common boundary are suppressed according to the degree of obstruction provided by the unresolved obstacle. In this approach, the numerical propagation scheme of the ULTIMATE QUICKEST scheme of Eq. (3.21) is modified as

$$\mathcal{N}^{n+1}_{i,j,l,m} = \mathcal{N}^{n}_{i,j,l,m} + \frac{\Delta t}{\Delta \phi} \left[\alpha_{i,-} \mathcal{F}_{i,-} - \alpha_{i,+} \mathcal{F}_{i,+} \right] \; . \tag{3.44}$$

where $\alpha_{i,-}$ and $\alpha_{i,+}$ are 'transparencies' of the corresponding cell boundaries, ranging from 0 (closed boundary) to 1 (no obstructions). For outflow boundaries, transparencies by definition are 1, otherwise energy will artificially accumulate in cells. For inflow boundaries, transparencies less than 1 result in elimination of obstructed energy at the cell boundary. This approach is graphically depicted in Fig. 3.3. Note that a similar approach is easily adopted in the first order scheme (3.9).

Two methods for defining the obstructions are available in the model. The first defines the obstructions directly at the grid boundary. This requires the generation of staggered depth-transparency grids. The second allows the user to define depths and transparencies at the same grid. In this case, the transparency at the inflow boundary becomes $0.5(1 + \alpha_i)$, and the outflow transparency by definition is 1. To complete the total transparency α_i, the next cell in the flow direction will have an inflow transparency $2\alpha_i/(1 +$

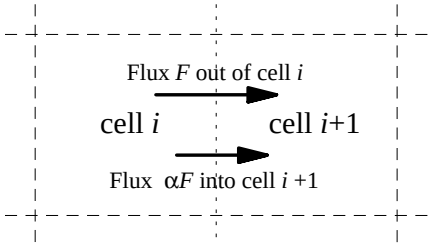

Figure 3.3: Graphical depiction of treatment of unresolved obstacles. Common cell boundary (dotted line) has transparency α. Dashed lines represent other cell boundaries. Numerical flux from left to right.

α_i). If consecutive cells are partially obstructed, the product of individual transparencies is applied.

This approach can also be used to continuously model the effects of ice coverage on wave propagation. This will be discussed in section 3.7. Details of the sub-grid treatment of islands and ice can be found in Tolman (2002e). A study of impacts of this approach in large scale wave models is presented in Tolman (2002d,e).

Because it requires the generation of additional obstruction grids, the default setting of WWATCH is not to include sub-grid modeling of obstacles. Note that this option does not involve compile-level choices, but is entirely controlled from the grid preprocessor (see following chapter).

3.4 Intra-spectral propagation

The third step of the numerical algorithm considers refraction and residual (current-induced) wavenumber shifts. For both the spherical and Cartesian grid, the equation to be solved in this step becomes

$$\frac{\partial N}{\partial t} + \frac{\partial}{\partial k}\,\dot{k}_g N + \frac{\partial}{\partial \theta}\,\dot{\theta}_g N = 0\,, \tag{3.45}$$

$$\dot{k}_g = \frac{\partial \sigma}{\partial d}\frac{\mathbf{U}\cdot\nabla_x d}{c_g} - \mathbf{k}\cdot\frac{\partial \mathbf{U}}{\partial s}\,. \tag{3.46}$$

where $\dot{k}_g$ is the wavenumber velocity relative to the grid, and $\dot{\theta}_g$ is given by (2.15) and (2.11). This equation does not require boundary conditions in θ-space, as the model by definition uses the full (closed) directional space. In k-space, however, boundary conditions are required. At low wavenumbers, it is assumed that no wave action exists outside the discrete domain. It is therefore assumed that no action enters the model at the discrete low-wavenumber boundary. At the high-wavenumber boundary, transport across the discrete boundary is calculated assuming a parametric spectral shape as given by Eq. (2.18). The derivatives of the depth as needed in the evaluation of $\dot{\theta}$ are mostly determined using central differences. For points next to land, however, one-sided differences using sea points only are used.

Propagation in θ-space can cause practical problems in an explicit numerical scheme, as the refraction velocity can become extreme for long waves in extremely shallow water. To avoid the need of extremely small time steps due to refraction, the propagation velocity in θ-space (2.11) is filtered with respect to the depth refraction term

$$\dot{\theta} = X_{rd}(\lambda,\phi,k)\dot{\theta}_d + \dot{\theta}_c \;, \tag{3.47}$$

where the indices d and c refer to the depth and current related fraction of the refraction velocity in (2.11). The filter factor X_{rd} is calculated for every wavenumber and location separately, and is determined so that the CFL number for propagation in θ-space due to the *depth* refraction term cannot exceed a pre-set (user defined) value (default 0.7). This corresponds to a reduction of the bottom slope for some low frequency wave components. The effected components are expected to carry little energy because they are in extremely shallow water. Long wave components carrying significant energy are usually traveling towards the coast, where their energy is dissipated anyway.

As with the propagation in physical space, a first order and an ULTIMATE QUICKEST scheme are available. In the first order scheme the fluxes in θ- and k-space are calculated Cf. Eqs. (3.6) through (3.8) (replacing $\mathcal{N}$ with N and rotating the appropriate counters). The complete first order scheme becomes

$$N^{n+1}_{i,j,l,m} = N^{n}_{i,j,l,m} + \frac{\Delta t}{\Delta\theta}\left[\mathcal{F}_{l,-} - \mathcal{F}_{l,+}\right] + \frac{\Delta t}{\Delta k_m}\left[\mathcal{F}_{m,-} - \mathcal{F}_{m,+}\right] \;, \tag{3.48}$$

where $\Delta\phi$ is the directional increment, and Δk_m is the (local) wavenumber increment. The low-wavenumber boundary conditions is applied by taking $\mathcal{F}_{m,-} = 0$ for $m = 1$, and the high wavenumber boundary condition is calculated using the parametric approximation (2.18) for N, extending the discrete grid by one grid point to high wavenumbers.

The ULTIMATE QUICKEST scheme for the θ-space is implemented similar to the scheme for physical space, with the exception that the closed direction space does not require boundary conditions. The variable grid spacing in k-space requires some modifications to the scheme as outlined by (Leonard, 1979, Appendix). Equations (3.10) through (3.14) then become

$$\mathcal{F}_{m,-} = \left[\dot{k}_{g,b}\, N_b \right]^n_{i,j,l} \,, \tag{3.49}$$

$$\dot{k}_{g,b} = 0.5 \, \left(\dot{k}_{g,m-1} + \dot{k}_{g,m} \right) \,, \tag{3.50}$$

$$N_b = \frac{1}{2} \left[\, (1+C) N_{i-1} + (1-C) N_i \, \right] - \frac{1-C^2}{6} \, \mathcal{CU} \, \Delta k^2_{m-1/2}, \tag{3.51}$$

$$\mathcal{CU} = \left\{ \begin{array}{ll} \frac{1}{\Delta k_{m-1}} \left[\frac{N_m - N_{m-1}}{\Delta k_{m-1/2}} - \frac{N_{m-1} - N_{m-2}}{\Delta k_{m,-3/2}} \right] & \text{for} \quad \dot{k}_b \geq 0 \\ \frac{1}{\Delta k_m} \left[\frac{N_{m+1} - N_m}{\Delta k_{m+1/2}} - \frac{N_m - N_{m-1}}{\Delta k_{m-1/2}} \right] & \text{for} \quad \dot{k}_b < 0 \end{array} \right. \,, \tag{3.52}$$

$$C = \frac{\dot{k}_{g,b}\, \Delta t}{\Delta k_{m-1/2}} \,, \tag{3.53}$$

where Δk_m is the discrete band or cell width at grid point m, and where $\Delta k_{m-1/2}$ is the distance between grid points with counters m and $m-1$. The ULTIMATE limiter can be applied as in Eqs. (3.15) through (3.18), if the CFL number of Eq. (3.53) is used. At the low- and high-wavenumber boundaries the fluxes again are estimated using a first-order upwind approach, with boundary conditions as above defined for the first-order scheme. The final scheme in k-space becomes

$$N^{n+1}_{i,j,l,m} = N^n_{i,j,l,m} + \frac{\Delta t}{\Delta k_m} \left[\mathcal{F}_{m,-} - \mathcal{F}_{m,+} \right] \,, \tag{3.54}$$

3.5 Source terms

Finally, the source terms are accounted for by solving

$$\frac{\partial N}{\partial t} = \mathcal{S} \, . \tag{3.55}$$

As in WAM, a semi-implicit integration scheme is used. In this scheme the discrete change of action density ΔN becomes (WAMDIG, 1988)

$$\Delta N(k,\theta) = \frac{\mathcal{S}(k,\theta)}{1 - \epsilon D(k,\theta)\Delta t} \, , \tag{3.56}$$

where D represents the diagonal terms of the derivative of $\mathcal{S}$ with respect to N (WAMDIG, 1988, Eqs. 4.1 through 4.10), and where ϵ defines the offset of the scheme. Originally, $\epsilon = 0.5$ was implemented to obtain a second order accurate scheme. Presently, $\epsilon = 1$ is used as it is more appropriate for the large time steps in the equilibrium range of the spectrum (Hargreaves and Annan, 1998, 2001), and as it result in much smoother integration of the spectrum. The change of ϵ has little impact on mean wave parameters, but makes the dynamical time stepping as described below more economical.

The semi-implicit scheme is applied in the framework of a dynamic time-stepping scheme (Tolman, 1992). In this scheme, integration over the global time step Δt_g can be performed in several dynamic time steps Δt_d, depending on the net source term $\mathcal{S}$, a maximum change of action density ΔN_m and the remaining time in the interval Δt_g. For the n^{th} dynamic time step in the integration over the interval Δt_g, Δt_d^n is calculated in three steps as

$$\Delta t_d^n = \min_{f<f_{hf}} \left[\frac{\Delta N_m}{|\mathcal{S}|} \left(1 + \epsilon D \frac{\Delta N_m}{|\mathcal{S}|} \right)^{-1} \right] \, , \tag{3.57}$$

$$\Delta t_d^n = \max \left[\, \Delta t_d^n \, , \, \Delta t_{d,\min} \right] \, , \tag{3.58}$$

$$\Delta t_d^n = \min \left[\, \Delta t_d^n \, , \, \Delta t_g - \sum_{i=1}^{n-1} \Delta t_d^i \right] \, , \tag{3.59}$$

where $\Delta t_{\min}$ is a user-defined minimum time step, which is added to avoid excessively small time steps. The corresponding new spectrum N^n becomes

$$N^n = \max \left[\, 0 \, , \, N^{n-1} + \left(\frac{\mathcal{S}\Delta t_d}{1 - \epsilon D \Delta t_d} \right) \right] \, . \tag{3.60}$$

	X_p	X_r	X_f	$\Delta t_{d,\min}$
WAM equivalent	$\frac{\pi}{24}10^{-3}\Delta t$	$\infty(\geq 1)$	–	Δt_g
suggested	0.1-0.2	0.1-0.2	0.05	$\approx 0.1\Delta t_g$
default setting	0.15	0.10	0.05	–

Table 3.1: User-defined parameters in the source term integration scheme

The maximum change of action density ΔN_m is determined from a parametric change of action density ΔN_p and a filtered relative change ΔN_r

$$\Delta N_m(k,\theta) = \min\,[\,\Delta N_p(k,\theta)\,,\;\Delta N_r(k,\theta)\,]\;, \tag{3.61}$$

$$\Delta N_p(k,\theta) = X_p\,\frac{\alpha}{\pi}\,\frac{(2\pi)^4}{g^2}\,\frac{1}{\sigma k^3}\;, \tag{3.62}$$

$$\Delta N_r(k,\theta) = X_r\;\max\,[\,N(k,\theta)\,,\;N_f\,]\;, \tag{3.63}$$

$$N_f = \max\left[\,\Delta N_p(k_{\max},\theta)\,,\;X_f\;\max_{\forall k,\theta}\{N(k,\theta)\}\,\right]\;, \tag{3.64}$$

where X_p, X_r and X_f are user-defined constants (see Table 3.1), α is a PM energy level (set to $\alpha = 0.62\,10^{-4}$) and $k_{\max}$ is the maximum discrete wavenumber. The parametric spectral shape in (3.62) corresponds in deep water to the well-known high-frequency shape of the one-dimensional frequency spectrum $F(f) \propto f^{-5}$. The link between the filter level and the maximum parametric change in (3.64) is used to assure that the dynamic time step remains reasonably large in cases with extremely small wave energies. A final safeguard for stability of integration is provided by limiting the discrete change of action density to the maximum parametric change (3.62) in conditions where Eq. (3.58) dictates Δt_d^n. In this case Eq. (3.58) becomes a limiter as in the WAM model. Impacts of limiters are discussed in detail in for instance Hersbach and Janssen (1999, 2001), Hargreaves and Annan (2001) and Tolman (2002c).

Note that the dynamic time step is calculated for each grid point separately, thus adding additional computational effort only for grid points in which the spectrum is subject to rapid change. Note furthermore, that the source terms are re-calculated for every dynamic time step.

The present source terms do not include a linear growth term. This implies that waves can only grow if some energy is present in the spectrum. In small-scale applications with persistent low wind speeds, wave energy might disappear completely from part of the model. To assure that wave growth can occur when the wind increases, a so-called seeding option is available in WWATCH (selected during compilation). If the seeding option is selected, the energy level at the seeding frequency $\sigma_{\rm seed} = \min(\sigma_{\rm max}, 2\pi f_{hf})$ is required to at least contain a minimum action density

$$\begin{aligned} N_{\rm min}(k_{seed},\theta) &= 6.25\,10^{-4} \frac{1}{k_{\rm seed}^3\,\sigma_{\rm seed}} \max\left[\,0.\,,\,\cos^2(\theta-\theta_w)\right] \\ &\qquad \min\left[\,1\,,\,\max\left(\,0\,,\,\frac{|u_{10}|}{X_{\rm seed} g \sigma_{\rm seed}^{-1}} - 1\,\right)\right]\,, \end{aligned} \tag{3.65}$$

where $g\sigma_{\rm seed}^{-1}$ approximates the equilibrium wind speed for the highest discrete spectral frequency. This minimum action distribution is aligned with the wind direction, goes to zero for low wind speeds, and is proportional to the integration limiter (3.62) for large wind speeds. $X_{\rm seed} \geq 1$ is a user-defined parameter to shift seeding to higher frequencies. Seeding starts if the wind speed reaches $X_{\rm seed}$ times the equilibrium wind speed for the highest discrete frequency, and reaches its full strength for twice as high wind speeds. The default model settings include the seeding algorithm, with $X_{\rm seed} = 1$.

3.6 Winds and currents

Model input mainly consists of wind and current fields. Within the model, winds and currents are updated at every time step Δt_g and represent values at the end of the time step considered. Several interpolation methods are available (selected during compilation). By default, the interpolation in time consists of a linear interpolation of the velocity and the direction (turning the wind or current over the smallest angle). The wind speed or current velocity can optionally be corrected to (approximately) conserve the energy instead of the wind velocity. The corresponding correction factor X_u is calculated as

$$X_u = \max\left[\,1.25\,,\,\frac{u_{10,rms}}{u_{10,l}}\right]\,, \tag{3.66}$$

where $u_{10,l}$ is the linearly interpolated velocity and $u_{10,rms}$ is the rms interpolated velocity. Finally, winds can optionally be kept constant and changed discontinuously (option not available for current).

Note that the auxiliary programs of WWATCH include a program to preprocess input fields (see section 4.4.4). This program transfers gridded fields to the grid of the wave model. For winds and currents this program utilizes a bilinear interpolation of vector components. This interpolation can be corrected to (approximately) conserve the velocity or the energy of the wind or the current by utilizing a correction factor similar to Eq. (3.66).

3.7 Ice coverage

Ice covered sea is considered as 'land' in WWATCH, assuming zero wave energy and boundary conditions at ice edges are identical to boundary conditions at shore lines. Grid points are taken out of the calculation if the ice concentration becomes larger than a user-defined concentration. If the ice concentration drops below its critical value, the corresponding grid point is 're-activated'. The spectrum is then initialized with a PM spectrum based on the local wind direction with a peak frequency corresponding to the second-highest discrete frequency in the grid. A small spectrum is used to assure that spectra are realistic, even for shallow coastal points.

The above discontinuous ice treatment represents the default model setting in WWATCH. In the framework of the modeling of unresolved obstacles as discussed in section 3.3.3, a continuous method is also available, as given by Tolman (2002e). In this method, a user-defined critical ice concentration at which obstruction begins ($\epsilon_{c,0}$) and is complete ($\epsilon_{c,n}$) are given (defaults are $\epsilon_{c,0} = \epsilon_{c,n} = 0.5$, i.e., discontinuous treatment of ice). From these critical concentrations, corresponding decay length scales are calculated as,

$$l_0 = \epsilon_{c,0} \min(\Delta x, \Delta y) \quad . \tag{3.67}$$

$$l_n = \epsilon_{c,n} \min(\Delta x, \Delta y) \quad . \tag{3.68}$$

from which cell transparencies in x and y (α_x and α_y, respectively) are calculated as

$$\alpha_x = \begin{cases} 1 & \text{for} \quad \epsilon\Delta x < l_0 \\ 0 & \text{for} \quad \epsilon\Delta x > l_n \\ \frac{l_n - \epsilon\Delta x}{l_n - l_0} & \text{otherwise} \end{cases} \quad , \quad \alpha_y = \begin{cases} 1 & \text{for} \quad \epsilon\Delta y < l_0 \\ 0 & \text{for} \quad \epsilon\Delta y > l_n \\ \frac{l_n - \epsilon\Delta y}{l_n - l_0} & \text{otherwise} \end{cases} \quad . \tag{3.69}$$

Details of this model can be found in Tolman (2002e).

Updating of the ice map within the model takes place at the discrete model time approximately half way in between the valid times of the old and new ice maps. The map will not be updated, if the time stamps of both ice fields are identical.

3.8 Transferring boundary conditions

WWATCH includes options to transfer boundary conditions from large-scale runs to small-scale runs. Each run can simultaneous accept one data set with boundary conditions, and generate up to 9 data sets with boundary conditions. To assure conservation of wave energy with incompatible depths and currents, the boundary data consists of energy spectra $F(\sigma,\theta)$. The data file consists of spectra at grid points of the generating run, and information needed to interpolate spectra at the requested boundary points. The size of the transfer files is thus minimized if the input points for a small-scale run are located on grid lines in the large-scale run. When used as input, the spectra are interpolated in space and time for every global time step Δt_g, using a linear interpolation of spectral components.

This page is intentionally left blank.

4 Running the wave model

4.1 Program design

The core of WWATCH is the wave model subroutine. Starting with version 2.00 of the model, all initializations for the model have been included in the wave model routine. The wave model routine can be called by either a stand-alone program shell or any other program that requires dynamically updated wave data. Auxiliary programs include a grid preprocessor, a program to generate artificial initial conditions, a generic program shell (and a corresponding input pre-processor) and output post-processors. A relational diagram including the basic data flow is presented in figure 4.1. In the discussion of the model below, file names will be identified by the file type font, the contents of a file by the code type font and FORTRAN program elements by the FORTRAN type font.

The grid preprocessor writes a model definition file mod_def.ww3 with bottom data and parameter values defining the physical and numerical approaches. The wave model requires initial conditions, consisting of a restart file restart.ww3, written by either the wave model itself, or by the initial conditions program. If this file is not available, the wave model will be initialized with a parametric fetch-limited spectrum based on the initial wind field (see the corresponding option in the initial conditions program). The wave model optionally generates up to 9 restart files restart*n*.ww3, where *n* represents a single digit integer number. The wave model also optionally reads boundary conditions from the file nest.ww3 and generates boundary conditions for consecutive runs in nest*n*.ww3. The model furthermore dumps raw data to the output files out_grd.ww3 , out_pnt.ww3 and track_o.ww3 (gridded mean wave parameters, spectra at locations and spectra along tracks, respectively). The tracks along which spectra are to be presented is defined in the file track_i.ww3. Finally, gridded mean wave parameters can be transferred to the program shell through its parameter list. Note that the wave model does not write to standard output, because this would be inconvenient if WWATCH is part of an integrated model. Instead, it maintains its own log file log.ww3 and optionally a test output files test.ww3 for a shared memory version of the model, or test*nnn*.ww3 for distributed memory versions, where *nnn* is the processor number starting with 1. Finally, six output post-

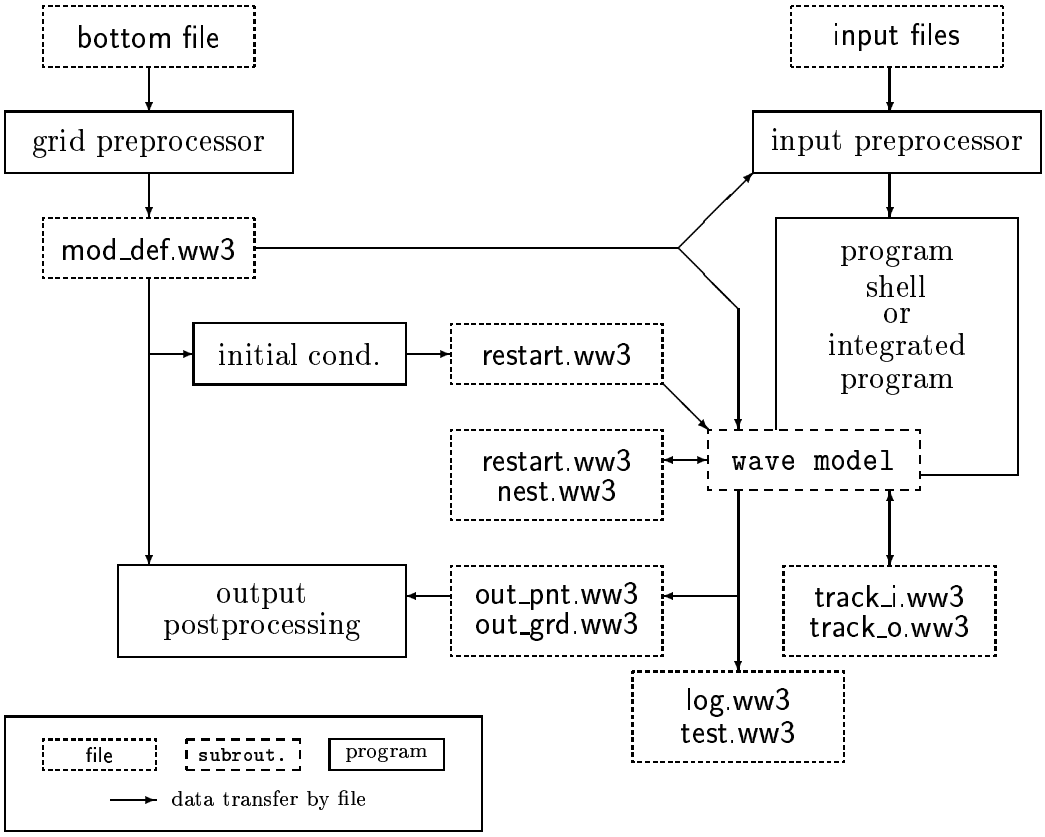

Figure 4.1: Basic program elements and data flow

processors are available (binary post-processing of raw gridded fields, point output and track output files; GRIB packing of gridded data; post-processing for later GrADS graphical processing of gridded and spectral data). A more detailed description of all program elements and there input files is given below. Note that the source codes of each routine are fully documented. This documentation is an additional source of information about WWATCH.

Files specific to WWATCH are opened by name within the program. The unit numbers, however, have to be defined by the user, guaranteeing the largest possible flexibility for implementation in integrated models.

Next to the wave model subroutine, and interface routine for data assimilation is provided. The routine is intended to be run side by side with the wave model routine. The routine includes a generic interface that provides all necessary model components to perform full spectral data assimilation.

This routine is integrated into the generic wave model shell, which is set up to perform time step managements for a wave model with or without data assimilation. The shell also provides a simple yet flexible way to provide the data assimilation scheme with various types of data.

4.2 The wave model routine

As discussed above, the actual wave model is a subroutine. To run the model, a program shell is needed. WWATCH is provided with a simple stand-alone shell as will be discussed in section 4.4.5. The present section concentrates on the wave model subroutine.

Upon first entry, the wave model routine performs model initialization. This includes setting up part of the I/O system by defining unit numbers, initializing internal time management, processing the model definition file (mod_def.ww3), processing initial conditions (restart.ww3), preparing model output, and calculating grid-dependent parameters. If the model is compiled for an MPI environment, all necessary communication for both calculations and output are determined and initialized (the model uses persistent MPI communication throughout).

The wave model routine W3WAVE can be called any number of times to propagate the wave field in time after the initialization has taken place. After some initial checks, the subroutine interpolates winds and currents, updates ice concentrations and water levels, propagates the wave field, and applies the selected source terms for a number of time steps. The internal time step is defined by the interval for which the calculations are to be performed, and by the requested output times. At the end of the calculations, the routine provides the calling program with the requested fields of wave data. A documentation of the interface of W3WAVE can be found in the source code (w3wavemd.ftn).

Apart from the raw data files as described above, the program maintains a log file log.ww3. This file is opened by W3INIT (contained in W3WAVE in w3wavemd.ftn), which writes some self-explanatory header information to this file. Each consecutive call to W3WAVE adds several lines to an 'action table' in this log file as is shown in Fig. 4.2. The column identified as 'step' shows the discrete time step considered. The column identified as 'pass' identifies the sequence number of the call to W3WAVE; i.e., 3 identifies that

```
                                        |    input    |    output   |
                                        |-------------|-------------|
  step | pass |    date      time       | b w l c i d | m g p t r b |
-------|------|---------------------|-------------|-------------|
     0 |   1  | 1968/06/06 00:00:00 |   F         |   X X       |
     8 |   1  |            02:00:00 |             |     X       |
    12 |   1  |            03:00:00 |             |   X         |
    16 |   1  |            04:00:00 |             |     X       |
    24 |   1  |            06:00:00 |   X         |   X X       |
    32 |   2  |            08:00:00 |             |     X       |
    36 |   2  |            09:00:00 |             |   L         |
    40 |   2  |            10:00:00 |             |     X       |
    48 |   2  |            12:00:00 |   X       X |     L   L   |
-------+------+---------------------+-------------+-------------+
```

Figure 4.2: Example action table from file log.ww3.

---

this action took place in the third call to W3WAVE. The third column shows the ending time of the time step. In the input and output columns the corresponding actions of the model are shown. An `X` identifies that the input has been updated, or that the output has been performed. An `F` indicates a first field read, and an `L` identifies the last output. The seven input columns identify boundary conditions (`b`), wind fields (`w`), water levels (`l`), current fields (`c`), ice concentrations (`i`), and data for assimilation (`d`), respectively. Note that data assimilation takes place at the end of the time step after the wave routine call. The six output columns identify direct output to a main program or program shell through the parameter list of W3WAVE (`m`), gridded output (`g`), point output (`p`), output along tracks (`t`), restart files (`r`), and boundary data (`b`), respectively. The output type `m` is by definition not used in the present stand-alone shell, but can be tested using the preprocessor switch TPAR (see next chapter).

4.3 The data assimilation interface

As discussed above, the wave model subroutine is supplemented with a data assimilation interface routine (W3WDAS in w3wdasmd.ftn). This routine is integrated in the stand-alone shell (see section 4.4.5) to provide time step management of a combined wave model / data assimilation scheme. In this a fairly simple approach is assumed where data assimilation is performed at selected times, while the wave model marches forward in time. In the setup of the shell, the data assimilation is performed after the model has reached the target time, but has not yet produced output. After the data assimilation is performed, the wave model routine is called again only to generate output as requested. Thus, the wave model output for a given time will include the effects of data assimilation for that specific target time.

The generic program shell also processes several types of data to be assimilated, and passes it on to the data assimilation interface routine. All data needs to be preprocessed using the wave model input preprocessor (see section 4.4.4), and will be recognized by the generic shell by file name. Presently, up to three different data files can be used. Tentatively, these could be mean wave parameters, one dimensional spectral data, and two dimensional spectral data, respectively. This is, however, not hardwired to the model and in fact needs to be defined by the user.

Presently, no data assimilation packages are available. User supplied data assimilation schemes can be included in the wave model using the interface routine (W3WDAS in w3wdasmd.ftn), the documentation of which should be sufficient for the necessary programming. Details on how to add user supplied software to the WWATCH compilation system can be found in the following chapter. NCEP is presently working on wave data assimilation techniques, but presently has no plans to distribute wave data assimilation software.

4.4 Auxiliary programs

4.4.1 General concepts

All auxiliary programs presented here with the exception of the track output post-processor read input from a pre-defined input file. The first character on the first line of the input file will be considered to be the comment character, identifying comment lines in the input file. This comment character has to appear on the first position of input lines to be effective. In all examples in the following sections lines starting with '$' therefore only contain comment. The programs furthermore all write formatted output to the standard output unit.

In the following sections, all available auxiliary programs are described using an example input file with all options included (partially as comment). These files are identical to the distributed example input files. The sections furthermore show the name of the executable program, the program name (as appears in the program statement), the source code file and input and output files and there unit numbers (in brackets behind the file name). Input and output files marked with *are optional. The intermediate files mentioned below are all UNFORMATTED, and are not described in detail here. Each file is written and read by a single routine, to which reference is made for additional documentation.

mod_def.ww3	Subroutine W3IOGR (w3iogrmd.ftn).
out_grd.ww3	Subroutine W3IOGO (w3iogomd.ftn).
out_pnt.ww3	Subroutine W3IOPO (w3iopomd.ftn).
track_o.ww3	Subroutine W3IOTR (w3iotrmd.ftn).
restart.ww3	Subroutine W3IORS (w3iorsmd.ftn).
nest.ww3	Subroutine W3IOBC (w3iobcmd.ftn).

Preprocessing and compilation of the programs is discussed in the following two chapters. Examples of test runs of the model are provided with the source code.

4.4.2 The grid preprocessor

Program	:	ww3_grid		(W3GRID)
Code	:	ww3_grid.ftn		
Input	:	ww3_grid.inp	(10)	Formatted input file for program.
		'grid file' *	(user)	File with bottom depths.
		'obstr. file' *	(user)	File with sub-grid obstructions.
Output	:	standard out	(6)	Formatted output of program.
		mod_def.ww3	(20)	Model definition file in WWATCH format.
		mask.ww3 *	(20)	Land-sea mask file (switch O2a).
Scratch	:	ww3_grid.scratch	(90)	Formatted scratch file.

Note that bottom and obstruction data may be in same file.

start of example input file

```
$ -------------------------------------------------------------------- $
$ WAVEWATCH III Grid preprocessor input file                            $
$ -------------------------------------------------------------------- $
$ Grid name (C*30, in quotes)
$
  'TEST GRID (GULF OF NOWHERE)   '
$
$ Frequency increment factor and first frequency (Hz) ---------------- $
$ number of frequencies (wavenumbers) and directions
$
   1.1  0.04118  25  24
$
$ Set model flags ---------------------------------------------------- $
$  - FLDRY         Dry run (input/output only, no calculation).
$  - FLCX, FLCY    Activate X and Y component of propagation.
$  - FLCTH, FLCK   Activate direction and wavenumber shifts.
$  - FLSOU         Activate source terms.
$
   F T T T F T
$
$ Set time steps ----------------------------------------------------- $
$ - Time step information (this information is always read)
$     maximum global time step, maximum CFL time step for x-y and
$     k-theta, minimum source term time step (all in seconds).
$
    900. 900. 900. 300.
```

```
$
$ Start of namelist input section ------------------------------------ $
$   Starting with WAVEWATCH III version 2.00, the tunable parameters
$   for source terms, propagation schemes, and numerics are read using
$   namelists. Any namelist found in the folowing sections up to the
$   end-of-section identifier string (see below) is temporarily written
$   to ww3_grid.scratch, and read from there if necessary. Namelists
$   not needed for the given switch settings will be skipped
$   automatically, and the order of the namelists is immaterial.
$   As an example, namelist input to change SWELLF and ZWND in the
$   Tolman and Chalikov input would be
$
$   &SIN2 SWELLF = 0.1, ZWND = 15. /
$
$ Define constants in source terms ---------------------------------- $
$
$ Input  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
$   WAM-3               : Namelist SIN1
$                           CINP   : Proportionality constant.
$
$   Tolman and Chalikov : Namelist SIN2
$                           ZWND   : Height of wind (m).
$                           SWELLF : swell factor in (2.48).
$                           STABSH, STABOF, CNEG, CPOS, FNEG :
$                                    c0, ST0, c1, c2 and f1 in . (2.63)
$                                    through (2.65) for definition of
$                                    effective wind speed (!/STAB2).
$
$ Nonlinear interactions - - - - - - - - - - - - - - - - - - - - - - -
$   Discrete I.A.       : Namelist SNL1
$                           LAMBDA : Lambda in source term.
$                           NLPROP : C in sourc term. NOTE : default
$                                    value depends on other source
$                                    terms selected.
$                           KDCONV : Factor before kd in Eq. (2.24).
$                           KDMIN, SNLCS1, SNLCS2, SNLCS3 :
$                                    Minimum kd, and constants c1-3
$                                    in depth scaling function.
$   Exact interactions  : Namelist SNL2
$                           IQTYPE : Type of depth treatment
$                                     1 : Deep water
$                                     2 : Deep water / WAM scaling
$                                     3 : Shallow water
$                           TAILNL : Parametric tail power.
$                           NDEPTH : Number of depths in for which
```

```
$                                  integration space is established.
$                                  Used for IQTYPE = 3 only
$                        Namelist ANL2
$                          DEPTHS : Array with depths for NDEPTH = 3
$
$ Dissipation  - - - - - - - - - - - - - - - - - - - - - - - - - - - -
$   WAM-3               : Namelist SDS1
$                          CDIS, APM : As in source term.
$
$   Tolman and Chalikov : Namelist SDS2
$                          SDSA0, SDSA1, SDSA2, SDSB0, SDSB1, PHIMIN :
$                                  Constants a0, a1, a2, b0, b1 and
$                                  PHImin.
$
$ Bottom friction  - - - - - - - - - - - - - - - - - - - - - - - - - -
$   JONSWAP             : Namelist SBT1
$                          GAMMA   : As it says.
$
$ Propagation schemes ------------------------------------------------ $
$   First order         : Namelist PR01
$                          CFLTM  : Maximum CFL number for refraction.
$
$   UQ with diffusion   : Namelist PR02
$                          CFLTM  : Maximum CFL number for refraction.
$                          FLSOFT : Flag for 'soft' land boundaries.
$                          DTIME  : Swell age (s) in garden sprinkler
$                                   correction. If 0., all diffusion
$                                   switched off. If small non-zero
$                                   (DEFAULT !!!) only wave growth
$                                   diffusion.
$                          LATMIN : Maximum latitude used in calc. of
$                                   strength of diffusion for prop.
$
$   UQ with averaging   : Namelist PR03
$                          CFLTM  : Maximum CFL number for refraction.
$                          FLSOFT : Flag for 'soft' land boundaries.
$                          WDTHCG : Tuning factor propag. direction.
$                          WDTHTH : Tuning factor normal direction.
$
$   UQ with divergence  : Namelist PR04
$                          CFLTM  : Maximum CFL number for refraction.
$                          FLSOFT : Flag for 'soft' land boundaries.
$                          QTFAC  : Tuning factor Eq. (3.41).
$                          RSFAC  : Tuning factor Eq. (3.42).
$                          RNFAC  : Tuning factor Eq. (3.43).
```

```
$
$ Miscellaneous ------------------------------------------------------ $
$   Misc. parameters    : Namelist MISC
$                           CICE0  : Ice concentration cut-off.
$                           CICEN  : Ice concentration cut-off.
$                           XSEED  : Xseed in seeding alg. (!/SEED).
$                           FLAGTR : Indicating presence and type of
$                                    subgrid information :
$                                     0 : No subgrid information.
$                                     1 : Transparancies at cell boun-
$                                         daries between grid points.
$                                     2 : Transp. at cell centers.
$                                     3 : Like 1 with cont. ice.
$                                     4 : Like 2 with cont. ice.
$                           XP, XR, XFILT
$                                    Xp, Xr and Xf for the dynamic
$                                    integration scheme.
$
$ In the 'Out of the box' test setup we run with sub-grid obstacles
$ and with continuous ice treatment.
$
  &MISC CICE0 = 0.25, CICEN = 0.75, FLAGTR = 4 /
$
$ Mandatory string to identify end of namelist input section.
$
END OF NAMELISTS
$
$ Define grid -------------------------------------------------------- $
$ Four records containing :
$  1 NX, NY. As the outer grid lines are always defined as land
$    points, the minimum size is 3x3.
$  2 Grid increments SX, SY (degr.or m) and scaling (division) factor.
$    If NX*SX = 360., latitudinal closure is applied.
$  3 Coordinates of (1,1) (degr.) and scaling (division) factor.
$  4 Limiting bottom depth (m) to discriminate between land and sea
$    points, minimum water depth (m) as allowed in model, unit number
$    of file with bottom depths, scale factor for bottom depths (mult.),
$    IDLA, IDFM, format for formatted read, FROM and filename.
$      IDLA : Layout indicator :
$                  1   : Read line-by-line bottom to top.
$                  2   : Like 1, single read statement.
$                  3   : Read line-by-line top to bottom.
$                  4   : Like 3, single read statement.
$      IDFM : format indicator :
$                  1   : Free format.
```

```
$                 2   : Fixed format with above format descriptor.
$                 3   : Unformatted.
$      FROM : file type parameter
$               'UNIT' : open file by unit number only.
$               'NAME' : open file by name and assign to unit.
$
$ Example for longitude-latitude grid (switch !/LLG), for Cartesian
$ grid the unit is meters (NOT km).
$
     12     12
      1.     1.     4.
     -1.    -1.     4.
     -0.1 2.50  10  -10. 3 1 '(....)' 'NAME' 'bottom.inp'
$
$ If the above unit number equals 10, the bottom data is read from
$ this file and follows below (no intermediate comment lines allowed).
$
  6 6 6 6 6 6 6 6 6 6 6 6
  6 6 6 5 4 2 0 2 4 5 6 6
  6 6 6 5 4 2 0 2 4 5 6 6
  6 6 6 5 4 2 0 2 4 5 6 6
  6 6 6 5 4 2 0 0 4 5 6 6
  6 6 6 5 4 4 2 2 4 5 6 6
  6 6 6 6 5 5 4 4 5 6 6 6
  6 6 6 6 6 6 5 5 6 6 6 6
  6 6 6 6 6 6 6 6 6 6 6 6
  6 6 6 6 6 6 6 6 6 6 6 6
  6 6 6 6 6 6 6 6 6 6 6 6
  6 6 6 6 6 6 6 6 6 6 6 6
$
$ If sub-grid information is avalaible as indicated by FLAGTR above,
$ additional input to define this is needed below. In such cases a
$ field of fractional obstructions at or between grid points needs to
$ be supplied.  First the location and format of the data is defined
$ by (as above) :
$  - Unit number of file (can be 10, and/or identical to bottem depth
$    unit), scale factor for fractional obstruction, IDLA, IDFM,
$    format for formatted read, FROM and filename
$
   10 0.2  3 1 '(....)' 'NAME' 'obstr.inp'
$
$ *** NOTE if this unit number is the same as the previous bottom
$     depth unit number, it is assumed that this is the same file
$     without further checks.                                    ***
$
```

```
$ If the above unit number equals 10, the bottom data is read from
$ this file and follows below (no intermediate comment lines allowed,
$ except between the two fields).
$
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 5 0 0 0 0 0
  0 0 0 0 0 0 5 0 0 0 0 0
  0 0 0 0 0 0 4 0 0 0 0 0
  0 0 0 0 0 0 4 0 0 0 0 0
  0 0 0 0 0 0 5 0 0 0 0 0
  0 0 0 0 0 0 5 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
$
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 5 5 5 0
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
$
$ *** NOTE size of fields is always NX * NY                        ***
$
$ Input boundary points ---------------------------------------------- $
$   An unlimited number of lines identifying points at which input
$   boundary conditions are to be defined. If the actual input data is
$   not defined in the actual wave model run, the initial conditions
$   will be applied as constant boundary conditions. Each line contains:
$     Discrete grid counters (IX,IY) of the active point and a
$     connect flag. If this flag is true, and the present and previous
$     point are on a grid line or diagonal, all intermediate points
$     are also defined as boundary points.
$
      2   2   F
      2  11   T
$
```

```
$  Close list by defining point (0,0) (mandatory)
$
      0   0   F
$
$ Output boundary points ------------------------------------------- $
$ Output boundary points are defined as a number of straight lines,
$ defined by its starting point (X0,Y0), increments (DX,DY) and number
$ of points. A negative number of points starts a new output file.
$ Note that this data is only generated if requested by the actual
$ program. Example again for spherical grid in degrees.
$
$    1.75  1.50  0.25 -0.10     3
$    2.25  1.50 -0.10  0.00    -6
$    0.10  0.10  0.10  0.00   -10
$
$  Close list by defining line with 0 points (mandatory)
$
     0.    0.    0.    0.       0
$
$ ------------------------------------------------------------------ $
$ End of input file                                                  $
$ ------------------------------------------------------------------ $
```

end of example input file

4.4.3 The initial conditions program

Program	:	ww3_strt		(W3STRT)
Code	:	ww3_strt.ftn		
Input	:	ww3_strt.inp	(10)	Formatted input file for program.
		mod_def.ww3	(20)	Model definition file.
Output	:	standard out	(6)	Formatted output of program.
		restart.ww3	(20)	Restart file in WWATCH format.

start of example input file

```
$ ------------------------------------------------------------------- $
$ WAVEWATCH III Initial conditions input file                          $
$------------------------------------------------------------------- $
$ type of initial field ITYPE .
$
    1
$
$ ITYPE = 1 ---------------------------------------------------------- $
$ Gaussian in frequency and space, cos type in direction.
$ - fp and spread (Hz), mean direction (degr., oceanographic
$    convention) and cosine power, Xm and spread (degr. or m) Ym and
$    spread (degr. or m), Hmax (m) (Example for lon-lat grid in degr.).
$
$   0.10  0.01  270. 2  1. 0.5 1. 0.5 2.5
    0.10  0.01  270. 2  0. 1000. 1. 1000. 2.5
$
$ ITYPE = 2 ---------------------------------------------------------- $
$ JONSWAP spectrum with Hasselmann et al. (1980) direct. distribution.
$ - alfa, peak freq. (Hz), mean direction (degr., oceanographical
$   convention), gamma, sigA, sigB, Xm and spread (degr. or m) Ym and
$   spread (degr. or m)  (Example for lon-lat grid in degr.).
$   alfa, sigA, sigB give default values if less than or equal to 0.
$
$   0.0081  0.1  270.  1.0 0. 0. 1. 100. 1. 100.
$
$ ITYPE = 3 ---------------------------------------------------------- $
$ Fetch-limited JONSWAP
$ - No additional data, the local spectrum is calculated using the
$   local wind speed and direction, using the spatial grid size as
$   fetch, and assuring that the spectrum is within the discrete
$   frequency range.
$
$
```

```
$ ITYPE = 4 ---------------------------------------------------------- $
$ User-defined spectrum
$ - Scale factor., defaults to 1 if less than or equal 0.
$ - Spectrum F(f,theta) (single read statement)
$
$ -0.1
$  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 1 4 2 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 1 2 3 2 1 1 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 1 3 9 7 5 3 2 1 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 1 3 4 3 2 1 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$
$ -------------------------------------------------------------------- $
$ End of input file                                                    $
$ -------------------------------------------------------------------- $
```

end of example input file

4.4.4 The field preprocessor for the generic shell

Program	:	ww3_prep		(W3PREP)
Code	:	ww3_prep.ftn		
Input	:	ww3_prep.inp	(10)	Formatted input file for program.
		mod_def.ww3	(11)	Model definition file.
		'user input'*	(user)	See example below.
Output	:	standard out	(6)	Formatted output of program.
		level.ww3*	(12)	Water levels file.
		current.ww3*	(12)	Current fields file.
		wind.ww3*	(12)	Wind fields file.
		ice.ww3*	(12)	Ice fields file.
		data0.ww3*	(12)	Assimilation data ('mean').
		data1.ww3*	(12)	Assimilation data ('1-D spectra').
		data2.ww3*	(12)	Assimilation data ('2-D spectra').

start of example input file

```
$ ------------------------------------------------------------------- $
$ WAVEWATCH III Field preprocessor input file                         $
$ ------------------------------------------------------------------- $
$ Mayor types of field and time flag
$   Field types  :  ICE   Ice concentrations.
$                   LEV   Water levels.
$                   WND   Winds.
$                   WNS   Winds (including air-sea temp. dif.)
$                   CUR   Currents.
$                   DAT   Data for assimilation.
$   Format types :  AI    Transfer field 'as is'.
$                   LL    Field defined on regular longitude-latitude
$                         or Cartesian grid.
$                   F1    Arbitrary grid, longitude and latitude of
$                         each grid point given in separate file.
$                   F2    Like F1, composite of 2 fields.
$
$   NOTE : Options F1 and F2 have not yet been set up or tested for the
$          Cartesian version of the model.
$        - Format type not used for field type 'DAT'.
$
$   Time flag    : If true, time is included in file.
$
   'ICE' 'LL' F
```

```
$
$ Additional time input ------------------------------------------- $
$ If time flag is .FALSE., give time of field in yyyymmdd hhmmss format.
$
   19680606 053000
$
$ Additional input format type 'LL' ------------------------------- $
$ Grid range (degr. or m) and number of points for axes, respectively.
$ Example for longitude-latitude grid.
$
    -0.25 2.5  15  -0.25  2.5  4
$
$ Additional input format type 'F1' or 'F2' ------------------------ $
$ Three or four additional input lines, to define the file(s) with
$ the grid information :
$ 1) Discrete size of input grid (NXI,NYI).
$ 2) Define type of file using the parameters FROM, IDLA, IDFM (see
$    input for grid preprocessor), and a format
$ 3) Unit number and (dummy) name of first file.
$ 4) Unit number and (dummy) name of second file (F2 only).
$
$ 15  3
$ 'UNIT' 3 1 '(.L.L.)'
$ 10 'll_file.1'
$ 10 'll_file.2'
$
$ Additional input for data --------------------------------------- $
$ Dimension of data (0,1,2 for mean pars, 1D or 2D spectra), "record
$ length" for data, data value for missing data
$
$  0  4  -999.
$
$ Define data files ----------------------------------------------- $
$ The first input line identifies the file format with FROM, IDLA and
$ IDFM, the second (third) lines give the file unit number and name.
$
  'UNIT' 3 1 '(..T..)' '(..F..)'
  10 'data_file.1'
$ 10 'data_file.2'
$
$ If the above unit numbers are 10, data is read from this file
$ (no intermediate comment lines allowed),
$ This example is an ice concentration field.
$
   1. 1. 1. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0.
```

```
   1. 1. .5 .5 .5 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
   0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
   0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
$
$ This example is mean parameter assimilation data
$ First record gives number of data records, data are read as as
$ individual records of reals with recored length as given above
$
$  3
$ 1.5  1.6 0.70 10.3
$ 1.7  1.5 0.75  9.8
$ 1.9  1.4 0.77 11.1
$
$ ------------------------------------------------------------------- $
$ End of input file                                                   $
$ ------------------------------------------------------------------- $
```

end of example input file

Note that the four optional output files are specific to the *generic shell* of WWATCH, but are not processed by the actual wave model routines. These files are consequently not needed if the wave model routines are used in a different shell or in an integrated program. However, the routines reading and writing these files are system-independent and could therefore be used in customized applications of the basic wave model. The reading and writing of these files is performed by the subroutine W3FLDG (w3fldsmd.ftn). For additional documentation and file formats reference if made to this routine.

4.4.5 The generic shell

Program	:	ww3_shel		(W3SHEL)
Code	:	ww3_shel.ftn		
Input	:	ww3_shel.inp	(10)	Formatted input file for program.
		mod_def.ww3	(30)	Model definition file.
		restart.ww3	(30)	Restart file.
		nest.ww3*	(33)	Boundary conditions file.
		level.ww3*	(11)	Water levels file.
		current.ww3*	(12)	Current fields file.
		wind.ww3*	(13)	Wind fields file.
		ice.ww3*	(14)	Ice fields file.
		data0.ww3*	(15)	Assimilation data.
		data1.ww3*	(16)	Assimilation data.
		data2.ww3*	(17)	Assimilation data.
		track_i.ww3*	(22)	Output track information.
Output	:	standard out	(6)	Formatted output of program.
		log.ww3	(20)	Output log of wave model (see section 4.2).
		test.ww3*	(6/21)	Test output of wave model.
		restart*n*.ww3*	(30)	Restart file(s).
		nest*n*.ww3*	(34-42)	Nesting file(s).
		out_grd.ww3*	(31)	Raw output of gridded fields.
		out_pnt.ww3*	(32)	Raw output of spectra.
		track_o.ww3*	(23)	Raw output of spectra along tracks.
Scratch	:	ww3_shel.scratch	(90)	Formatted scratch file.

start of example input file

```
$ ------------------------------------------------------------------- $
$ WAVEWATCH III shell input file                                      $
$ ------------------------------------------------------------------- $
$ Define input to be used with flag for use and flag for definition
$ as a homogeneous field (first three only); seven input lines.
$
   F F     Water levels
   F F     Currents
   T T     Winds
```

```
   T       Ice concentrations
   F       Assimilation data : Mean parameters
   F       Assimilation data : 1-D spectra
   F       Assimilation data : 2-D spectra.
$
$ Time frame of calculations ------------------------------------------ $
$ - Starting time in yyyymmdd hhmmss format.
$ - Ending time in yyyymmdd hhmmss format.
$
   19680606 000000
   19680607 000000
$
$ Define output data --------------------------------------------------- $
$ Five output types are available (see below). All output types share
$ a similar format for the first input line:
$ - first time in yyyymmdd hhmmss format, output interval (s), and
$   last time in yyyymmdd hhmmss format (all integers).
$ Output is disabled by setting the output interval to 0.
$
$ Type 1 : Fields of mean wave parameters
$          Standard line and line with flags to activate output fields
$          as defined in section 2.4 of the manual. The second line is
$          not supplied if no output is requested.
$                               The raw data file is out_grd.ww3,
$                               see w3iogo.ftn for additional doc.
$
$
   19680606 000000   3600  19680608 000000
     T T T T T  T T T T T  T T T T T  T T T
$
$ Type 2 : Point output
$          Standard line and a number of lines identifying the
$          longitude, latitude and name (C*10) of output points.
$          The list is closed by defining a point with the name
$          'STOPSTRING'. No point info read if no point output is
$          requested (i.e., no 'STOPSTRING' needed).
$          Example for spherical grid.
$                               The raw data file is out_pnt.ww3,
$                               see w3iogo.ftn for additional doc.
$
$   NOTE : Spaces may be included in the name, but this is not
$          advised, because it will break the GrADS utility to
$          plots spectra and source terms, and will make it more
$          diffucult to use point names in data files.
$
```

```
   19680606 000000    900  19680608 000000
$
    -0.25 -0.25 'Land      '
     0.0   0.0  'Point_1   '
     2.0   1.0  'Point_2   '
     1.8   2.2  'Point_3   '
     2.1   0.9  'Point_4   '
     5.0   5.0  'Outside   '
$
     0.0   0.0  'STOPSTRING'
$
$ Type 3 : Output along  track.
$          Flag for formatted input file.
$                         The data files are track_i.ww3 and
$                         track_o.ww3, see w3iotr.ftn for ad. doc.
$
   19680606 000000   1800  19680606 013000
     T
$
$ Type 4 : Restart files (no additional data required).
$                               The data file is restartN.ww3, see
$                               w3iors.ftn for additional doc.
$
   19680606 030000      1  19680606 030000
$
$ Type 5 : Boundary data (no additional data required).
$                               The data file is nestN.ww3, see
$                               w3iobp.ftn for additional doc.
$
   19680606 000000   3600  20010102 000000
$
$ Testing of output through parameter list (C/TPAR) ------------------ $
$    Time for output and field flags as in above output type 1.
$
$  19680606 014500
$    T T T T T  T T T T T  T T T T T  T
$
$ Homogeneous field data ------------------------------------------- $
$ Homogeneous fields can be defined by a list of lines containing an ID
$ string 'LEV' 'CUR' 'WND', date and time information (yyyymmdd
$ hhmmss), value (S.I. units), direction (current and wind, oceanographic
$ convention degrees)) and air-sea temparature difference (degrees C).
$ 'STP' is mandatory stop string.
$
   'LEV' 19680606 010000    1.0
```

```
   'CUR' 19680606 073125    2.0    25.
   'WND' 19680606 000000   20.    145.    2.0
   'STP'
$
$ ------------------------------------------------------------------- $
$ End of input file                                                   $
$ ------------------------------------------------------------------- $
```

end of example input file

4.4.6 Gridded output post-processor

Program	:	ww3_outf		(W3OUTF)
Code	:	ww3_outf.ftn		
Input	:	ww3_outf.inp	(10)	Input file for gridded output post-processor.
		mod_def.ww3	(20)	Model definition file.
		out_grd.ww3	(20)	Raw gridded output data.
Output	:	standard out	(6)	Formatted output of program.
		...*	(50)	Transfer file.

start of example input file

```
$ ------------------------------------------------------------------- $
$ WAVEWATCH III Grid output post-processing                           $
$------------------------------------------------------------------- $
$ Time, time increment and number of outputs
$
  19680607 000000  3600. 1
$
$ Request flags identifying fields as in ww3_shel input and
$ section 2.4 fo the manual.
$
  F F F F F  T F F F F  F F F F F  F F F
$
$ Output type ITYPE [0,1,2,3]
$
  1
$ ------------------------------------------------------------------- $
$ ITYPE = 0, inventory of file.
$            No additional input, the above time range is ignored.
$
$ ------------------------------------------------------------------- $
$ ITYPE = 1, print plots.
$            IX,IY range and stride, flag for automatic scaling to
$            maximum value (otherwise fixed scaling),
$            vector component flag (dummy for scalar quantities).
$
  1 12 1 1 12 1 F T
$
$ ------------------------------------------------------------------- $
$ ITYPE = 2, field statistics.
$            IX,IY range.
$
```

```
$ 1 12 1 12
$
$ ------------------------------------------------------------------- $
$ ITYPE = 3, transfer files.
$            IX, IY range, IDLA and IDFM as in ww3_grid.inp.
$            The additional option IDLA=5 gives ia longitude, lattitude
$            and parameter value(s) per record (defined points only).
$
$ 2 11 2 11 5 2
$
$ For each field and time a new file is generated with the file name
$ ww3.yymmddhh.xxx, where yymmddhh is a conventional time idicator,
$ and xxx is a field identifier. The first record of the file contains
$ a file ID (C*13), the time in yyyymmdd hhmmss format, the lowest,
$ highest and number of longitudes (2R,I), id.  latitudes, the file
$ extension name (C*$), a scale factor (R), a unit identifier (C*10),
$ IDLA, IDFM, a format (C*11) and a number identifying undefined or
$ missing values (land, ice, etc.).  The field follows as defined by
$ IDFM and IDLA, defined as in the grid proprocessor. IDLA=5 is added
$ and gives a set of records containing the longitude, latitude and
$ parameter value. Note that the actual data is written as an integers.
$
$ ------------------------------------------------------------------- $
$ End of input file                                                   $
$ ------------------------------------------------------------------- $
```

end of example input file

4.4.7 Point output post-processor

Program	:	ww3_outp		(W3OUTP)
Code	:	ww3_outp.ftn		
Input	:	ww3_outp.inp	(10)	Input file for point output post-processor.
		mod_def.ww3	(20)	Model definition file.
		out_pnt.ww3	(20)	Raw point output data.
Output	:	standard out	(6)	Formatted output of program.
		tab*nn*.ww3 *	(*nn*)	Table of mean parameters where *nn* is a two-digit integer.
		. . . *	(user)	Transfer file.

start of example input file

```
$ ------------------------------------------------------------------- $
$ WAVEWATCH III Point output post-processing                          $
$-------------------------------------------------------------------- $
$ First output time (yyyymmdd hhmmss), increment of output (s),
$ and number of output times.
$
  19680606 030000  3600.  1
$
$ Points requested ------------------------------------------------- $
$ Define points for which output is to be generated.
$
  1
  2
  3
  4
$ mandatory end of list
 -1
$
$ Output type ITYPE [0,1,2,3]
$
  1
$ ------------------------------------------------------------------- $
$ ITYPE = 0, inventory of file.
$           No additional input, the above time range is ignored.
$
$ ------------------------------------------------------------------- $
$ ITYPE = 1, Spectra.
$          - Sub-type OTYPE :  1 : Print plots.
```

```
$                             2 : Table of 1-D spectra
$                             3 : Transfer file.
$          - Scaling factors for 1-D and 2-D spectra Negative factor
$            disables, output, factor = 0. gives normalized spectrum.
$          - Unit number for transfer file, also used in table file
$            name.
$          - Flag for unformatted transfer file.
$
    1   0.  0.  33  F
$
$ The transfer file contains records with the following contents.
$
$ - File ID in quotes, number of frequencies, directions and points.
$   grid name in quotes (for unformatted file C*21,3I,C*30).
$ - Bin frequenies in Hz for all bins.
$ - Bin directions in radians for all bins (Oceanographic conv.).
$                                                          -+
$ - Time in yyyymmdd hhmmss format                          | loop
$                                                 -+        |
$ - Point name (C*10), lat, lon, d, U10 and        |  loop    | over
$   direction, current speed and direction        |   over   |
$ - E(f,theta)                                     |  points  | times
$                                                 -+        -+
$
$ The formatted file is readable usign free format throughout.
$ This datat set can be used as input for the bulletin generator
$ w3split.
$
$ ------------------------------------------------------------------- $
$ ITYPE = 2, Tables of (mean) parameter
$          - Sub-type OTYPE :  1 : Depth, current, wind
$                              2 : Mean wave pars.
$                              3 : Nondimensional pars. (U*)
$                              4 : Nondimensional pars. (U10)
$                              5 : 'Validation table'
$          - Unit number for file, also used in file name.
$
$ 5  33
$
$ If output for one point is requested, a time series table is made,
$ otherwise the file contains a separate tables for each output time.
$
$ ------------------------------------------------------------------- $
$ ITYPE = 3, Source terms
$          - Sub-type OTYPE :  1 : Print plots.
```

```
$                              2 : Table of 1-D S(f).
$                              3 : Table of 1-D inverse time scales
$                                  (1/T = S/F).
$                              4 : Transfer file
$          - Scaling factors for 1-D and 2-D source terms. Negative
$            factor disables print plots, factor = 0. gives normalized
$            print plots.
$          - Unit number for transfer file, also used in table file
$            name.
$          - Flags for spectrum, input, interactions, dissipation,
$            bottom and total source term.
$          - scale ISCALE for OTYPE=2,3
$                              0 : Dimensional.
$                              1 : Nondimensional in terms of U10
$                              2 : Nondimensional in terms of U*
$                             3-5: like 0-2 with f normalized with fp.
$          - Flag for unformatted transfer file.
$
$ 1 -1. 0. 50   T F T F F F  0  F
$
$ The transfer file contains records with the following contents.
$
$ - File ID in quotes, nubmer of frequencies, directions and points,
$   flags for spectrum and source terms (C*21, 3I, 6L)
$ - Bin frequenies in Hz for all bins.
$ - Bin directions in radians for all bins (Oceanographic conv.).
$                                                          -+
$ - Time in yyyymmdd hhmmss format                          | loop
$                                              -+           |
$ - Point name (C*10), depth, wind speed and    |  loop     | over
$   direction, current speed and direction     |   over    |
$ - E(f,theta) if requested                     |  points   | times
$ - Sin(f,theta) if requested                   |           |
$ - Snl(f,theta) if requested                   |           |
$ - Sds(f,theta) if requested                   |           |
$ - Sbt(f,theta) if requested                   |           |
$ - Stot(f,theta) if requested                  |           |
$                                              -+          -+
$
$ ------------------------------------------------------------------- $
$ End of input file                                                   $
$ ------------------------------------------------------------------- $
```

end of example input file

The spectral data transfer file generated with ITYPE = 1 and OTYPE = 3 can be converted into a spectral bulletin using w3split (see section 5.2). This program reads the following five records from standard input (no comment lines allowed) :

- Name of output location.
- Identifier for run to be used in table.
- Name of input file.
- Logical identifying UNFORMATTED input file.
- Name of output file.

All above strings are read as characters using free format, and therefore need to be enclosed in quotes.

4.4.8 Track output post-processor

Program	:	ww3_trck		(W3TRCK)
Code	:	ww3_trck.ftn		
Input	:	track_o.ww3	(11)	Raw track output data.
Output	:	standard out	(6)	Formatted output of program.
		track.ww3	(51)	Formatted data file.

This post-processor does not require a formatted input file with program commands. It will simply convert the entire unformatted file to an integer compressed formatted file. The file contains the following header records :

- File identifier (character string of length 34).
- Number of frequencies and directions, first direction and directional increment (radians, oceanographical convention).
- Radian frequencies of each frequency bin.
- Corresponding directional bin size times frequency bin size to obtain discrete energy per bin.

For each output point the following records are printed :

- Date and time in `yyyymmdd hhmmss` format, longitude and latitude in degrees, and a status identifier 'ICE', 'LND' or 'SEA'. The following two records are written only for sea points.
- Water depth in meters, current and wind u and v components in meters per second, friction velocity in meters per second, air-sea temperature difference in degrees centigrade and scale factor for spectrum.
- The entire spectrum in integer packed format (can be read using free format).

4.4.9 GRIB output post-processor

Program	:	ww3_outf		(W3GRIB)
Code	:	ww3_grib.ftn		
Input	:	ww3_grib.inp	(10)	Input file for gridded output post-processor.
		mod_def.ww3	(20)	Model definition file.
		out_grd.ww3	(20)	Raw gridded output data.
Output	:	standard out	(6)	Formatted output of program.
		gribfile	(50)	GRIB file.

start of example input file

```
$ ------------------------------------------------------------------- $
$ WAVEWATCH III Grid output post-processing ( GRIB )                  $
$-------------------------------------------------------------------- $
$ Time, time increment and number of outputs.
$
  19680606 000000  3600. 3
$
$ Request flags identifying fields as in ww3_shel input and
$ section 2.4 of the manual.
$
  T T T T T  T T T T T  T T T T T  T T T
$
$ Aditional info needed for grib file
$   Forecast time, center ID, generating process ID, grid definition
$   and GDS/BMS flag
$
  19680606 010000 7 10 255 192
$
$ ------------------------------------------------------------------- $
$ End of input file                                                   $
$ ------------------------------------------------------------------- $
```

end of example input file

This post-processor packs fields of mean wave parameters in GRIB format, using GRIB version II and NCEP's w3 and bacio library routines. The GRIB packing is performed using the NCEP's GRIB tables as described in NCEP (1998). Because the w3 and bacio routine are not fully portable, they are not supplied with the code. The user will have to provide corresponding routines.

It is suggested that such routines are activated with additional WWATCH switches in the mandatory switch group containing the 'NOGRB' switch, as if presently the case with the NCEP routines.

Although the above input file contains flags for all 18 output fields of WWATCH, not all fields can be packed in GRIB. If a parameter is chosen for which GRIB packing is not available, a message will be printed to standard output. Table 4.1 shows which parameter can be packed in GRIB, and which kpds value has been used in packing the data. Note that there is a slight deviation from the standard WMO GRIB description in the peak and wind sea period, and the corresponding mean parameters. These are packed using KPDS values reserved for the 'primary' and 'secondary' period and direction. Note that these periods are simply obtained by inverting the corresponding frequency, which is the actual output of WWATCH.

WWATCH output number	description	KPDS number for GRIB packing
1	depth	–
2	mean current	–
3	wind speed	32
	wind direction	31
	wind u	33
	wind v	34
4	air-sea temp. dif.	–
5	friction velocity	–
6	wave height H_s	100
7	mean wave length	–
8	mean wave period T_m	103
9	mean wave direction θ_m	101
10	directional spread	–
11	peak period T_p	108
12	peak direction θ_p	107
13	wind sea period T_w	110
14	wind sea direction θ_w	109
15	average time step	–
16	cut-off frequency f_c	–
17	ice coverage	91
18	water level	–

Table 4.1: GRIB packing information for WWATCH. If no KPDS number is given, data cannot be packed in GRIB.

4.4.10 Gridded output post-processor for GrADS

Program	:	gx_outf		(GXOUTF)
Code	:	gx_outf.ftn		
Input	:	gx_outf.inp	(10)	Input file for gridded output post-processor.
		mod_def.ww3	(20)	Model definition file.
		out_grd.ww3	(20)	Raw gridded output data.
Output	:	standard out	(6)	Formatted output of program.
		ww3.grads	(50)	GrADS data file.
		ww3.ctl	(51)	GrADS control file.

start of example input file

```
$ ------------------------------------------------------------------- $
$ WAVEWATCH III Grid output post-processing ( GrADS )                 $
$-------------------------------------------------------------------- $
$ Time, time increment and number of outputs.
$
  19680606 000000  3600. 25
$
$ Request flags identifying fields as in ww3_shel input and
$ section 2.4 of the manual.
$
  F F T F F  T F F F F  T T F F F  F F F
$
$ Grid range in discrete counters IXmin,max, IYmin,max
$
  0 999 0 999
$
$ NOTE : In the Cartesian grid version of the code, X and Y are
$        converted to longitude and latitude assuming that 1 degree
$        equals 100 km if th maximum of X or Y is larger than 1000km.
$        For maxima between 100 and 1000km 1 degree is assumed to be
$        10km etc. Adjust labels in GrADS scripts accordingly.
$
$ ------------------------------------------------------------------- $
$ End of input file                                                   $
$ ------------------------------------------------------------------- $
```

end of example input file

This post-processor generates input files with gridded model parameters for

the Grid Analysis and Display System (GrADS, Doty, 1995). This graphical software can be obtained from http://www.iges.org/grads. Although GrADS can also work with GRIB files, the present preprocessor is preferable, as the data file also gives access to a land-sea-ice map.

4.4.11 Point output post-processor for GrADS

Program	:	gx_outp		(GXOUTP)
Code	:	gx_outp.ftn		
Input	:	gx_outp.inp	(10)	Input file for point output post-processor.
		mod_def.ww3	(20)	Model definition file.
		out_pnt.ww3	(20)	Raw point output data.
Output	:	standard out	(6)	Formatted output of program.
		ww3.spec.grads	(30)	GrADS data file sith spectra and source terms.
		ww3.mean.grads	(31)	File with mean wave parameters.
		ww3.spec.ctl	(32)	GrADS control file.

start of example input file

```
$ ------------------------------------------------------------------- $
$ WAVEWATCH III Point output post-processing ( GrADS )                $
$-------------------------------------------------------------------- $
$ First output time (yyyymmdd hhmmss), increment of output (s),
$ and number of output times.
$
  19680606 000000  10800. 9
$
$ Points requested ------------------------------------------------- $
$ Define points for which output is to be generated.
$
  1
  2
  3
  4
$ mandatory end of list
 -1
$
$ ------------------------------------------------------------------- $
$   Flags for plotting F, Sin, Snl, Sds, Sbt, Stot
$
  T T T T T T
$
$ NOTE : In the Cartesian grid version of the code, X and Y are
$        converted to km. Use source_xy.gs instead of source.gs
$
$ ------------------------------------------------------------------- $
```

```
$ End of input file                                                    $
$ ---------------------------------------------------------------------- $
```

end of example input file

This post-processor is intended to generate data files with which GrADS (see previous section) can plot polar plots of spectra and source terms. To achieve this, spectra and source terms are store as "longitude-latitude" grids. For each output point a different name is generated for the data, typically LOC*nnn*. When the data file is loaded in GrADS, the variable LOC001 will contain a spectral grid for the first requested output point at level 1, the input source term at level 2, etc. For the second output point the data is stored in LOC002 etc. The actual output point names are passed to GrADS through the control file ww3.spec.ctl. Wave heights and environmental data are obtained from ww3.mean.grads The user, however, need not be aware of the details of the GrADS data files and data storage. The GrADS scripts spec.gs, source.gs and 1source.gs are provided to automatically generate spectral plots from the output files of this post-processor.

Note: for the GrADS scripts to work properly, the names of the output points should not contain spaces.

5 Installing the wave model

5.1 Introduction

WWATCH is written in ANSI standard FORTRAN-90, with in essence no machine-dependent elements, so that WWATCH can be installed without modifications on most platforms. WWATCH utilizes its own preprocessor to process include files (presently only used for adding NCEP-specific documentation), to select model options at the compile level, and to switch test output on or off. This approach proved to be efficient during the development of WWATCH, but it complicates the installation of WWATCH. To minimize complications, a set of UNIX/Linux scripts is provided to automate the installation in general and the use of the preprocessor in particular. This option is not supported for other operation systems like MS or Mac products. If the code is to be compiled on one of the latter platforms, it is suggested to extract a working code in a UNIX/Linux environment using the utility `w3_source` (see below), and than to port this clean code to the platform of choice.

WARNING

If version 2.22 is implemented as an upgrade to previous versions of WWATCH, please note that this version is not compatible with previous model versions. It is therefore prudent *NOT* to install the new version of WWATCH on top of the old version. The new version should be installed in a new directory, or backup copies of the old system should be made, after which the old system should be removed completely. It is necessary to manually edit or remove the file .wwatch3.env (see below) if the new version is installed in a new directory.

WARNING

5.2 Installing files

In its packaged version, WWATCH is contained in several files:

install_wwatch3	The WWATCH install program.
wwatch3.aux.tar	Archive file containing programs and scripts controlling the compiling and linking of WWATCH.
wwatch3.ftn.tar	Archive file containing source code.
wwatch3.inp.tar	Archive file containing example input files.
wwatch3.tst.tar	Archive file containing test cases with results.

As the first step of installing WWATCH, these files have to be copied to a work directory on the machine on which WWATCH will be installed. Because this directory will be the 'home' directory of WWATCH, it is suggested that a new directory is created (see also warning in previous section). Furthermore install_wwatch3 has to be made executable by typing

```
chmod 700 install_wwatch3
```

after which the installation of the files is started by typing

```
install_wwatch3
```

WARNING

The install program will ask for a compiler to compile some auxiliary FORTRAN codes used in putting WWATCH together. Unlike the actual WWATCH source code, these programs are still written in FORTRAN-77. It is therefore sufficient to point towards the generic FORTRAN-77 compiler on the system. This choice of mixed codes was made to assure that if WWATCH is put on a UNIX/Linux box only to put the code together, it will not be necessary to buy a FORTRAN-90 compiler.

WARNING

When install_wwatch3 is executed for the first time, it will ask the user to identify the directory in which WWATCH will be installed. This has to be the directory in which the above five files reside. The program will then

print a message that it cannot find the setup file, and ask some questions. The default answer or options are shown in square brackets. After the user has verified that the setup parameters are satisfactorily, the install program will create the (hidden) setup file .wwatch3.env in the user's home directory. The setup can be modified by rerunning the install program, or by manually editing the setup file .wwatch3.env. The 'home' directory of WWATCH can only be changed by editing or removing .wwatch3.env.

After the setup file is processed, the install program asks if the user wants to continue with the installation. If the user chooses to continue, the program will look for the four archive files. If a file is found, the program asks if the contents of the file should be installed. If no files are found, the archive files do not reside in the home directory, or the home directory is erroneously defined. To avoid that the install program generates unwanted files and directories in such a case, abort the install program (type Ctrl C), and check the location of the archive files, and the 'home' directory of WWATCH (see previous paragraph).

After files to be unpacked have been identified, the program will ask if old files should be overwritten automatically. If the user chooses 'n', the program will ask permission to overwrite each file that already exists. Files that contain user specific information, such as compile and link options, will never be replaced by the install program.

As the first step of the actual installation, the install program checks if the following directories exist in the 'home' directory of WWATCH.

arc	Archive directory.
aux	Raw auxiliary programs (source codes etc.).
bin	Executables and shell scripts for compiling and linking.
exe	WWATCH executables.
ftn	Source code and makefile.
inp	Input files.
mod	Module files.
obj	Object files.
test	Scripts with test cases.
work	Auxiliary work directory.

All these directories are generated by the install program install_wwatch3, except for the archive directory, which is generated by arc_wwatch3 (see below).

If installation of the auxiliary programs is requested (file wwatch3.aux.tar), the install program will first process source codes of auxiliary programs, using

the compiler as defined by the user in the setup file. Note that these codes are still in fixed format FORTRAN-77.

w3adc.f	WWATCH FORTRAN preprocessor.
w3prnt.f	Print files (source codes) including page and line numbers.
w3list.f	Generate a generic source code listing.
w3split.f	Generate spectral bulletin identifying individual wave fields within a spectrum from the spectral output of the point output post-processor (see section 4.4.7).

The above source codes are stored in the directory aux and the executables are stored in the directory bin. A more detailed description of these programs (including instructions on running the executables) can be found in the documentation included in the above source code files. After the compilation of these programs, several UNIX shell scripts and auxiliary files are installed in the bin directory.

ad3	Script to run the preprocessor w3adc and the compile script comp for a given source code file.
ad3_test	Test version of ad3, showing modifications to original source file. This script does not compile code.
all_switches	Generates a list of all w3adc switches present in the source code files.
arc_wwatch3	Program to archive versions of WWATCH in the directory arc.
comp.gen	Generic compiler script. The actual compiler script comp will be copied from this script if it does not exists.
find_switch	Script to find WWATCH source code files containing compiler switches (or arbitrary strings).
link.gen	Generic linker script. Actual script is link.
list	Script to print source code listing using w3prnt.
ln3	Script to make symbolic link of source code file to work directory.
make_makefile.sh	Script to generate the of the makefile based on selections in the file switch).
switch.gen	Generic file with preprocessor switches (section 5.4).

w3_clean	Script to clean up work and scratch directories by removing files generated during compilation or test runs.
w3_make	Script to compile and link components of WWATCH using a makefile.
w3_new	Script to touch correct source code files to account for changes in compiler switches in combination with the makefile.
w3_source	Script to generate a true FORTRAN source code for any of he WWATCH program elements.

The use of these scripts is explained in section 5.3. After this, several GrADS scripts are installed in the aux directory.

cbarn.gs	Semi-standard GrADS script for displaying color bars.
colorset.gs	Script to define colors used in shading.
map2_n.gs	Script used in propagation test ww3_tp2.n, can be used as template for specialized map plots.
source[_xy].gs	Script for composite plot of spectra and source terms (2-D polar plots), separate version for Cartesian grid.
1source.gs	Script to plot single source term.
spec.gs	Script to plot spectra.

As the final step of processing wwatch3.aux.tar, some links between directories are established.

If the installation of the source code is requested (file wwatch3.ftn.tar), the install program will copy source code and include files to the directory ftn. All the files considered are discussed in detail in chapter 6.

If the installation of the input files is requested (file wwatch3.inp.tar), the install program will copy the example input files as presented in the previous chapter to the directory inp. Links are make to the work directory work. No other action is taken.

If the installation of the test cases is requested (file wwatch3.tst.tar), the install program will copy the test cases to the directory test (see section 5.6). No other action is taken.

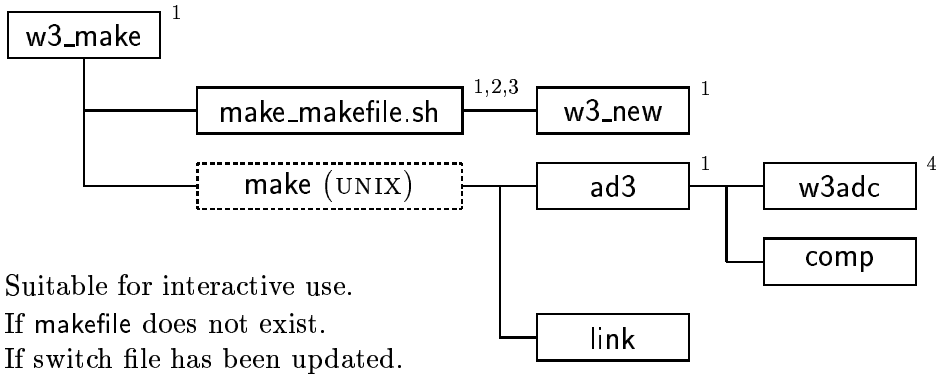

Figure 5.1: General layout of the compiler program w3_make.

Finally, the install program lists manual modifications required by or suggested to the user. These messages are printed only if the compile and link system are installed (file wwatch3.aux.tar).

5.3 Compiling and linking

Compilation of WWATCH is performed using the script w3_make in the bin directory[5]. If this script is used without parameters, all basic programs of WWATCH are compiled. Optionally, names of programs to be compiled can be given as part of the compile command. For instance

```
w3_make ww3_grid ww3_strt
```

will compile the grid preprocessor and the initial conditions program only. w3_make uses several of the scripts described in the previous section. A graphical representation is given in Fig. 5.1.

If necessary, the script w3_make uses the scripts make_makefile.sh to generate a makefile. make_makefile.sh generates a list of modules to be linked, based on the program switches in the file switch (see section 5.4), and checks

[5] Note that before running w3_make several user interventions are needed as described in the remainder of this section.

all needed sources for module dependencies. If switches have been changed since the last call to w3_make, w3_new is used to 'touch' relevant source code or to delete relevant object files. After the makefile has been completed, the standard UNIX make utility is used to compile and link the programs. Instead of directly using the FORTRAN compiler, the makefile invokes the preprocessor and compile scripts ad3 and comp, and the link script link. The script ad3 uses the extension of the file name to determine the necessary action. Files with extension .ftn are processed by w3adc, files with extension .f or .f90 are send to the script comp directly. Although a user could try out several of these scripts interactively, he or she generally needs to run w3_make only.

Before a first attempt is made at compiling, user intervention is required in three scripts/files. For convenience of debugging and development, links to these three files are made in the work directory work. The files in the work directory are

comp	Compiler script. This script requires the correct definition of the compiler and its options. Linked to ../bin/comp
link	Linker script. This script requires the correct definition of the linker and its options. Linked to ../bin/link
switch	File containing a list of switches as recognized by the preprocessor w3adc. Linked to ../bin/switch. The file provided with WWATCH should result in a hardware independent code.

After the appropriate changes have been made, (parts of) WWATCH can be compiled and linked. When the program is compiled for the first time, it is suggested to compile program parts one-by-one to avoid lengthy errors messages, and to set up error capturing in comp. A good place to start is compilation of the simple test code CTEST. First go to the directory work and make a link to the source code of this routine by typing

```
ln3 ctest
```

This link is made to facilitate later inclusion of errors to test or set-up error capturing in the script comp. The inner workings of the preprocessor w3adc can be seen by typing the command

```
ad3_test ctest
```

which will show how the actual source code is constructed from ctest.ftn, include files and program switches. Next, the compilation of this subroutine can be tested by typing

```
ad3 ctest 1
```

which invokes both the preprocessor w3adc and the compile script comp. The 1 at the end of this line activates test output. If it is omitted, this command should result in a single line of output, identifying that the routine is being processed. If ad3 works as expected, an object file obj/ctest.o is generated. If requested during the initial set up, a source code and listing file (ctest.f and ctest.l) can be found in the scratch directory. The listing file is also retained if compilation errors are detected by comp. At this time, it is prudent to test error capturing in the script comp by adding errors and warnings to ctest.ftn in the work directory. The error capturing is discussed in some detail in the documentation of comp. After comp has been tested, and the errors in ctest.ftn have been removed, the link to the work directory and the file obj/ctest.o can be deleted.

After a single routine has been compiled successfully, the next step is to try to compile and link an entire program. The grid preprocessor can be compiled by typing

```
w3_make ww3_grid
```

If the compilation appears successful, and if the input files have been installed (see above), the grid preprocessor can be tested by typing

```
ww3_grid
```

in the work directory. If the input files have been installed, a link to the input file ww3_grid.inp will be present in the work directory, and the grid preprocessor will run and send its output to the screen. Output files of the grid preprocessor will appear in the work directory. When a program is compiled for the first time, the operating system might not be able to find the executable. If this occurs, try to type

```
rehash
```

or open a new shell to work from. In this way all separate programs can be compiled and tested. To clean up all temporarily files (such as listings) and data files of the test runs, type

```
w3_clean
```

Note that w3_make only checks the switch file for changes. If the user changes the compile options in the compile and link scripts comp and link, it is advised to force the recompilation of the entire program. This can be achieved by typing

```
w3_new all or w3_new
```

before invoking w3_make. This might also be useful if the compilation is unsuccessful for no apparent reason.

Two additional remarks need to be made regarding parallel versions of the model (OpenMP and MPI versions). First, only the main program shell (ww3_shel) and the initial conditions program (ww3_strt) are parallelized. The parallel version of the latter program is only relevant if the single processor version takes too much memory to run. Many compilers do not allow for transparently and efficiently running parallel code on a single processor. The main program ww3_shel therefore needs to be compiled separately from the other programs. As the programs share subroutines, separate parallel and non-parallel objects are needed for shared routines. To avoid mixed linking, changes of parallel options in the switch file will therefore remove all old object modules. Proper compilation of the code for a parallel environment can thus be achieved in the following way. First set the compiler options in comp and link, and the options in switch for non-parallel compilation of the code and invoke

```
w3_make
```

to generate a non-parallel model version. Then set the proper parallel options in comp, link and switch, and compile only the main wave program ww3_shel by typing

```
w3_make ww3_shel
```

Alternatively, w3_source can be used to extract complete source codes for all programs, which can be compiled separately at will. Secondly, the OpenMP code should be compiled using directives only, i.e., do not use compiler options that automatically thread the code.

5.4 Selecting model options

The file switch in the bin and work directories contains a set of strings identifying model options to be selected. Many options are available. Of several groups of options it is mandatory to select exactly one. These mandatory switches are described in section 5.4.1. Other switches are optional, and are described in section 5.4.2. Default model setting are identified in section 5.4.3. The order in which the switches appear in switch is arbitrary. How these switches are included in the source code files is described in section 6.2.

5.4.1 Mandatory switches

Of each of the below groups of switches exactly one has to be selected. The first group of switches controls the selection of machine-dependent code. With the introduction of FORTRAN-90 this set of switches should have become obsolete. Problems with some compilers have prompted the retention of the second switch.

F90	FORTRAN-90 style date and time capturing and program abort.
DUM	Dummy to be used if WWATCH is to be installed on previously untried hardware.

Hardware model (first group) and message passing protocol (second group). Note that these two groups share a switch. This implies that the MPI switch can only be used in combination with the DIST switch.

SHRD	Shared memory model.
DIST	Distributed memory model.

SHRD	Shared memory model, no message passing.
MPI	Message Passing Interface (MPI).

Word length used to determine record length in direct access files

LRB4	4 byte words.
LRB8	8 byte words.

Selection of grid type:

LLG	Spherical grid.
XYG	Cartesian grid.

Selection of propagation schemes:

PR0	No propagation scheme used.
PR1	First order propagation scheme.
PR2	ULTIMATE QUICKEST propagation scheme with Booij and Holthuijsen (1987) dispersion correction.
PR3	ULTIMATE QUICKEST propagation scheme with Tolman (2002a) averaging technique.
PR4	ULTIMATE QUICKEST propagation scheme with Tolman (2002a) divergence technique.
PRX	Experimental (user supplied).

Selection of input and dissipation:

ST0	No input and dissipation used.
ST1	WAM3 source term package.
ST2	Tolman and Chalikov (1996) source term package. See also the optional STAB2 switch.
STX	Experimental (user supplied).

Selection of nonlinear interactions:

NL0	No nonlinear interactions used.
NL1	Discrete interaction approximation (DIA).
NL2	Exact interaction approximation (WRT).
NLX	Experimental (user supplied).

Selection of bottom friction:

BT0	No bottom friction used.
BT1	JONSWAP bottom friction formulation.
BTX	Experimental (user supplied).

Selection of method of wind interpolation:

WND0	No interpolation.
WND1	Linear interpolation.
WND2	Approximately quadratic interpolation.

Selection of method of current interpolation:

CUR1	Linear interpolation.
CUR2	Approximately quadratic interpolation.

Switch for user supplied GRIB package.

NOGRB	No package included.
NCEP1	NCEP package for IBM SP.

5.4.2 Optional switches

All switches below activate model behavior if selected, but do not require particular combinations. The following switches control optional output for WWATCH programs.

O0	Output of namelists in grid preprocessor.
O1	Output of boundary points in grid preprocessor.
O2	Output of the grid point status map in grid preprocessor.
O2a	Generation of land-sea mask file mask.ww3 in grid preprocessor.
O2b	Output of obstruction map in grid preprocessor.
O3	Additional output in loop over fields in field preprocessor.
O4	Print plot of normalized one-dimensional energy spectrum in initial conditions program.
O5	Id. two-dimensional energy spectrum.

O6	Id. spatial distribution of wave heights (not adapted for distributed memory).
O7	Echo input data for homogeneous fields in generic shell.
O7a	Diagnostic output for output points in generic shell.
O8	Filter field output for extremely small wave heights in wave model (useful for some propagation tests).
O9	Assign a negative wave height to negative energy in wave model. Used in testing phase of new propagation schemes.

The following switches enable parallelization of the model using OpenMP directives, also known as 'threading'. Note that in the present version of the model, threading and parallelization using the MPI switch cannot be used simultaneously.

OMP0	High level parallelization of calls to source term and propagation subroutines.
OMP1	Parallelization of loops in output and other processing.

Furthermore the following miscellaneous switches are available:

C90	Compiler directives for Cray C90 (vectorization).
DSS0	Switch off frequency dispersion in diffusive dispersion correction.
MPIT	Test output for MPI initializations.
NCO	Code modifications for operational implementation at NCO (NCEP Central Operations). Mostly changes unit numbers and file names. Not recommended for general use.
RWND	Correct wind speed for current velocity.
S	Enable subroutine tracing in the main WWATCH subroutines by activating calls to the subroutine STRACE.
SEED	Seeding of high-frequency energy.
STAB2	Enable stability correction (2.72) - (2.75)for Tolman and Chalikov (1996) source term package.
T	Enable test output throughout the program(s).
Tn	Id.
TDYN	Dynamic increment of swell age in diffusive dispersion correction (test cases only).
TPAR	Test of the output in the parameter list of W3WAVE.
VT	Instrumentation for vampirtrace under MPI.

XW0	Swell diffusion only in ULTIMATE QUICKEST scheme.
XW1	Id. wave growth diffusion only.

5.4.3 Default model settings

The default WWATCH is defined by the following selection of mandatory and optional switches. Mandatory switch groups for which no option is listed here do not influence model results, and their setting is therefore irrelevant with respect to the definition of a default version of WWATCH. The default model is defined by using the following switches :

LLG	Longitude-lattitude grid.
PR3	Propagation scheme (UQ with averaging).
ST2	Basic source terms (Tolman and Chalikov, 1996).
STAB2	Switching on retuning and stability correction in basic source terms.
NL1	Nonlinear interactions (DIA).
BT1	Bottom friction (JONSWAP).
WND1	Wind interpolation (linear).
RWND	Define wind as relative to current.
CUR1	Current interpolation (linear).
SEED	Seeding on.

The optional switches O8, O9, DSS0, TDYN, XW0 and XW1 modify model behavior and are *not* used in the default version of WWATCH. All optional switches not explicitly mentioned in this section do not influence model behavior and their setting is therefore irrelevant for the definition of the default model.

Default parameter settings for all model options have been presented in chapter 2. The model will automatically default to these values, unless they are explicitly overwritten using NAMELIST input provided by the user in the input file for the grid preprocessor (ww3_grid.inp). The only parameter setting for which no default setting is given is the swell age T_s in the propagation option PR2. The value of this parameter is set to 0, and needs to be overwritten by the user to turn the corresponding GSE alleviation method on.

5.5 Modifying the source code

Source code can obviously be modified by editing the source code files in the ftn directory. However, it is usually more convenient to modify source code files from the work directory work. This can be done by generating a link between the ftn and work directories. Such a link can be generated by typing

```
ln3 filename
```

where `filename` is the name of a source code or include file, with or without its proper extension. Working from the work directory is recommended for several reasons. First, the program can be tested from the same directory, because of similar links to the input files. Secondly, links to the relevant switch, compile and link programs are also available in this directory. Third, it makes it easy to keep track of files which have been changed (i.e., only those files to which links have been created might have been changed), and finally, source codes will not disappear if files (links) are accidentally removed from the work directory.

Modifying source codes is straightforward. Adding new switches to existing subroutines, or adding new modules requires modification of the automated compilation scripts. If a new subroutine is added to an existing module, no modifications are necessary. If a new module is added to WWATCH, the following steps are required to include it in the automatic compilation:

1) Add the file name to sections 2.b and c of the script make_makefile.sh to assure that the file is included in the makefile under the correct conditions.
2) Modify section 3.b of this script accordingly to assure that the proper module dependency is checked. Note that the dependency with the object code is checked, allowing for multiple or inconsistent module names in the file.
3) Run script interactively to assure that makefile is updated.

For details of inclusion, see the actual scripts. Adding a new switch to the compilation systems requires the following actions:

1) Put switch in required source code files.
2) If the switch is part of a new group of switches, add a new 'keyword' to w3_new.

3) Update files to be touched in w3_new if necessary.
4) Update make_makefile.sh with the switch and/or keyword.

These modifications need only be made if the switch selects program parts. For test output etc., it is sufficient to simply add the switch to the source code. Finally, adding an old switch to an additional subroutine requires these actions:

1) Update files to be touched in w3_new.

If WWATCH is modified, it is convenient to maintain copies of previous versions of the code and of the compilation scripts. To simplify this, an archive script (arc_wwatch3) is provided. This script generates tar files that can be reinstalled by the install program install_wwatch3. The archive files are gathered in the directory arc. The names of the archive files can contain user defined identifiers (if no identifier is used, the name will be identical to the original WWATCH files). The archive program is invoked by typing

```
arc_wwatch3
```

The interactive input to this script is self-explanatory. An archive file can be re-installed by copying the corresponding tar files to the WWATCH home directory, renaming them to the file names expected by the install program, and running the install program.

5.6 Running test cases

If WWATCH is installed and compiled successfully, it can be tested by running the different program elements interactively from the work directory. The switch settings in the generic switch file correspond to the activated inputs in the example input files. It should therefore be possible to run all model elements by typing

```
ww3_grid | more
ww3_strt | more
ww3_prep | more
ww3_shel | more
```

```
ww3_outf | more
ww3_outp | more
ww3_trck | more
ww3_grib | more
gx_outf | more
gx_outp | more
```

where the `more` command is added to allow for on-screen inspection of the output. Note that `ww3_grib` will only provide GRIB output if a user-supplied packing routine is linked in. GrADS can then be run from the work directory to generate graphical output for these calculations. All intermediate output files are placed in the work directory, and can be removed conveniently by typing

```
w3_clean
```

Also available are several test cases in the directory test. Example output files for selected runs are identified with the extension '.tar'. The documentation of each script identifies the preprocessor switches required to run the test case, and example outputs if available. The test cases are using the scratch directory as defined during the installation of WWATCH, and relevant output files will be saved in the directory test (this can easily be changed in the top of each test script). Hence, running a test case consists of five steps :

1) Look up the required switches in the documentation of the test cases, and set these switches in switch. Do not add optional switches like SEED unless specified explicitly.
2) Compile all programs by executing w3_make.
3) Execute the test script from the test directory test.
4) Check the output files in the test directory.
5) Clean up by executing w3_clean.

Note that all test cases expect that the code is compiled for a single processor using the shared memory model. The OpenMP and MPI versions of the model can also be tested with the standard test cases, and should give identical results. Converting a test case to run in parallel mode requires two modifications to the above list

1) Only the main program ww3_shel should be compiled with the parallel options. When changing compiler settings in comp and

link make sure to recompile all necessary modules by issuing the w3_new command before compilation.

2) Modify the test script so that only ww3_shel runs in the proper parallel environment. Because this environment is system dependent, this option has not been build into the script.

6 System documentation

6.1 Introduction

In this chapter a brief system documentation is presented. Discussed are the custom preprocessor used by WWATCH (section 6.2), the contents of the different source code files (section 6.3), optimization (section 6.4), and the internal data storage (section 6.5). For a more elaborate documentation, reference is made to the source code itself, which is fully documented.

6.2 The preprocessor

The WWATCH source code files are not ready to use FORTRAN files; mandatory and optional program options still have to be selected, and test output may be activated[6]. Compile level options are activated using 'switches'. The arbitrary switch 'SWT' is included in the WWATCH files as comment of the form !/SWT, where the switch name SWT is followed by a space or by a '/'. If a switch is selected, the preprocessor removes the comment characters, thus activating the corresponding source code line. If '/' follows the switch, it is also removed, thus allowing the selective inclusion of hardware-dependent compiler directives etc. The switches are case sensitive, and available switches are presented in section 5.4. Files which contain the switch C/SWT can be found by typing

```
find_switch '!/SWT'
```

A list of all switches included in the WWATCH files can be obtained by typing

```
all_switches
```

[6] Exceptions are some modules that are not originally part of WWATCH, like the exact interaction modules. Such modules with the extension .f of .f90 bypass the preprocessor and get copied to the work directory with the .f extension.

```
0 1
'w3wavemd.ftn'        'w3wavemd.f'
'F90 LRB4 SHRD NOGRB LLG PR1 ST1 NL2 BT0 WND1 CUR1'
'!docb_w3wavemd.doc'  'w3wavemd.doc'
'!docb_w3wave.doc'    'w3wave.doc'
'!docb_w3init.doc'    'w3init.doc'
'!docb_w3gath.doc'    'w3gath.doc'
'!docb_w3scat.doc'    'w3scat.doc'
'!docb_w3nmin.doc'    'w3nmin.doc'
'!docb_w3mpii.doc'    'w3mpii.doc'
'!docb_w3mpio.doc'    'w3mpio.doc'
```

Figure 6.1: Example input for W3ADC.

Pre-processing is performed by the program `w3adc`. This program is found in the file `w3adc.f`, which contains a ready to compile FORTRAN source code and a full documentation[7]. Various properties of `w3adc` are set in PARAMETER statements in `w3adc.f`, i.e., the maximum length of switches, the maximum number of include files, the maximum number of lines in an include file and the line length. `w3adc` reads its 'commands' from standard input. An example input file for `w3adc` is given in figure 6.1. Line-by-line, the input consists of

- → Test indicator and compress indicator
- → File names of the input and output code
- → Switches to be turned on in a single string (see section 5.4)
- → Multiple lines with include string identifying an include file and name of this file.

A test indicator 0 disables test output, and increasing values increase the detail of the test output. A compress indicator 0 leaves the file as is. A compress indicator 1 results in the removal of all comment lines indicated by '!', except for empty switches, i.e., lines starting with '!/'. A compress

[7] Presently still in fixed-format FORTRAN-77.

indicator 2 results in the subsequent removal of all comments. Comment lines are not allowed in this input file. The above input for `w3adc` is read using free format. Therefore quotes are needed around strings. Echo and test output is send to the standard output device. To facilitate the use of the preprocessor, several UNIX scripts are provided with WWATCH as discussed in section 5.3. Note that compiler directives are protected from file compression by defining them using a switch.

Note that the present version of `w3adc` assumes that the computer system allows for file identification by name. If this option is not available on the system, the program is easily adapted to read by unit number.

6.3 Program files

The WWATCH source code files are stored in files with the extension ftn[8]. Starting with version 2.00, the code has been organized in modules. This allows for bundling of variables with the their subroutines, which in turn makes management of alternative numerics and physics packages easier. Only the main programs are not packaged in modules. The subroutines contained in the modules are described in some detail below. The relation between the various subroutines is graphically depicted in Figs. 6.2 through 6.5.

6.3.1 Wave model modules

At the core of the wave model is the wave model module.

Main wave model module — w3wavemd.ftn

w3wave	The actual wave model W3WAVE.
w3init	The initialization routine W3INIT, which prepares the wave model for computations (internal).
w3gath	Data transpose to gather data for spatial propagation in a single array (internal).
w3scat	Corresponding scatter operation (internal).
w3nmin	Calculate minimum number of sea points per processor (internal).

[8] with the exception of some modules provided by others.

w3mpii	MPI initialization (internal).
w3mpio	MPI initialization for I/O (internal).

There are presently four propagation schemes available, as well as a 'slot' for a user supplied propagation routine. The propagation schemes are packaged in the following modules.

Propagation module (first order) w3pro1md.ftn

w3map1	Generation of auxiliary maps.
w3xyp1	Propagation in physical space.
w3ktp1	Propagation in spectral space.

Propagation module (generic ULTIMATE QUICKEST) w3uqckmd.ftn

w3qck$n$	Routines performing ULTIMATE QUICKEST scheme in arbitrary spaces (1: regular grid. 2: irregular grid 3: regular grid with obstructions).

Propagation module (UQ scheme with diffusion) w3pro2md.ftn

w3map2	Generation of auxiliary maps.
w3xyp2	Propagation in physical space.
w3ktp2	Propagation in spectral space.

Propagation module (UQ scheme with averaging) w3pro3md.ftn

w3map3	Generation of auxiliary maps.
w3mapt	Generation of transparency maps.
w3xyp3	Propagation in physical space.
w3ktp3	Propagation in spectral space.

Propagation module (UQ scheme with divergent advection) w3pro4md.ftn

w3map4	Generation of auxiliary maps.
w3xyp4	Propagation in physical space.
w3ktp4	Propagation in spectral space.

Propagation module (slot for user supplied routines) w3proxmd.ftn

w3xypx	Propagation in physical space.
w3ktpx	Propagation in spectral space.

The source term calculation and integration is contained in several modules. The module w3srcemd.ftn manages the general calculation and integration. Additional modules contain the actual source term options.

Source term integration module w3srcemd.ftn

w3srce	Integration of source terms.

Input and dissipation module (WAM-3) w3src1md.ftn

w3spr1	Calculation of mean wave parameters (single grid point).
w3sin1	Calculation of S_{in}.
w3sds1	Calculation of S_{ds}.

Input and dissipation module (Tolman and Chalikov, 1996) w3src2md.ftn

w3spr2	Calculation of mean wave parameters (single grid point).
w3sin2	Calculation of S_{in}.
w3sds2	Calculation of S_{ds}.
inptab	Generation of the interpolation table for β.
w3beta	Function to calculate β (internal).

Input and dissipation module (slot for user supplied routines) w3srcxmd.ftn

w3sinx	Calculation of S_{in}.
w3sdsx	Calculation of S_{ds}.

Nonlinear interaction module (DIA) w3snl1md.ftn

w3snl1	Calculation of S_{nl}.
insnl1	Initialization for S_{nl}.

Nonlinear interaction module (WRT) w3snl2md.ftn

w3snl2	Interface routine for S_{nl}.

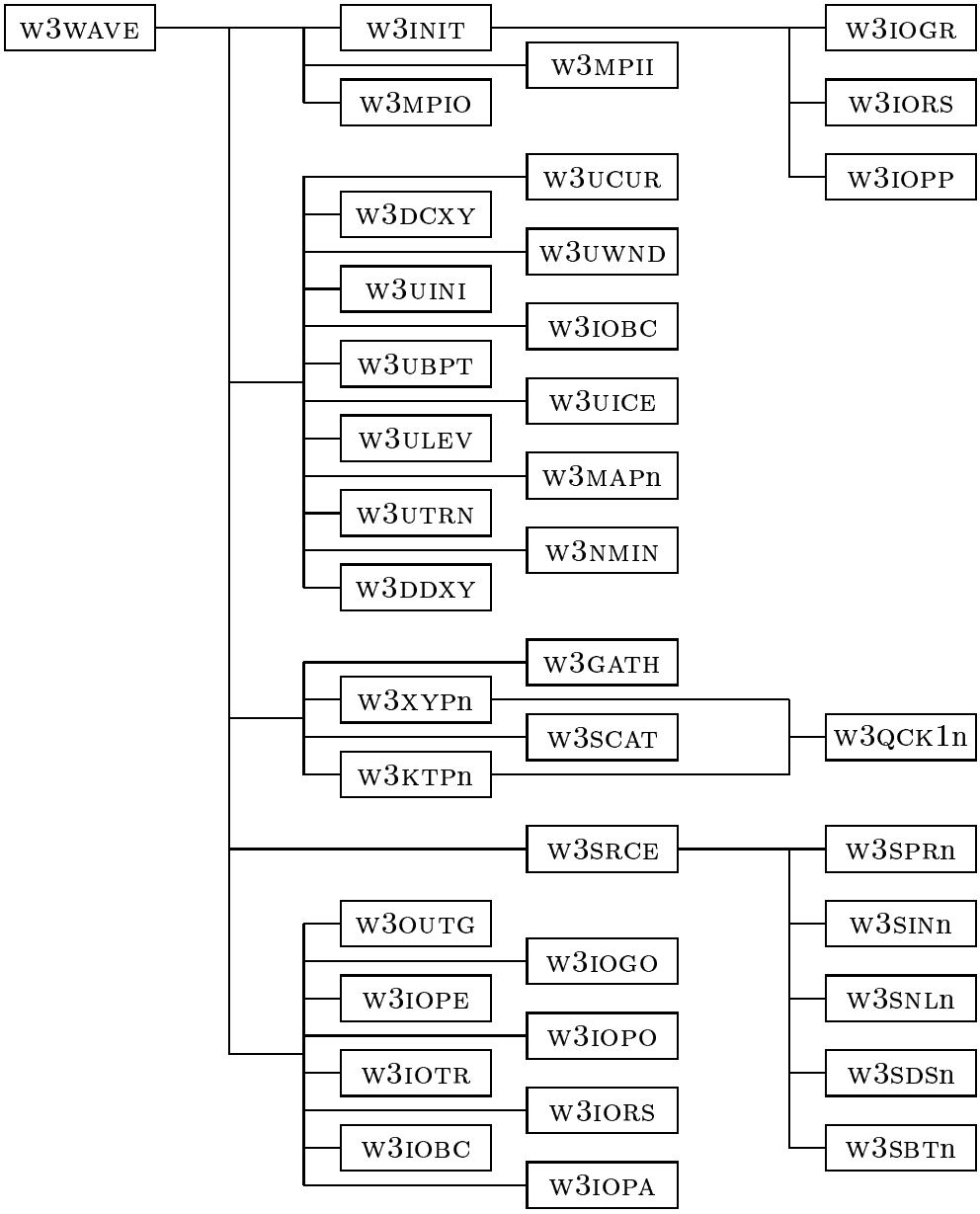

Figure 6.2: Subroutine structure for wave model routine without service routines. Note that W3IOGR on reading data in W3INIT calls all necessary initialization routines for interpolation tables and phsyics parameterizations.

insnl2	Initialization for S_{nl}.

These routines provide the interface to the WRT routines. The WRT routines are provided in the files mod_constants.f90, mod_fileio.f90, mod_xnl4v4.f90, and serv_xnl4v4.f90. For details on these files, see Van Vledder (2002b).

Nonlinear interaction module (slot for user supplied routines) w3snlxmd.ftn

w3snlx	Calculation of S_{nl}.

Bottom friction module (JONSWAP) w3sbt1md.ftn

w3bt1	Calculation of S_{bot}.

Bottom friction module (slot for user supplied routines) w3sbtxmd.ftn

w3sbtx	Calculation of S_{bot}.

The input fields such as winds and currents are transferred to the model through the parameter list of W3WAVE. The information is processed within W3WAVE by the routines in the following module.

Input update module w3updtmd.ftn

w3ucur	Interpolation in time of current fields.
w3uwnd	Interpolation in time of wind fields.
w3uini	Generate initial conditions from the initial wind field.
w3ubpt	Updating of boundary conditions in nested runs.
w3uice	Updating of the ice coverage.
w3ulev	Updating of water levels.
w3utrn	Updating grid box transparencies.
w3ddxy	Calculation of spatial derivatives of the water depth.
w3dcxy	Calculation of spatial derivatives of the currents.

There are six types of WWATCH data files (other than the preprocessed input fields, which are part of the program shall rather than the actual wave model). The corresponding routines are gathered in six modules.

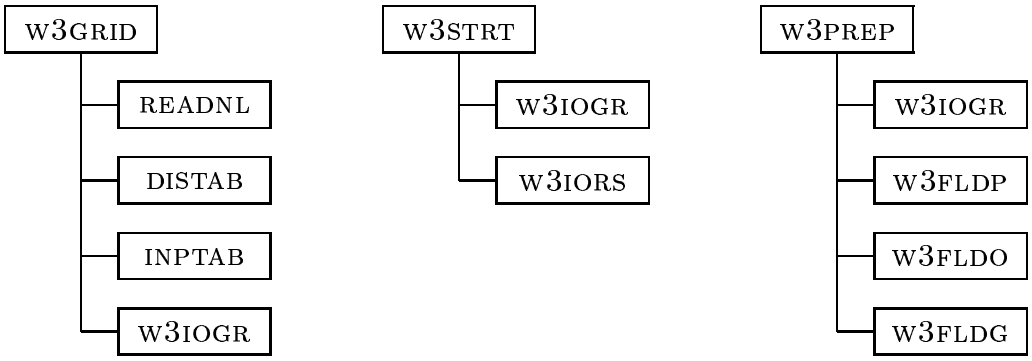

Figure 6.3: Subroutine structure of pre-processors

I/O module (mod_def.ww3) w3iogrmd.ftn

w3iogr	Reading and writing of mod_def.ww3.

I/O module (out_grd.ww3) w3iogomd.ftn

w3outg	Calculation of gridded output parameters.
w3iogo	Reading and writing of out_grd.ww3.
w3iopa	Providing gridded output to main program.

I/O module (out_pnt.ww3) w3iopomd.ftn

w3iopp	Processing of requests for point output.
w3iope	Calculating point output data.
w3iopo	Reading and writing of out_pnt.ww3.

I/O module (track_o.ww3) w3iotrmd.ftn

w3iotr	Generate track output in track_o.ww3.

I/O module (restart.ww3) w3iorsmd.ftn

w3iors	Reading and writing of restart$n$.ww3.

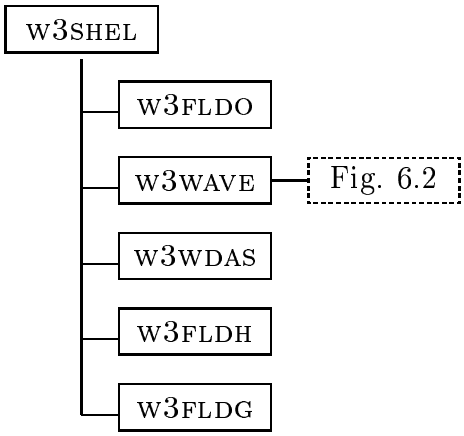

Figure 6.4: Subroutine structure for the generic program shell.

I/O module (nest.ww3) w3iobcmd.ftn

w3iobc	Reading and writing of nest*n*.ww3.

To complete the actual wave model, three modules containing variables only, and four modules with service routines are needed. For a description of the variables see section 6.5. For the actual contents of the service modules see the documentation in the source code files.

constants.ftn	Physical and mathematical constants
w3testmd.ftn	Test information such as unit numbers of output files and processor ID's.
w3parmmd.ftn	Auxiliary parameters for the discrete spectrum.
w3dispmd.ftn	Routines to solve the dispersion relation, including interpolation tables.
w3timemd.ftn	Time management routines.
w3servmd.ftn	General service routines.
w3arrymd.ftn	Array manipulation routines including 'print plot' routines.

6.3.2 Data assimilation module

WAVEWATCH III includes a data assimilation module that can work in conjunction with the main wave model routine, and is integrated in the generic program shell. The module is intended as an interface to a data assimilation package to be provided by the user.

Data assimilation module w3wdasmd.ftn

w3wdas	Data assimilation interface.

6.3.3 Auxiliary programs

WAVEWATCH III has several auxiliary pre- and post-processors, and a generic stand-alone shell (see section 4.4). These main programs and some additional routines are stored in the following files. Generally, subroutines used only by the programs are stored as internal subroutines with the main program. There is no need for using the module structure in this case. The exception is an additional module w3fldsmd.ftn which deals with the data flow of input fields for the wave model between the field pre-processor and the stand-alone model shell. The latter module does not have any explicit WWATCH dependencies, and can therefore be integrated in any custom data pre-processor.

Input data file management module w3fldsmd.ftn

w3fldo	Opening and checking of data files for W3SHEL.
w3fldg	Reading and writing of data files for W3SHEL (model input).
w3fldd	Reading and writing of data files for W3SHEL (data assimilation).
w3fldp	Prepare interpolation of input fields from arbitrary grids.
w3fldh	Management of homogeneous input fields in W3SHEL.

Grid pre-processing program ww3_grid.ftn

w3grid	The grid preprocessor.

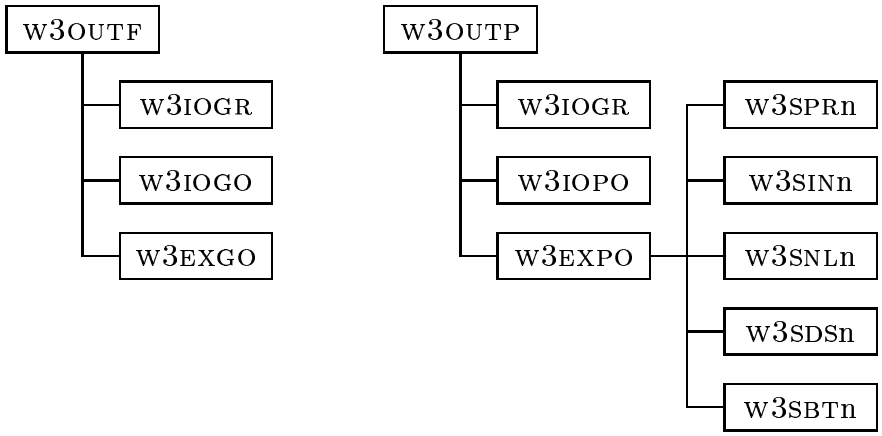

Figure 6.5: Subroutine structure for the postprocessors. The structure of the GRIB postprocessor is similar to W3OUTF, replacing W3OUTF and W3EXGO with W3GRIB and W3EXGB, respectively. The structure of the GrADS postprocessors is similar, replacing W3OUTF, W3OUTP, W3EXGO and W3EXPO with GXOUTF, GXOUTP, GXEXGO and GXEXPO, respectively

---

readnl	Reading NAMELIST input (internal).

Initial conditions program — ww3_strt.ftn

w3strt	The initial conditions program.

Input field pre-processing program — ww3_prep.ftn

w3prep	Pre-processor for the input fields for the generic shell.

Generic wave model program — ww3_shel.ftn

w3shel	The generic program shell.

Gridded data post-processing program — ww3_outf.ftn

w3outf	The post-processing program for gridded fields of mean wave parameters.
w3exgo	Actual output routine (internal).

Point post-processing program ww3_outp.ftn

w3outp	The post-processing program output at selected locations.
w3expo	Actual output routine (internal).

Track output post-processing program ww3_trck.ftn

w3trck	Converting unformatted direct access track output file to integer-packed formatted file.

Gridded data post-processing program (GRIB) ww3_grib.ftn

w3grib	The post-processing program for generating GRIB files.
w3exgb	Actual output routine (internal).

Gridded data post-processing program (GrADS) gx_outf.ftn

gxoutf	The post-processing program for converting gridded fields of mean wave parameters to input files for GrADS.
gxexgo	Actual output routine (internal).

Point post-processing program (GrADS) gx_outp.ftn

gxoutp	The post-processing program for converting output at selected locations to input files for GrADS.
gxexpo	Actual output routine (internal).

6.4 Optimization

The source code of WWATCH is written in ANSI standard FORTRAN-90, and has been compiled and run on a variety of platforms ranging from PC's to supercomputers.

Optimization for vector computers has been performed by structuring the code in long vector loops where possible. Optimization was originally performed for the Cray YMP and C90. Note that some compiler directives for vectorization have been used. Note also that the vector optimization has not been updated since about 1997, and therefore needs to be revisited if

the model is implemented on a vector machine. Vectorization directives are activated by the corresponding preprocessor switch (C90).

Parallelization for shared memory machines using threading has been implemented using standard OpenMP directives. Such parallelization takes place mainly in the loop calling the source term routine W3SRCE and the different propagation routines. OpenMP directives are activated by the corresponding preprocessor switches (OMPn).

Parallelization for distributed memory machines is discussed in some detail in section 6.5.2.

Note that an important part of the optimization is the use of interpolation tables for the solution of the dispersion relation and for the calculation of the wind-wave interaction parameter).

6.5 Internal data storage

6.5.1 Grids

For convenience and economy of programming, spatial and spectral grids are considered separately. This approach is inspired by the splitting technique described in chapter 3. For spatial propagation, a simple 'rectangular' spatial grid is used, as is illustrated in Fig. 6.6. The grid can either be a Cartesian '(x, y)' grid or a spherical grid. In a spherical grid, the longitudes are denoted throughout the program by the counter IX, and latitudes by the counter IY, and the corresponding grid dimensions (NX,NY). All spatial field arrays are dynamically allocated within the code, corresponding work arrays are usually automatic, to allow for thread-safe code. The closure of the grid in case of a global applications is handled within the model, and does not require user intervention. To simplify the calculation of derivatives of in particular the current, the outer grid points (IX=1,NX, unless the grid is global) and (IY=1,NY) will be considered as land points. The minimum grid size therefore is NX=3, NY=3. Input arrays are typically assumed to be of the form

$$\text{ARRAY(NX,NY)} \ ,$$

and are read row by row (see also chapter 4). Within the program, however,

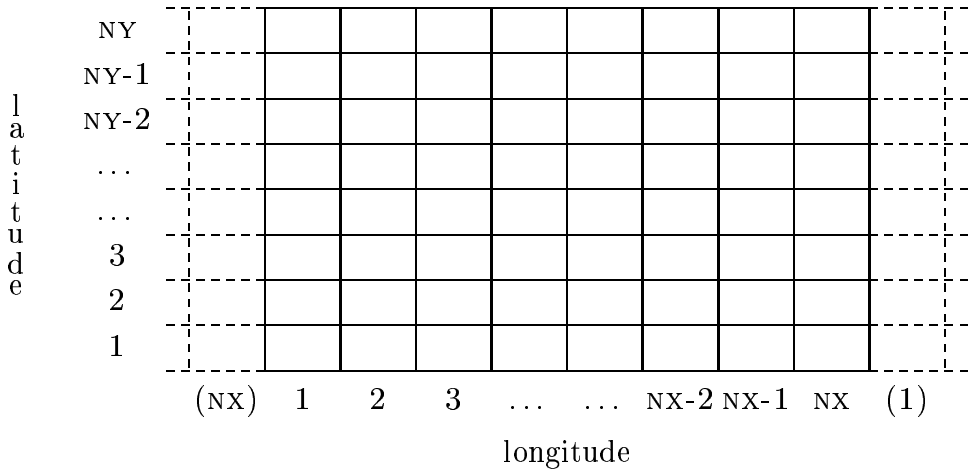

Figure 6.6: Layout of the spatial grid. Grid points are denoted as boxes, dotted boxes denoted repeated collumns for global model applications.

---

such two-dimensional array are usually treated as one-dimensional arrays, to increase vector lengths. To assure the extending the arrays for global closure (dashed grid boxes in Fig. 6.6) does not influence this one-dimensional counter IXY, the array ARRAY, its one-dimensional equivalent VARRAY and IXY are defined as

$$\text{ARRAY(MY,MX)} \ , \ \text{VARRAY(MY*MX)} \ ,$$
$$\text{IXY} = \text{IY} + (\text{IX-1})*\text{MY} \ .$$

Note that this representation of the grid is used *internally* within the model only.

The spectral grid for a given spatial grid point (IX,IY) is defined similarly, using a directional counter ITH and a wavenumber counter IK (Fig. 6.7). The size of the spectral grid is set using dynamic allocation. As with the spatial grid, the internal description of the spectrum A is defined as

$$\text{A(NTH,NK)} \ ,$$

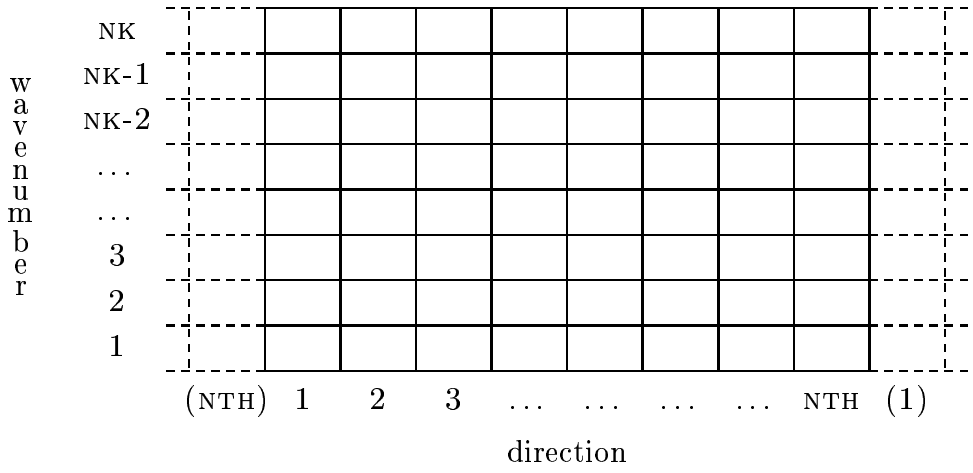

Figure 6.7: Layout of the spectral grid. Dotted boxes denoted repeated collumns for directional closure.

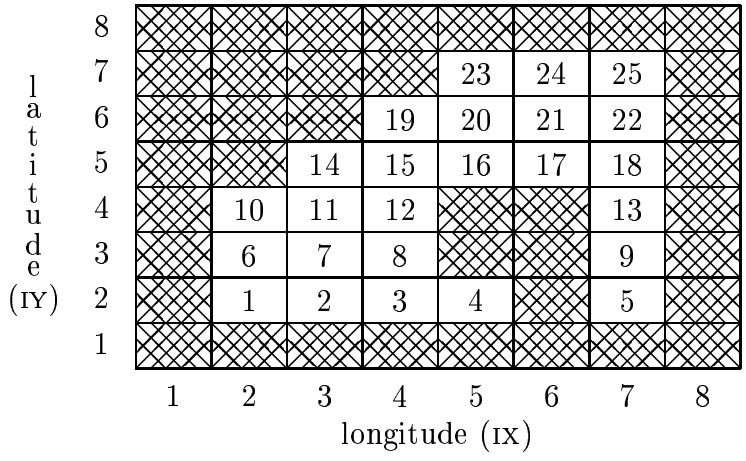

Figure 6.8: An example of the onedimensional storage grid for spectra. Hatched grid boxes denote land points. Numbers within the grid boxes show the grid counter ISEA of the storage grid.

and equivalent one-dimensional arrays are used throughout the program. Inside the model, directions are always Cartesian, $\theta = 0^\circ$ corresponds to propagation from east to west (positive x or IX direction), and $\theta = 90^\circ$ corresponds to propagation from south to north (positive y or IY direction). Output directions use other conventions, as is discussed in chapter 4.

The storage of the wave spectra accounts for the majority of the memory required by the model, because the splitting technique used assures that any part of the model operates on a small subset of the entire wave field. To minimize the amount of memory needed, only spectra for actual sea points are stored. Sea points are here defined as points where spectra are potentially needed. This includes active boundary points, and sea points covered by ice. For archiving purposes, a one-dimensional sea point grid is defined using the counter ISEA. Spectra are then stored as

$$\mathrm{A(ITH,IK,ISEA)} \ .$$

An example of the layout of this storage grid in relation to the full grid of Fig. 6.6 is given in Fig. 6.8. Obviously, the relation between the storage grid and the full spatial grid requires some bookkeeping. For this purpose, two 'maps' MAPFS and MAPSF are defined.

$$\begin{aligned} \mathrm{MAPSF(ISEA,1)} &= \mathrm{IX} \ , \\ \mathrm{MAPSF(ISEA,2)} &= \mathrm{IY} \ , \\ \mathrm{MAPSF(ISEA,3)} &= \mathrm{IXY} \ , \\ \mathrm{MAPFS(IY,IX)} = \mathrm{VMAPFS(IXY)} &= \mathrm{ISEA} \ , \end{aligned}$$

where MAPFS(IY,IX) $= 0$ for land points. Finally, a status map MAPSTA(IY,IX) is maintained to identify sea, land, active boundary and ice points.

MAPSTA(IY,IX) $= 0$	for land points,
MAPSTA(IY,IX) $= 1$	for sea points,
MAPSTA(IY,IX) $= 2$	for active boundary points.

Sea points and active boundary point which are not considered in the wave model due to the presence of ice are marked by their corresponding negative status indicator (-1 or -2).

6.5.2 Distributed memory concepts.

The general grid structure described in the previous paragraph is used for both shared and distributed memory versions of the model, with some minor differences. For the distributed memory version of the model, not all data is kept at each processor. Instead, each spectrum is kept at a single processor only. The spectra on the storage grid are distributed over the available processors with a constant stride. Because only part of the spectra are stored locally on a given processor, a distinction needs to be made between the above global sea point counter ISEA, and the local sea point counter JSEA. If the actual number of processors is NAPROC, and if IAPROC is the processor number ranging form 1 to NAPROC, these parameters are related in the following way

$$\begin{aligned} \text{ISEA} &= \text{IAPROC} + (\text{JSEA-}1)\ \text{NAPROC} \ , \\ \text{JSEA} &= 1 + (\text{ISEA-}1)\ /\ \text{NAPROC} \ , \\ \text{IAPROC} &= 1 + \text{MOD}(\text{ISEA-}1,\text{NAPROC}) \ . \end{aligned}$$

With this data distribution, source terms and intra-spectral propagation can be calculated at the each given processor without the need for communication between processors. For spatial propagation, however, a data transpose is required where the spectral components (ITH,IK) for all spatial grid points have to be gathered at a single processor. After propagation has been performed, the modified data have to be scattered back to their 'home' processor. Individual spectral components are assigned to specific processors in such a way that the number of partial propagation steps to be performed by each processor is roughly identical. This makes a good load balance possible. The actual algorithm can be found in section 4.d of the subroutine W3INIT (w3wavemd.ftn).

The data transpose for the gather operation is implemented in two steps using the Message Passing Interface (MPI) standard (e.g. Gropp et al., 1997). First, values for each spatial grid point for a given spectral bin (ITH,IK) are gathered in a single target processor in a one-dimensional array STORE(ISEA), which then is converted to the full two-dimensional field of spectral components. After propagation has been performed, the transpose for the scatter operation reverses this process, using the same one-dimensional array STORE. Whereas the algorithm for distributing spatial propagation over individual

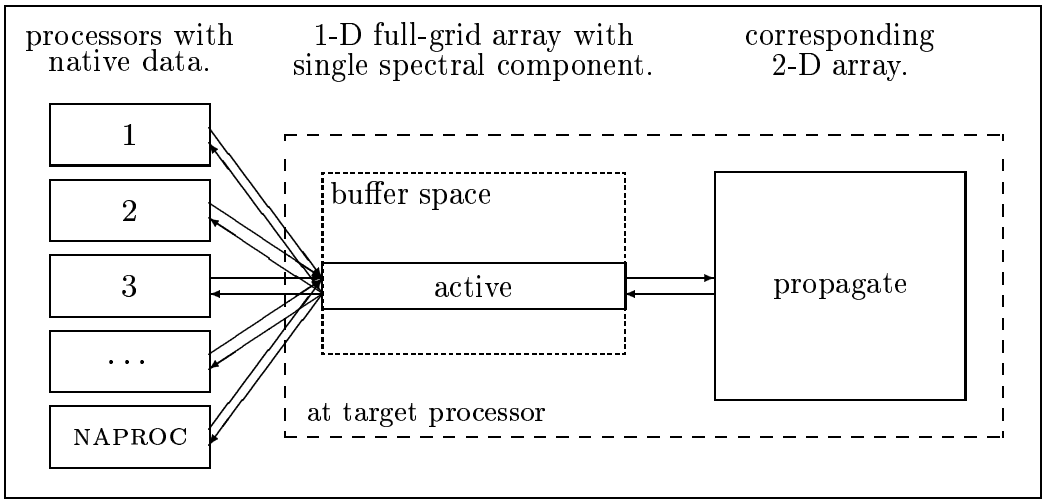

Figure 6.9: Data transpose in distributed memory model version. First, the data is moved from left to right in the figure during the gather operation. After the calculation is performed, the data is moved from right to left in the scatter operation.

processors assures a global (per time step) load balance, it does not assure that communication is synchronized, because not each calculation at each processor will take the same effort. To avoid that this results in a load imbalance, non-blocking communication has been used. Furthermore, the one-dimensional array STORE(ISEA) is replaced by STORE(ISEA,IBUF), where the added dimension of the array supplies an actively managed buffer space (see W3GATH and W3SCAT in w3wavemd.ftn). These buffers allow that spare clock cycles as may occur during communication can be used for calculation, and that hiding of communication behind calculation will occur if the hardware is capable of doing this. The buffered data transposes are graphically depicted in Fig. 6.9. More details can be found in Tolman (2002b)

In principle only the storage array A(ITH,IK,JSEA) is influenced by the data distribution. Input fields, maps and output fields of mean wave parameters in principle are retained at full resolution at each grid point. Full maps are available at each processor at each phase of the calculation. Input and output fields generally contain pertinent data at the stride NAPROC only.

Distributed memory also requires modifications to the I/O. Input files are

read completely by each separate processor. Restart and track output files have been converted to direct access files, so that each processor can write its part to file directly without the need for gathering all spectra at a single processor. Presently, the record length of these files is set to the minimum record length required for the data, without consideration for optimal or required record lengths. On NFS mounted file systems, this may cause errors in writing the files from multiple processors at the same time. For gridded output, point output and boundary data files, the data is gathered in a single processor using MPI, and than written by this processor to an unformatted file without changes relative to previous model versions. Some potential problems with I/O on distributed systems could be avoided by assigning one processor to be the 'data server'. This approach is presently not used, but might be considered for future releases.

The present algorithm for data distribution has been chosen for several reasons. First, it results in an automatic and efficient load balancing with respect to the (dynamic) integration of source terms, the exclusion of ice covered grid points, and of intra-spectral propagation. Secondly, the communication by definition becomes independent of the numerical propagation scheme, unlike for the more conventional domain decomposition. In the latter case, only a so-called 'halo' of boundary data needs to be converted to neighboring 'blocks' of grid points. The size of the halo depends on the propagation scheme selected. The main disadvantage of the present data distribution scheme is that the amount of data to be communicated each time step is much larger than for a more conventional domain decomposition, particularly when relatively small numbers of processors are used. On an IBM RS6000 SP, on which the distributed memory version of WWATCH was tested, the relatively large amount of communication did not constitute a significant part of the overall time of computation, and the model shows excellent scaling behavior for up to O(100) processors (Tolman, 2002b).

6.5.3 Variables in modules

Below, most PUBLIC and PRIVATE variables in modules are described briefly (full documentation can also be found in the source code). The file name of the module is given at the right margin of the start of each list. The second column of each list identifies the type of the variable. I, R, L and C represent integer, real, logical and character, A identifies an array, and P

identifies a PARAMETER declaration. All variables are public, unless marked with *. Before describing the variables bundled with subroutines, the three modules with variables only will be discussed.

Physical and mathematical constants : constants.ftn

GRAV	RP	Acceleration of gravity g.	(m s^{-2})
DWAT	RP	Density of water.	(kg m^{-3})
DAIR	RP	Density of air.	(kg m^{-3})
PI	RP	π.	
TPI	RP	2π.	
HPI	RP	0.5π.	
TPIINV	RP	$(2\pi)^{-1}$.	
HPIINV	RP	$(0.5\pi)^{-1}$.	
RADE	RP	Conversion factor from radians to degrees.	
DERA	RP	Conversion factor from degrees to radians.	
RADIUS	RP	Radius of the earth.	(m)
G2PI3I	RP	$g^{-2}(2\pi)^{-3}$.	
G1PI1I	RP	$g^{-1}(2\pi)^{-1}$.	

Unit/processor number information (all initialized) : w3testmd.ftn

NDSO	I	Unit number for file log.ww3.
NDSE	I	Unit number for error messages.
NDST	I	Unit number for file test.ww3.
SCREEN	I	Unit number for 'screen' output. Time step identification is send here by the generic shell if SCREEN $\neq$ NDSO.
NAPROC	I	Number of processors.
IAPROC	I	Local processor number (starting at 1).
NAPLOG	I	Processor to write log file.
NAPOUT	I	Processor to write standard output.
NAPERR	I	Processor to write error output.
NAPFLD	I	Processor to write gridded output.
NAPPNT	I	Processor to write point output.
NAPRST	I	Processor to write fields of mean wave parameters in restart file.
NAPBPT	I	Processor to write general info in boundary data file.
NAPPAR	I	Processor with parameter list output in W3WAVE.

Variables describing the discrete spectrum : w3parmmd.ftn

NTH	I	Number of discrete spectral directions, NTH ≥ 4.
NK	I	Number of discrete spectral wavenumbers, NK ≥ 3.
NK2	I	Extended wavenumber range.
NSPEC	I	Number of spectral bins.
MAPWN	IA	Map with discrete wavenumber for onedimensional description of spectrum.
MAPTH	IA	Id. for discrete directions.
DTH	R	Spectral directional increment. (rad)
TH	RA	Spectral directions. (rad)
ESIN	RA	$\sin(\theta)$ for discrete spectral directions.
ECOS	RA	Id. $\cos(\theta)$.
ES2	RA	$\sin^2(\theta)$ for entire spectrum.
ESC	RA	Id. $\sin(\theta)\cos(\theta)$.
EC2	RA	Id. $\cos^2(\theta)$.
XFR	R	Factor defining frequency grid [X_σ in Eq. (3.1)].
FR1	R	Lowest discrete frequency. (Hz)
SIG	RA	Frequencies for discrete wavenumbers. (rad s^{-1})
SIG2	RA	Id. entire discrete spectrum.
DSIP	RA	Frequency band widths for each wavenumber as used in propagation. (rad s^{-1})
DSII	RA	Id. for spectral integration.
DDEN	RA	Composite band with and conversion to energy for each wavenumber (DDEN = DTH * DSII * SIG). (rad s^{-1})
DDEN2	RA	Id. for entire spectrum.
FTE	R	Factor in tail integration of total energy.
FTF	R	Id. mean frequency.
FTWN	R	Id. mean wavenumber.
FTTR	R	Id. mean period.
FTWL	R	Id. mean wave length.
FACTI1	R	Auxiliary to calculate discrete frequency from continuous frequency.
FACTI2	R	Id.
FACHFA	R	Factor defining parametric tail for the action spectrum $N(k,\theta)$.
FACHFE	R	Id. for the energy spectrum $F(f)$.

Other module variables that are used throughout the code independent of the model options can be found in the following modules. Note that internal data arrays of the model are defined in W3WAVE in w3wavemd.ftn (see corresponding documentation).

Main wave model module : w3wave.ftn

WWVER	CP*	Version number of the main program.
MPIBIF	IP*	Buffer depth for data transpose in distributed memory version. MPIBUF ≥ 1.

General source term integration module : w3srcemd.ftn

OFFSET	RP*	Offset ϵ in source term integration scheme (3.56)
FACP	R	Composite constant in Eq. (3.62).
XR	R	X_r in Eq. (3.63).
XFILT	R	X_f in Eq. (3.64).
FXFM	R	First constant in Eq. (2.39).
FXPM	R	Second constant in Eq. (2.39).
XFT	R	Constant for f_2 in Eq. (2.71).
XFC	R	Constant for f_{hf} in Eq. (2.71).
XSEED	R	X_{seed} in Eq. (3.65).

I/O module (mod_def.ww3) : w3iogrmd.ftn

VERGRD	CP*	Version number of file mod_def.ww3.	
IDSTR	CP*	ID string for file.	
GNAME	C	Grid name.	
NX,NY	I	discrete grid dimensions, NX, NY ≥ 3.	
NSEA	I	Number of sea points in grid.	
NSEAL	I	Id. locally stored on present process.	
FLAGLL	LP	Flag for spherical grid (otherwise Cartesian).	
SX,SY	R	Spatial grid increments.	(°)
X0,Y0	R	Lower left point of spatial grid.	(°)
ZB	RA	Bottom depths on storage grid.	(m)
MAPSTA	IA	Grid status map.	
MAPSF	IA	Storage grid map.	
MAPFS	IA	Is.	
TRNX/Y	RA	grid box transparencies.	(–)

DTCFL	R	Maximum CFL number spatial propagation.	
DTCFLI	R	Id. intra-spectral propagation.	
DTMAX	R	Maximum overall time step.	
DTMIN	R	Minimum source term integration time step.	
DMIN	R	Minimum water depth.	(m)
CFLTM	R	Maximum CFL number depth refraction.	
CICE0/N	R	Ice concentration cut-off.	(-)
GLOBAL	L	Flag for global grid.	
FLDRY	L	Flag for 'dry run' (no calculations).	
FLC*xx*	L	Flags for propagation in all spaces.	
FLSOU	L	Flag for source term integration.	
CLAT(I)	RA	(Inverse) cosine of latitude.	
CLATS	RA	Id.	
CTHG0	RA	Constant in great circle propagation speed.	

I/O module (out_grd.ww3) : w3iogomd.ftn

VEROGR	CP*	Version number of file out_grd.ww3.	
IDSTR	CP*	ID string for file.	
NOGRID	IP	Number of field output options, do not change.	
IDOUT	CA	ID strings for output fields.	
NRQGO	I	Number of MPI handles for persistent communication to collect data for W3IOGO.	(!/MPI)
IRQGO	IA	Corresponding handles.	(!/MPI)
NRQPA	I	Id. parameter list output in W3WAVE.	(!/MPI)
IRQPA	IA	Id. parameter list output in W3WAVE.	(!/MPI)

I/O module (out_pnt.ww3) : w3iopomd.ftn

VEROPT	CP*	Version number of file out_pnt.ww3.	
IDSTR	CP*	ID string for file.	
NOPTS	I	Number of output points.	
PTLOC	RA	Location of output points.	
PTNME	CA	Names of output points.	
IPTINT	IA	Interpoltaion data for output points.	
PTIFAC	RA	Id.	
IL	IA	Number of land points in interpolation for each output point.	
IW	IA	Id. sea points.	

II	IA	Id. ice covered points.
DPO	RA	Interpolated output depths.
WAO	RA	Interpolated output wind speeds.
WDO	RA	Interpolated output wind directions.
ASO	RA	Interpolated output air-sea temperature differences.
CAO	RA	Interpolated output current speeds.
CDO	RA	Interpolated output current directions.
SPCO	RA	Interpolated output spectra.
NRQPO	I	Number of MPI handles for persistent communication to collect data for W3IOPO. (!/MPI)
NRQPO2	I	Id. (!/MPI)
IRQPOn	IA	Corresponding handles. (!/MPI)

I/O module (track_o.ww3) : w3iotrmd.ftn

VERTRK	CP*	Version number of file track_o.ww3.
IDSTRI	CP*	ID string for file track_i.ww3.
IDSTRO	CP*	ID string for file track_o.ww3.

I/O module (restart.ww3) : w3iorsmd.ftn

VERINI	CP*	Version number of file restart.ww3.
IDSTR	CP*	ID string for file.
NRQRS	I	Number of MPI handles for persistent communication to collect data for W3IORS. (!/MPI)
IRQRS	IA	Corresponding handles. (!/MPI)

I/O module (nest.ww3) : w3iobcmd.ftn

VERBPT	CP*	Version number of file nest.ww3.
IDSTR	CP*	ID string for file.
FLBPI	L	Flag for input boundary data.
FLBPO	L	Flag for output boundary data.
NBI(2)	I	Number of input boundary points/spectra.
NFBPO	I	Number of output boundary files.
NBO(2)	IA	Number of output boundary points/spectra.
IPBPI	IA	Interpolation data for input boundary points.
ISBPI	IA	Sea point counters for input boundary points.
X/YBPI	RA	Locations of input boundary points.
RDBPI	RA	Interpolation factors for input boundary points.

IPBPO, ISBPO, X/YBPO, RDBPO		
	...	Id. output boundary points.
ABPI0/N	RA	Storage for input boundary spectra.
ABPOS	RA	Storage for output boundary spectra.
NRQBP	I	Number of MPI handles for persistent communication to collect data for W3IOBC. (!/MPI)
NRQBP2	I	Id. (!/MPI)
IRQBPn	IA	Corresponding handles. (!/MPI)

Updating input fields : w3updtmd.ftn

ATRNX/Y	RA	Actual transparencies used.
CA0/I	RA*	Initial current velocity and increment.
CD0/I	RA*	Id. direction.
UA0/I	RA*	Id. wind speed.
UD0/I	RA*	Id. direction.
AS0/I	RA*	Id air-sea temperature difference.

Apart from the generally used variables in the modules, the various numerical and physical options carry their own module variables. Due to the bundling of the approaches in modules, many of these variables are private to the module. Because the he private variables are used transparently within the module, they will not be discussed here (see documentation in source code).

Propagation scheme (UQ scheme with diffusion) : w3pro2md.ftn

DTIME	R	Swell age in diffusive dispersion correction.
CLATMN	R	Id. minimum cosine of latitude.
FLSOFT	L	Flag for soft boundary treatment.

Propagation scheme (UQ scheme with averaging) : w3pro3md.ftn

FLSOFT	L	Flag for soft boundary treatment.
WDTHxx	R	Tuning factors for averaging widths in Eq. (3.40).

Propagation scheme (UQ scheme with divergence) : w3pro4md.ftn

FLSOFT	L	Flag for soft boundary treatment.
*xx*FAC	R	Tuning factors in Eqs. (3.41) through (3.43).

Input and dissipation (WAM-3) : w3src1md.ftn

SINC1	R	Proportionality constant in Eq. (2.34).
SDSC1	R	Composite constant in Eq. (2.36).

Input and dissipation (Tolman and Chalikov, 1996) : w3src2md.ftn

NITTIN	I	Number of iteration in calculation of u_*.
ZWND	R	Height of input wind.
CINXSI	R	χ in Eq. (2.48).
SWELLF	R	Swell regative input reduction factor in Eq. (2.56).
STABSH	R	c_0 in Eq. (2.72).
STABOF	R	$\mathcal{ST}_o$ in Eq. (2.72).
CNEG	R	c_1 in Eq. (2.73).
CPOS	R	c_2 in Eq. (2.74).
FNEG	R	f_1 in Eq. (2.73).
FPOS	R	f_2 in Eq. (2.74).
SDSAn	R	Constants a_n in Eq. (2.65).
SDSALN	R	α_r in Eq. (2.66).
SDSBn	R	Constants b_n in Eq. (2.59).
FPIMIN	R	Minimum value of $\tilde{\phi}_{p,i}$ allowed in Eq. (2.59).
XFH	R	Constant for f_h in Eq. (2.71).
XF1	R	Constant for f_1 in Eq. (2.71).
XF2	R	Constant for f_2 in Eq. (2.71).

Nonlinear interactions (DIA) : w3snl1md.ftn

NFR	I	Number of frequencies NFR = NK.
SNLC1	R	Constant C in Eq. (2.20).
LAMBDA	R	Constant λ in Eq. (2.19).
KDCONV	R	Constant in Eq. (2.24).
KDMIN	R	Minimum relative depth allowed in Eq. (2.21).
SNLCSn	R	Constants $c_n$cin Eq. (2.21) .

Nonlinear interactions (WRT) : w3snl2md.ftn

IQTYPE	I	Type of interactions to be calculated (deep, shallow scaled, shallow calculated).
TAILNL	R	Power of parametric tail.
NDEPTH	I	Number of depths for which integration space is set up.
DPTHNL	RA	Id. depths.

Bottom friction (JONSWAP) : w3sbt1md.ftn

SBTC1	R	Composite constant in Eq. (2.76).

The service modules contain private variables only, with the exception of the interpolation tables for the solution of the dispersion relation

Solving the dispersion relation : w3dispmd.ftn

NAR1D	IP	Dimension of interpolation tables.
DFAC	RP	Maximum nondimensional water depth kd.
ECG1	RA	Table for calculating group velocities from the frequency and the depth.
EWN1	RA	Id. wavenumbers.
N1MAX	I	Largest index in tables.
DSIE	R	Nondimensional frequency increment.

Finally, the main programs only contain locally defined variables, which need not be documented here in detail.

This page is intentionally left blank.

References

Booij, N. and L. H. Holthuijsen, 1987: Propagation of ocean waves in discrete spectral wave models. *J. Comput. Physics*, **68**, 307–326.

Booij, N., R. C. Ris and L. H. Holthuijsen, 1999: A third-generation wave model for coastal regions, Part I, model description and validation. *J. Geophys. Res.*, **104**, 7649–7666.

Bouws, E. and G. J. Komen, 1983: On the balance between growth and dissipation in an extreme depth-limited wind-sea in the southern north sea. *J. Phys. Oceanogr.*, **13**, 1653–1658.

Bretherthon, F. P. and C. J. R. Garrett, 1968: Wave trains in inhomogeneous moving media. *Proc. Roy. Soc. London*, **A 302**, 529–554.

Cahyono, 1994: *Three-dimensional numerical modelling of sediment transport processes in non-stratified estuarine and coastal waters.* Ph.D. Thesis, University of Bradford, 315 pp.

Chalikov, D. V., 1995: The parameterization of the wave boundary layer. *J. Phys. Oceanogr.*, **25**, 1333–1349.

Chalikov, D. V. and M. Y. Belevich, 1993: One-dimensional theory of the wave boundary layer. *Bound. Layer Meteor.*, **63**, 65–96.

Charnock, H., 1955: Wind stress on a water surface. *Quart. J. Roy. Meteor. Soc.*, **81**, 639–640.

Christoffersen, J. B., 1982: Current depth refraction of dissipative water waves. Series Paper 30, Institute of Hydrodynamics and Hydraulic Engineering, Techn. Univ. Denmark.

Davis, R. W. and E. F. More, 1982: A numerical study of vortex shedding from rectangles. *J. Fluid Mech.*, **116**, 475–506.

Doty, B., 1995: *The grid analysis and display system GrADS.* COLA, http://www.iges.org/grads.

Falconer, R. A. and Cayhono, 1993: Water quality modelling in well mixed estuaries using higher order accurate differencing schemes. in S. S. Y. Wang, editor, *Advances in Hydro- Science and Engineering*, pp. 81–92. CCHE, University of Mississippi.

Fletcher, C. A. J., 1988: *Computational techniques for fluid dynamics, part I and II.* Springer, 409+484 pp.

Gropp, W., E. Lusk and A. Skjellum, 1997: *Using MPI: Portable parallel programming with the message-passing interface.* MIT Press, 299 pp.

Hargreaves, J. C. and J. D. Annan, 1998: Integration of source terms in WAM. in *Proceedings of the 5th International Workshop on Wave Fore-*

casting and Hindcasting, pp. 128–133.

Hargreaves, J. C. and J. D. Annan, 2001: Comments on improvement of the short fetch behavior in the WAM model. *J. Atmos. Oceanic Techn.*, **18**, 711–715.

Hasselmann, K., 1962: On the non-linear transfer in a gravity wave spectrum, part 1. General theory. *J. Fluid Mech.*, **12**, 481–500.

Hasselmann, K., 1963a: On the non-linear transfer in a gravity wave spectrum, part 2, Conservation theory, wave-particle correspondence, irreversibility. *J. Fluid Mech.*, **15**, 273–281.

Hasselmann, K., 1963b: On the non-linear transfer in a gravity wave spectrum, part 3. Evaluation of energy flux and sea-swell interactions for a Neuman spectrum. *J. Fluid Mech.*, **15**, 385–398.

Hasselmann, K., T. P. Barnett, E. Bouws, H. Carlson, D. E. Cartwright, K. Enke, J. A. Ewing, H. Gienapp, D. E. Hasselmann, P. Kruseman, A. Meerburg, P. Mueller, D. J. Olbers, K. Richter, W. Sell and H. Walden, 1973: Measurements of wind-wave growth and swell decay during the Joint North Sea Wave Project (JONSWAP). *Ergaenzungsheft zur Deutschen Hydrographischen Zeitschrift, Reihe A(8)*, **12**, 95 pp.

Hasselmann, S. and K. Hasselmann, 1985: Computations and parameterizations of the nonlinear energy transfer in a gravity-wave spectrum, Part I: A new method for efficient computations of the exact nonlinear transfer integral. *J. Phys. Oceanogr.*, **15**, 1369–1377.

Hasselmann, S., K. Hasselmann, J. H. Allender and T. P. Barnett, 1985: Computations and parameterizations of the nonlinear energy transfer in a gravity-wave spectrum, Part II: parameterizations of the nonlinear energy transfer for application in wave models. *J. Phys. Oceanogr.*, **15**, 1378–1391.

Hersbach, H. and P. A. E. M. Janssen, 1999: Improvement of the short fetch behavior in the WAM model. *J. Atmos. Oceanic Techn.*, **16**, 884–892.

Hersbach, H. and P. A. E. M. Janssen, 2001: Reply to comments on "improvement of the short fetch behavior in the WAM model". *J. Atmos. Oceanic Techn.*, **18**, 716–721.

Herterich, K. and K. Hasselmann, 1980: A similarity relation for the nonlinear energy transfer in a finite-depth gravity-wave spectrum. *J. Fluid Mech.*, **97**, 215–224.

Holthuijsen, L. H., N. Booij, R. C. Ris, I. G. Haagsma, A. T. M. M. Kieftenburg and E. E. Kriezi, 2001: *SWAN Cycle III version 40.11 user manual.* Delft University of Technology, Department of Civil Engineering, P.O. Box 5048, 2600 GA Delft, The Netherlands, see http://swan.ct.tudelft.nl.

Janssen, P. A. E. M., 1989: Wind-induced stress and the drag of air-flow over sea waves. *J. Phys. Oceanogr.*, **19**, 745–754.

Kahma, K. K. and C. J. Calkoen, 1992: Reconciling discrepancies in the observed growth rates of wind waves. *J. Phys. Oceanogr.*, **22**, 1389–1405.

Kahma, K. K. and C. J. Calkoen, 1994: *Komen et al. (1994)*. Chap. II.8 Growth curve observations, pp. 174–182. Cambridge Univ. Press.

Komen, G. J., L. Cavaleri, M. Donelan, K. Hasselmann, S. Hasselmann and P. E. A. M. Janssen, 1994: *Dynamics and modelling of ocean waves*. Cambridge University Press, 532 pp.

Komen, G. J., S. Hasselmann and K. Hasselmann, 1984: On the existence of a fully developed wind-sea spectrum. *J. Phys. Oceanogr.*, **14**, 1271–1285.

Kuik, A. J., G. Ph. Van Vledder and L. Holthuijsen, 1988: A method for the routine analysis of pitch-and-roll buoy wave data. *J. Phys. Oceanogr.*, **18**, 1020–1034.

Leonard, B. P., 1979: A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Comput. Methods Appl. Mech. Engng.*, **18**, 59–98.

Leonard, B. P., 1991: The ULTIMATE conservative difference scheme applied to unsteady one-dimensional advection. *Comput. Methods Appl. Mech. Engng.*, **88**, 17–74.

Longuet-Higgins, M. S. and R. W. Stewart, 1961: The changes in amplitude of short gravity waves on steady non-uniform currents. *J. Fluid Mech.*, **10**, 529–549.

Longuet-Higgins, M. S. and R. W. Stewart, 1962: Radiation stress and mass transport in gravity waves, with application to 'surf-beats'. *J. Fluid Mech.*, **10**, 529–549.

Mei, C. C., 1983: *The applied dynamics of ocean surface waves*. Wiley, New York, 740 pp.

NCEP, 1998: GRIB. Office Note 388, NOAA/NWS/NCEP, Available by anonymous ftp from ftp://nic.fb4.noaa.gov.

Phillips, O. M., 1977: *The dynamics of the upper ocean, second edition*. Cambridge Univ. Press, 336 pp.

Pierson, W. J. and L. Moskowitz, 1964: A proposed spectral form for fully developed wind seas based on the similarity theory of S. A. Kitaigorodskii. *J. Geophys. Res.*, **69**, 5181–5190.

Resio, D. T. and W. Perrie, 1991: A numerical study of nonlinear energy fluxes due to wave-wave interactions. Part 1: Methodology and basic results. *J. Fluid Mech.*, **223**, 609–629.

Shemdin, O., K. Hasselmann, S. V. Hsiao and K. Heterich, 1978: Nonlinear and linear bottom interaction effects in shallow water. in *Turbulent fluxes through the sea surface, wave dynamics and prediction*, pp. 347–365. NATO Conf. Ser. V, Vol 1.

Snyder, R. L., F. W. Dobson, J. A. Elliott and R. B. Long, 1981: Array measurements of atmospheric pressure fluctuations above surface gravity waves. *J. Fluid Mech.*, **102**, 1–59.

Tolman, H. L., 1990: The influence of unsteady depths and currents of tides on wind wave propagation in shelf seas. *J. Phys. Oceanogr.*, **20**, 1166–1174.

Tolman, H. L., 1991: A third-generation model for wind waves on slowly varying, unsteady and inhomogeneous depths and currents. *J. Phys. Oceanogr.*, **21**, 782–797.

Tolman, H. L., 1992: Effects of numerics on the physics in a third-generation wind-wave model. *J. Phys. Oceanogr.*, **22**, 1095–1111.

Tolman, H. L., 1995: On the selection of propagation schemes for a spectral wind wave model. Office Note 411, NWS/NCEP, 30 pp + figures.

Tolman, H. L., 1999a: User manual and system documentation of WAVEWATCH III version 1.18. Tech. Note 166, NOAA/NWS/NCEP/OMB, 110 pp.

Tolman, H. L., 1999b: WAVEWATCH III version 1.18 : Generating GRIB files. Tech. Note 167, NOAA/NWS/NCEP/OMB, 7 pp.

Tolman, H. L., 1999c: WAVEWATCH III version 1.18 : Post-processing unsing NCAR graphics. Tech. Note 168, NOAA/NWS/NCEP/OMB, 9 pp.

Tolman, H. L., 2001: Improving propagation in ocean wave models. in B. L. Edge and J. M. Hemsley, editors, *Ocean Wave Measurement and Analysis*, pp. 507–516. ASCE.

Tolman, H. L., 2002a: Alleviating the garden sprinkler effect in wind wave models. *Ocean Mod.*, **4**, 269–289.

Tolman, H. L., 2002b: Distributed memory concepts in the wave model WAVEWATCH III. *Parallel Computing*, **28**, 35–52.

Tolman, H. L., 2002c: Limiters in third-generation wind wave models. *The Global Atmosphere and Ocean System*, **8**, 67–83.

Tolman, H. L., 2002d: Testing of WAVEWATCH III version 2.22 in NCEP's NWW3 ocean wave model suite. Tech. Note 214, NOAA/NWS/NCEP/OMB, 99 pp.

Tolman, H. L., 2002e: Treatment of unresolved islands and ice in wind wave models. *Ocean Mod.*, Accepted.

Tolman, H. L., 2002f: Validation of WAVEWATCH III version 1.15 for a global domain. Tech. Note 213, NOAA/NWS/NCEP/OMB, 33 pp.

Tolman, H. L., B. Balasubramaniyan, L. D. Burroughs, D. V. Chalikov, Y. Y. Chao, H. S. Chen and V. M. Gerald, 2002: Development and implementation of wind generated ocean surface wave models at NCEP. *Wea. Forecasting*, **17**, 311–333.

Tolman, H. L. and N. Booij, 1998: Modeling wind waves using wavenumber-direction spectra and a variable wavenumber grid. *The Global Atmosphere and Ocean System*, **6**, 295–309.

Tolman, H. L. and D. V. Chalikov, 1996: Source terms in a third-generation wind-wave model. *J. Phys. Oceanogr.*, **26**, 2497–2518.

Tracy, B. and D. T. Resio, 1982: Theory and calculation of the nonlinear energy transfer between sea waves in deep water. WES Report 11, US Army Corps of Engineers.

Van Vledder, G. Ph., 2000: Improved method for obtaining the integration space for the computation of nonlinear quadruplet wave-wave interaction. in *Proceedings of the 6th International Workshop on Wave Forecasting and Hindcasting*, pp. 418–431.

Van Vledder, G. Ph., 2002a: Improved parameterizations of nonlinear four wave interactions for application in operational wave prediction models. Report 151a, Alkyon, The Netherlands.

Van Vledder, G. Ph., 2002b: A subroutine version of the Webb/Resio/Tracy method for the computation of nonlinear quadruplet interactions in a wind-wave spectrum. Report 151b, Alkyon, The Netherlands.

WAMDIG, 1988: The WAM model – a third generation ocean wave prediction model. *J. Phys. Oceanogr.*, **18**, 1775–1809.

Webb, D. J., 1978: Non-linear transfers between sea waves. *Deep-Sea Res.*, **25**, 279–298.

Whitham, G. B., 1965: A general approach to linear and non-linear dispersive waves using a Lagrangian. *J. Fluid Mech.*, **22**, 273–283.

Wu, J., 1982: Wind-stress coefficients over sea surface from breeze to hurricane. *J. Geophys. Res.*, **87**, 9704–9706.