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 1.18 †

Hendrik L. Tolman ‡
UCAR Visiting Scientist
Ocean Modeling Branch

April 1999

---

† OMB Contribution No. 166.

‡ e-mail: Hendrik.Tolman@NOAA.gov

Contents

1 Introduction

1.1 About this manual

This is the user manual and system documentation of version 1.18 the full-spectral third-generation ocean wind-wave model WAVEWATCH III (henceforth denoted as WWATCH). WWATCH has been developed at the Ocean Modeling Branch (OMB) 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, because users are expected to upgrade the model according to their own specifications. This will require a detailed knowledge of the source code. 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 previous documented version of WWATCH is version 1.15. This model version has mostly been used as the beta version of the model at NCEP. It has been used for the final testing and tuning of the model, and has provided the "experimental wave model" data for the OMB home page (http://polar.wwb.noaa.gov) from early 1997 to the present. The present version of WWATCH (version 1.18), is identical to the latter version with respect to the physical and numerical approaches available. The main differ-

ence is that the new version has been adapted for use in a distributed memory environment. This requires a data distribution strategy and message passing as described in section 6.5.2. To promote portability of the code, the Message Passing Interface (MPI) is used for message passing. The data distribution and message passing required some major modifications to the code. The strategy chosen, however, makes future modifications to both physical and numerical approaches completely transparent.

The new version of the model includes some changes in the model output. Restart and track output files now are unformatted direct access files. In the point output data files, the longitude and latitude of the output points have been added, as well a the name of the originating grid. The directions in the spectral data files have been modified to reflect the documentation. A new post-processor has been written to convert unformatted direct access track output files to the previous integer packed format. Finally, a new spectral bulletin generator has been provided, which runs off post-processed spectral data files.

Another significant change to the model is that an initial conditions or restart file is no longer necessary. If no restart file is found, the model will default to initializing with the initial wind field.

A final note of caution. 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.

For comments, questions and suggestions, please send E-mail to :

wavewatch@ncep.noaa.gov

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 Ocean Modeling Branch (OMB) 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.

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 p. 96; 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 Jacobean transformations

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

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

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

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 Jacobean transformation as described above is well behaved. 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 plane 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 WAM, WAMDIG 1988)

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

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

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

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

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.

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 calculation of S_{nl} will be discussed first in section 2.3.2. 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 2.3.3) and the parameterizations of Tolman and Chalikov (1996) (section 2.3.4). 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 2.3.5) is available.

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 Jacobean 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 are modeled using the discrete interaction approximation (DIA) of Hasselmann et al. (1985). This parameterization

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

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

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

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

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

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

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. Recommended 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)

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

Recommended 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 are set by the user in the input files of the model (see chapter 4).

2.3.3 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

$$\mathcal{S}_{in}(k,\theta) = C_{in}\frac{\rho_a}{\rho_w} \max\left[0, \left(\frac{28\,u_*}{c}\cos(\theta-\theta_w) - 1\right)\right]\; \sigma\, N(k,\theta) \,, \tag{2.26}$$

$$u_* = u_{10}\sqrt{(0.8 + 0.065 u_{10}) 10^{-3}} \,, \tag{2.27}$$

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

$$\mathcal{S}_{ds}(k,\theta) = C_{ds}\,\hat{\sigma}\,\frac{k}{\hat{k}}\left(\frac{\hat{\alpha}}{\hat{\alpha}_{PM}}\right)^2 N(k,\theta) \,, \tag{2.28}$$

$$\hat{\sigma} = \left(\overline{\sigma^{-1}}\right)^{-1} \,, \tag{2.29}$$

$$\hat{\alpha} = E\,\hat{k}^2\,g^{-2} \,, \tag{2.30}$$

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[1] $m = 4.5$ and by

$$f_{hf} = \max\left[\, 2.5\,\hat{f}_r \,,\, 4\,f_{r,PM} \,\right] \,, \tag{2.31}$$

$$f_{r,PM} = \frac{g}{28\,u_*} \,, \tag{2.32}$$

where $f_{r,PM}$ is the Pierson-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 defined by the user in the input files of the model.

2.3.4 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

$$\mathcal{S}_{in}(k,\theta) = \sigma\,\beta\,N(k,\theta) \,, \tag{2.33}$$

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

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

$$10^4\beta = \begin{cases} -a_1\tilde{\sigma}_a^2 - a_2 & , & & \tilde{\sigma}_a & \leq -1 \\ a_3\tilde{\sigma}_a(a_4\tilde{\sigma}_a - a_5) - a_6 & , & -1 < & \tilde{\sigma}_a & < \Omega_1/2 \\ (a_4\tilde{\sigma}_a - a_5)\tilde{\sigma}_a & , & \Omega_1/2 < & \tilde{\sigma}_a & < \Omega_1 \\ a_7\tilde{\sigma}_a - a_8 & , & \Omega_1 < & \tilde{\sigma}_a & < \Omega_2 \\ a_9(\tilde{\sigma}_a - 1)^2 + a_{10} & , & \Omega_2 < & \tilde{\sigma}_a & \end{cases} \tag{2.34}$$

where

$$\tilde{\sigma}_a = \frac{\omega\ u_\lambda}{g}\cos(\theta - \theta_w) \tag{2.35}$$

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

$$\lambda_a = \frac{2\pi}{k|\cos(\theta - \theta_w)|}\ . \tag{2.36}$$

The parameters $a_1 - a_{10}$ and Ω_1, Ω_2 in Eq. (2.34) depend on the drag coefficient C_λ at the height $z = \lambda_a$:

$$\begin{array}{llll} \Omega_1 = 1.075 + 75C_\lambda & \Omega_2 = & 1.2 + 300C_\lambda \\ a_1 = 0.25 + 395C_\lambda, & a_3 = & (a_0 - a_2 - a_1)/(a_0 - a_4 + a_5) \\ a_2 = 0.35 + 150C_\lambda, & a_5 = & a_4\Omega_1 \\ a_4 = 0.30 + 300C_\lambda, & a_6 = & a_0(1 - a_3) \\ a_9 = 0.35 + 240C_\lambda, & a_7 = & (a_9(\Omega_2 - 1)^2 + a_{10})/(\Omega_2 - \Omega_1) \\ a_{10} = -0.05 + 470C_\lambda, & a_8 = & a_7\Omega_1 \\ & a_0 = & 0.25a_5^2/a_4 \end{array} \tag{2.37}$$

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

$$u_z = \frac{v_*}{\kappa}\ln\left(\frac{z}{z_0}\right)\ , \tag{2.38}$$

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)

$$C_r = \kappa^2 \left[R - \ln(C) \right]^2 \ , \tag{2.39}$$

where

$$R = \ln \left(\frac{z_r g}{\chi \sqrt{\alpha} u_r^2} \right) \ , \tag{2.40}$$

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

$$C_r = 10^{-3} \left(0.021 + \frac{10.4}{R^{1.23} + 1.85} \right) \ . \tag{2.41}$$

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)

$$\alpha = 0.57 \left(\frac{u_*}{c_p} \right)^{3/2} \ . \tag{2.42}$$

As the latter equation depends on the drag coefficient, Eqs. (2.40) through (2.42) 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. (2.42) 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. (2.38) the roughness parameter z_0 becomes

$$z_0 = z_r \ \exp \left(-\kappa C_r^{-1/2} \right) \ , \tag{2.43}$$

and the wind velocity and drag coefficient at height λ become

$$u_\lambda = u_r \frac{ln(\lambda_a/z_0)}{ln(z_r/z_0)} \tag{2.44}$$

$$C_\lambda = C_r \left(\frac{u_a}{u_\lambda}\right)^2 \tag{2.45}$$

Finally, Eq. (2.42) 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)

$$f_{p,i} = \frac{\int\int f^{-2}\, c_g^{-1}\, \max\left[\, 0\,,\, \mathcal{S}_{wind}(k,\theta)\,\right] df\, d\theta}{\int\int f^{-3}\, c_g^{-1}\, \max\left[\, 0\,,\, \mathcal{S}_{wind}(k,\theta)\,\right] df\, d\theta}\,, \tag{2.46}$$

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

$$\tilde{f}_p \;=\; 3.6\,10^{-4} \;+\; 0.92\tilde{f}_{p,i} \;-\; 6.3\,10^{-10}\,\tilde{f}_{p,i}^{-3}\;. \tag{2.47}$$

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 1999), 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

$$\mathcal{S}_{i,m} = \left\{ \begin{array}{cccccc} \mathcal{S}_i & \text{for} & \beta \geq 0 & \text{or} & f > 0.8 f_p \\ X_s \mathcal{S}_i & \text{for} & \beta < 0 & \text{and} & f < 0.6 f_p \\ \mathcal{X}_s \mathcal{S}_i & \text{for} & \beta < 0 & \text{and} & 06 f_p < f < 0.8 f_p \end{array} \right. , \tag{2.48}$$

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 (2.33), 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,

$$\mathcal{S}_{ds,l}(k,\theta) = -2\, u_*\, h\, k^2 \phi\, N(k,\theta)\,, \tag{2.49}$$

$$h = 4 \left(\int_0^{2\pi} \int_{f_h}^{\infty} F(f,\theta)\, df\, d\theta \right)^{1/2} . \tag{2.50}$$

$$\phi = b_0 + b_1 \tilde{f}_{p,i} + b_2 \tilde{f}_{p,i}^{-b_3}\,. \tag{2.51}$$

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. (2.51) 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

$$b_2 = \tilde{f}_{p,i,\min}^{b_3} \left(\phi_{\min} - b_0 - b_1 \tilde{f}_{p,i,\min} \right)\,, \tag{2.52}$$

$$b_3 = 8\,. \tag{2.53}$$

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

$$\tilde{f}_a = \frac{\phi_{\min} - b_0}{b_1} \quad , \quad \tilde{f}_b = \max \left\{ \tilde{f}_a - 0.0025\,,\, \tilde{f}_{p,i,\min} \right\} \quad , \tag{2.54}$$

$$b_2 = \tilde{f}_b^{b_3} \left[\phi_{\min} - b_0 - b_1 \tilde{f}_b \right]\,, \tag{2.55}$$

$$b_3 = \frac{b_1 \tilde{f}_b}{\phi_{\min} - b_0 - b_1 \tilde{f}_b}\,. \tag{2.56}$$

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

$$\mathcal{S}_{ds,h}(k,\theta) = -a_0 \left(\frac{u_*}{g} \right)^2 f^3\, \alpha_n^B\, N(k,\theta) \quad , \tag{2.57}$$

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

$$\alpha_n = \frac{\sigma^6}{c_g \, g^2 \alpha_r} \int_0^{2\pi} N(k,\theta) \, d\theta \, , \tag{2.58}$$

where α_n is Phillips' nondimensional high-frequency energy level normalized with α_r, and where a_0 through a_3 and α_r are empirical constants. This parameterization implies that $m = 5$ in the parametric tail, which has been preset in the model.

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

$$\mathcal{S}_{ds}(k,\theta) = \mathcal{A} \mathcal{S}_{ds,l} + (1 - \mathcal{A}) \, \mathcal{S}_{ds,h} \, , \tag{2.59}$$

$$\mathcal{A} = \begin{cases} 1 & \text{for} \quad f < f_l \quad , \\ \frac{f - f_2}{f_1 - f_2} & \text{for} \quad f_1 \leq f < f_2 \quad , \\ 0 & \text{for} \quad f_2 \leq f \quad , \end{cases} \tag{2.60}$$

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)

$$N(k_i,\theta) = (1 - \mathcal{B}) \, N(k_i,\theta) + \mathcal{B} N(k_{i-1},\theta) \left(\frac{f_i}{f_{i-1}} \right)^{-m-2} \quad , \tag{2.61}$$

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

$$\left. \begin{array}{rcl} f_{hf} & = & 3.00 \, f_{p,i} \\ f_1 & = & 1.75 \, f_{p,i} \\ f_2 & = & 2.50 \, f_{p,i} \\ f_h & = & 2.00 \, f_{p,i} \end{array} \right\} \quad . \tag{2.62}$$

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

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).

Test results of these source terms in a global model implementation (Tolman 1998) 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 (1998) the following effective wind speed is used :

$$\frac{u_e}{u} = u \left(\frac{c_o}{1 + C_1 + C_2} \right)^{-1/2} , \tag{2.63}$$

$$C_1 = c_1 \tanh \left[\max(0, f_1 \{ \mathcal{ST} - \mathcal{ST}_o \}) \right] , \tag{2.64}$$

$$C_2 = c_2 \tanh \left[\max(0, f_2 \{ \mathcal{ST} - \mathcal{ST}_o \}) \right] , \tag{2.65}$$

$$\mathcal{ST} = \frac{hg}{u_h^2} \frac{T_a - T_s}{T_0} , \tag{2.66}$$

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 $f_2 = f_1 c_1 / c_2$. In Tolman (1998) $c_0 = 1.4$, $c_1 = -0.1$, $c_2 = 0.1$, $f_1 = -150$ 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)i was used. Note that this effective wind speed was derived for winds at 10 m height. The wind correction can be switched on or off by the user during compilation of the model, and parameter values are defined by the user in the program input files.

2.3.5 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

$$\mathcal{S}_{bot}(k,\theta) = 2\Gamma \, \frac{n-0.5}{gd} \, N(k,\theta) \, , \tag{2.67}$$

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 actual value for Γ is provided 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 4.3.5

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)]

$$H_s = 4\sqrt{E} \, . \tag{2.68}$$

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

$$L_m = 2\pi \overline{k^{-1}} \, . \tag{2.69}$$

8) Mean wave period (s)

$$T_m = 2\pi\overline{\sigma^{-1}} \,. \tag{2.70}$$

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

$$\theta_m = \mathrm{atan}\left(\frac{b}{a}\right) \,, \tag{2.71}$$

$$a = \int_0^{2\pi}\int_0^{\infty} \cos(\theta) F(\sigma,\theta)\, d\sigma\, d\theta \,, \tag{2.72}$$

$$b = \int_0^{2\pi}\int_0^{\infty} \sin(\theta) F(\sigma,\theta)\, d\sigma\, d\theta \,. \tag{2.73}$$

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

$$\sigma_\theta = \left[2\left\{1-\left(\frac{a^2+b^2}{E^2}\right)^{1/2}\right\}\right]^{1/2} \,, \tag{2.74}$$

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 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. (2.46) and (2.47).
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)

2.5 References

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.

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

——, and M. Yu. 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. Institute of Hydrodynamics and Hydraulic Engineering, Techn. Univ. Denmark, Series Paper No. 30.

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. Müller, 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). *Ergänzungsheft zur Deutschen Hydrographischen Zeitschrift*, Reihe A (8) Nr. 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.

——, ——, 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.

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.

——, and ——, 1994: Growth curve observations. In: Komen et al., 1994, 174-182.

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.

———, 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.

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

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.

——— , and ——— , 1962: Radiation stress and mass transport in gravity waves, with application to 'surf-beats'. *J. Fluid Mech.*, **10**, 529-549.

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

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

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. NATO Conf. Ser. V, Vol 1, 347-365.

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.

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

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

———, 1999: Validation of a global implementation of of NCEP's new ocean wave model. Submitted.

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

WAMDIG, 1988: The WAM model - a third generation ocean wave prediction model. *J. Phys. Oceanogr.*, **18**, 1775-1810.

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

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

3 Numerical approaches

3.1 Basic concepts

Equation (2.12) represents the basic equation of WWATCH. However, a modified version of this equation is used in the model, where (a) Eq. (2.12) is solved on a variable wavenumber grid (see below), and where (b) a discrete version of this equation including a diffusive dispersion correction is used for higher order propagation schemes (see section 3.3).

If Eq. (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.

For notational convenience, the effects of a spatially varying grid will be discussed for the 'plane-grid' equation (2.8). The longitude-latitude grid will be re-introduced in the following sections. 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 l, the maximum time step $\Delta t_{p,l}$ is calculated within the model as

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

   If the propagation time step is smaller than the global time step, the propagation effects are calculated with a number of successive smaller time step. 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). Transferred to a longitude-latitude grid [Eq. (2.12)] the 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 $\mathcal{N} \equiv N\, c_g^{-1} \cos\phi$. Note that (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. Two numerical schemes are available. Selection of the schemes is performed during the compilation of the model.

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

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

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

$$\mathcal{N}_u = \begin{cases} \mathcal{N}_{i-1} & \text{for} \quad \dot{\phi}_b \geq 0 \\ \mathcal{N}_i & \text{for} \quad \dot{\phi}_b < 0 \end{cases} \ , \tag{3.8}$$

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 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 Cahyono (Cahyono 1993; Falconer and Cahyono 1993; see 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[2]

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

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

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

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

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

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

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

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

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, 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 propagated 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

The ULTIMATE QUICKEST scheme is sufficiently free of numerical diffusion for the so-called 'garden sprinkler' effect 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). Booij and Holthuijsen have 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

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

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

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

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

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

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

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 is added as a separate step

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

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). Secondly, the diffusion coefficients D_{ss} and D_{nn} are calculated assuming deep water

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

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

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 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.

3.4 Intra-spectral propagation

The third step of the numerical algorithm considers refraction and residual (current-induced) wavenumber shifts. Again taking into account the transformation to a (ϕ, λ) grid, the equation to be solved in this step becomes

$$\frac{\partial N}{\partial t} + \frac{\partial}{\partial k} \dot{k}_g N + \frac{\partial}{\partial \theta} \dot{\theta}_g N = 0 \,, \tag{3.40}$$

$$\dot{k}_g = \frac{\partial \sigma}{\partial d} \frac{\mathbf{U} \cdot \nabla_x d}{c_g} - \mathbf{k} \cdot \frac{\partial \mathbf{U}}{\partial s} \,. \tag{3.41}$$

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

$$\dot{\theta} = X_{rd}(\lambda,\phi,k)\dot{\theta}_d + \dot{\theta}_c \;, \tag{3.42}$$

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. 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

$$N^{n+1}_{i,j,l,m} = N^{n}_{i,j,l,m} + \frac{\Delta t}{\Delta\theta}\left[\mathcal{F}_{l,-} - \mathcal{F}_{l,+}\right] + \frac{\Delta t}{\Delta k_m}\left[\mathcal{F}_{m,-} - \mathcal{F}_{m,+}\right] \;, \tag{3.43}$$

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

$$\mathcal{F}_{m,-} = \left[\dot{k}_{g,b}\, N_b \right]^n_{i,j,l} \;, \tag{3.44}$$

$$\dot{k}_{g,b} = 0.5 \left(\dot{k}_{g,m-1} + \dot{k}_{g,m} \right) \;, \tag{3.45}$$

$$N_b = \frac{1}{2} \left[(1+C) N_{i-1} + (1-C) N_i \right] - \frac{1-C^2}{6}\, \mathcal{CU}\, \Delta k^2_{m-1/2}, \tag{3.46}$$

$$\mathcal{CU} = \begin{cases} \frac{1}{\Delta k_{m-1}} \left[\frac{N_m - N_{m-1}}{\Delta k_{m-1/2}} - \frac{N_{m-1} - N_{m-2}}{\Delta k_{m,-3/2}} \right] & \text{for} \quad \dot{k}_b \geq 0 \\ \frac{1}{\Delta k_m} \left[\frac{N_{m+1} - N_m}{\Delta k_{m+1/2}} - \frac{N_m - N_{m-1}}{\Delta k_{m-1/2}} \right] & \text{for} \quad \dot{k}_b < 0 \end{cases} \;, \tag{3.47}$$

$$C = \frac{\dot{k}_{g,b}\, \Delta t}{\Delta k_{m-1/2}} \;, \tag{3.48}$$

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. (3.48) 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

$$N^{n+1}_{i,j,l,m} = N^n_{i,j,l,m} + \frac{\Delta t}{\Delta k_m} \left[\mathcal{F}_{m,-} - \mathcal{F}_{m,+} \right] \;, \tag{3.49}$$

3.5 Source terms

Finally, the source terms are accounted for by solving

$$\frac{\partial N}{\partial t} = \mathcal{S} \, . \tag{3.50}$$

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

$$\Delta N(k,\theta) = \frac{\mathcal{S}(k,\theta)}{1 - 0.5 D(k,\theta)\Delta t} \, , \tag{3.51}$$

where D represents the diagonal terms of the derivative of $\mathcal{S}$ with respect to N (WAMDIG 1988 Eqs. 4.1 through 4.10). This 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

$$\Delta t_d^n = \min_{f<f_{hf}} \left[\frac{\Delta N_m}{|\mathcal{S}|} \left(1 + 0.5 D \frac{\Delta N_m}{|\mathcal{S}|} \right)^{-1} \right] \, , \tag{3.52}$$

$$\Delta t_d^n = \max \left[\, \Delta t_d^n \, , \, \Delta t_{d,\min} \right] \, , \tag{3.53}$$

$$\Delta t_d^n = \min \left[\, \Delta t_d^n \, , \, \Delta t_g - \sum_{i=1}^{n-1} \Delta t_d^i \, \right] \, , \tag{3.54}$$

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

$$N^n = \max \left[\, 0 \, , \, N^{n-1} + \left(\frac{\mathcal{S} \Delta t_d}{1 - 0.5 D \Delta t_d} \right) \, \right] \, . \tag{3.55}$$

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

$$\Delta N_m(k,\theta) = \min \left[\, \Delta N_p(k,\theta) \, , \, \Delta N_r(k,\theta) \, \right] \, , \tag{3.56}$$

	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	$\approx 0.1\Delta t_g$

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

$$\Delta N_p(k,\theta) = X_p \, \frac{\alpha}{\pi} \, \frac{(2\pi)^4}{g^2} \, \frac{1}{\sigma k^3} \; , \tag{3.57}$$

$$\Delta N_r(k,\theta) = X_r \;\; \max \left[\, N(k,\theta) \; , \; N_f \, \right] \; , \tag{3.58}$$

$$N_f = \max \left[\, \Delta N_p(k_{\max},\theta) \; , \; X_f \; \max_{\forall k,\theta} \{ N(k,\theta) \} \, \right] \; , \tag{3.59}$$

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 (3.57) 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 (3.59) 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 (3.57) in conditions where Eq. (3.53) dictates Δt_d^n.

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 $\sigma_{\rm seed} = \min(\sigma_{\max}, 2\pi f_{hf})$ is required to at least contain a minimum action density

$$N_{\min}(k_{seed},\theta) = 6.25\,10^{-4}\frac{1}{k_{\rm seed}^3\,\sigma_{\rm seed}}\max\left[\,0.\,,\,\cos^2(\theta-\theta_w)\right] \min\left[\,1\,,\,\max\left(\,0\,,\,\frac{|u_{10}|}{X_{\rm seed}g\sigma_{\rm seed}^{-1}}-1\right)\right]\,, \quad (3.60)$$

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 (3.57) 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.

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). Generally, 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 furthermore be corrected to (approximately) conserve the energy instead of the wind velocity. The corresponding correction factor X_u is calculated as

$$X_u = \max\left[\,1.25\,,\,\frac{u_{10,rms}}{u_{10,l}}\right]\,, \quad (3.61)$$

where $u_{10,l}$ is the linearly interpolated velocity and $u_{10,rms}$ is the rms interpolated velocity. Finally, winds can 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 4.3.4). 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. (3.61).

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.

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.

3.9 References

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

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

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

Falconer, R.A., and Cahyono, 1993; Water quality modelling in well mixed estuaries using higher order accurate differencing schemes. Advances in hydroscience and engineering, S.S.Y. Wang Ed., 81-92.

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

Leonard, B.P., 1979: A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Comput. Methods Appl. Mech. Engng*, **19**, 59-98.

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

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

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

———, and N. Booij, 1998: Modeling wind waves using wavenumber-direction spectra and a variable wavenumber grid. *Global Atmosphere and Ocean System.* 295-309.

WAMDIG, 1988: The WAM model - a third generation ocean wave prediction model. *J. Phys. Oceanogr.*, **18**, 1775-1810.

4 Running the wave model

4.1 Program design

The core of WWATCH is not a separate program, but consists of two subroutines; an initialization routine and a routine to perform the actual calculations. These routines 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 COMMON 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. It furthermore performs most of the preprocessing required for the COMMON variables in the wave model, which are also stored in the model definition file. 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 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 mem-

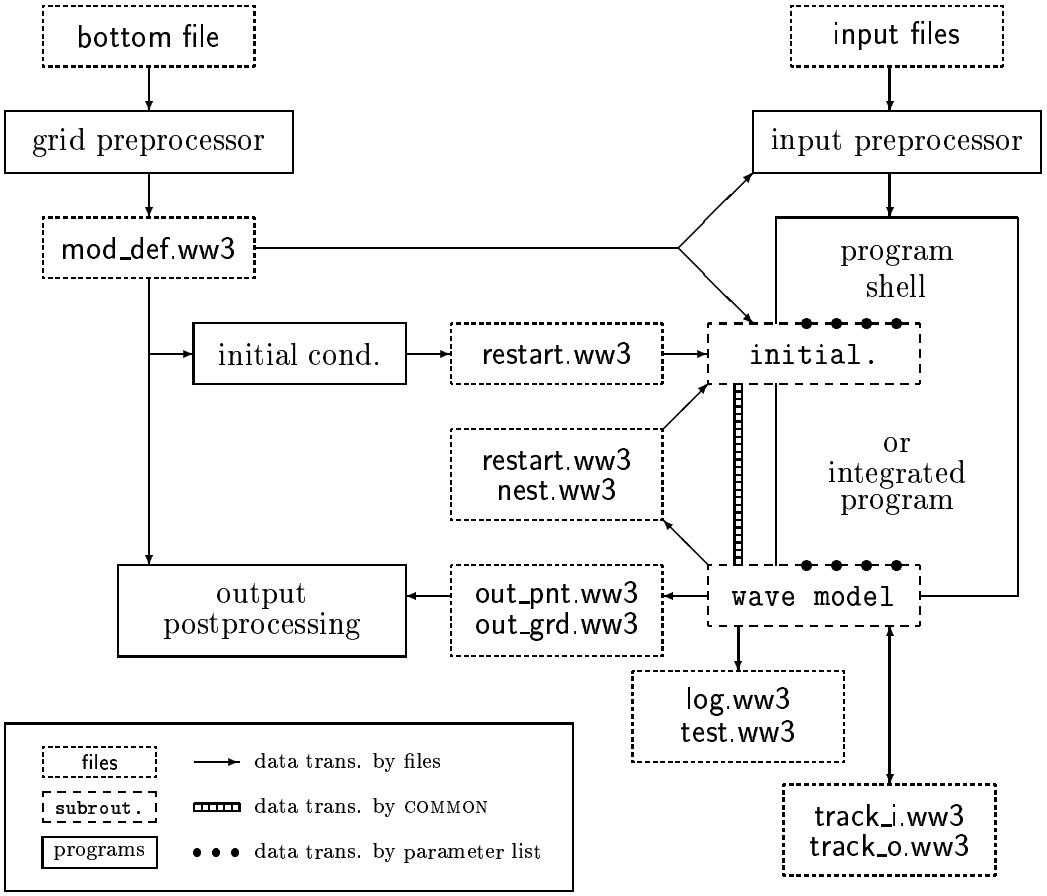

Figure 4.1: Basic program elements and data flow

ory version of the model, or nest*nnn*.ww3 for distributed memory versions, where *nnn* is the processor number starting with 1. Finally, some output post-processors are available. 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.

4.2 Initialization and wave model routines

As discussed above, the actual wave model consists of two subroutines; an initialization routine and a routine to advance the solution in time. To run the model, a program shell is needed. WWATCH is provided with a simple stand-alone shell as will be discussed in section 4.3.5. The present section concentrates on the two basic subroutines.

The initialization subroutine W3INIT performs the initialization of the wave model and has to be called before any calculation can take place. This routine sets up part of the I/O system by defining unit numbers, initializes internal time management, reads and processes the model definition file (mod_def.ww3), reads and processes initial conditions (restart.ww3), prepares model output, and calculates grid-dependent parameters. A documentation of its interface can be found in the source code (w3init.ftn).

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 (w3wave.ftn).

Apart form the raw data files as described above, the program maintains a log file log.ww3. This file is opened by W3INIT, 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 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 six input columns identify boundary conditions (`b`), wind

```
                                        |   input   |    output   |
                                        |-----------|-------------|
   step | pass |    date      time      | b w l c i | 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 |           |   X         |
     40 |   2  |            10:00:00 |           |     X       |
     48 |   2  |            12:00:00 |   X       |   L X       |
     56 |   3  |            14:00:00 |           |     X       |
     64 |   3  |            16:00:00 |           |     X       |
     72 |   3  |            18:00:00 |   X       |     X       |
     80 |   4  |            20:00:00 |           |     X       |
     88 |   4  |            22:00:00 |           |     X       |
     96 |   4  | 1968/06/07 00:00:00 |           |     L   L   |
--------+------+----------------------+-----------+-------------+
```

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

---

fields (w), water levels (l), current fields (c), and ice concentrations (i), respectively. 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 Auxiliary programs

4.3.1 General concepts

All auxiliary programs presented here 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. The programs furthermore write formatted output to the standard output unit.

Below, all available auxiliary programs are described using an example input file with all options included (partially as comment). 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. 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 (w3iogr.ftn).
out_grd.ww3	Subroutine W3IOGO (w3iogo.ftn).
out_pnt.ww3	Subroutine W3IOPO (w3iopo.ftn).
track_o.ww3	Subroutine W3IOTR (w3iotr.ftn).
restart.ww3	Subroutine W3IORS (w3iors.ftn).
nest.ww3	Subroutine W3IOBC (w3iobc.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.

It should be noted that input for specific model options as selected during compilation are read only if the option is activated. Input for options that are not used therefore need to be removed from the input (or converted to comments as in the example files).

4.3.2 The grid preprocessor

Program	:	ww3_grid	(W3GRID)	
Code	:	ww3_grid.ftn		
Input	:	ww3_grid.inp	: Formatted input file for program.	
		'grid file'	: File with bottom depths.	(opt)
Output	:	standard out	: Formatted output of program.	
		mod_def.ww3	: Model definition file in WWATCH format.	
		mask.ww3	: Land-sea mask file (switch O2a).	(opt)

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) ---------------- $
$
   1.1  0.0418
$
$ 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
$
$ Define constants in source terms ---------------------------------- $
$
$ Input  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
$   Not defined         : -
$   WAM-3               : Cin
$
$ 0.25
$
$   Tolman and Chalikov : Height of wind (zr), swell factor in (2.48).
$                       - c0, ST0, c1, c2 and f1 in Eq. (2.63) through
$                         (2.65) for definition of effective wind
$                         speed (only when C/STAB2 is activated).
```

```
$
  10.    0.125
   1.4  -0.01  -0.1  0.1  150.
$
$   Experimental        : -
$
$ Nonlinear interactions - - - - - - - - - - - - - - - - - - - - - - - -
$   Not defined         : -
$   Discrete I.A.       : lambda and C source term, conversion factor
$                         kd in Eq. (2.24),minimum kd, and
$                         constants c1-3 in depth scaling function.
$                           line 1: Cf. WAM model.
$                           line 2: Cf. Tolman and Chalikov (1996).
$
$  0.25     2.78E7     0.75    0.50    5.5    0.833   -1.25
   0.25     1.00E7     0.75    0.50    5.5    0.833   -1.25
$
$   Experimental        : -
$
$ Dissipation  - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
$   Not defined         : -
$   WAM-3               : Cdis, Apm
$
$ -2.36E-5  3.02E-3
$
$   Tolman and Chalikov : constants a0, a1, a2, b0, b1 and PHImin.
$
  4.8  1.7e-4  2.0  0.3e-3  0.47  0.003
$
$   Experimental        : -
$
$ Bottom friction  - - - - - - - - - - - - - - - - - - - - - - - - - - -
$   Not defined         : -
$   JONSWAP             : Gamma
$
  -0.067
$
$   Experimental        : -
$
$ Miscellaneous ---------------------------------------------------- $
$ - Ice concentration cut-off (always needed)
$ - Xseed in seeding algorithm (if C/SEED selected in complation).
$
  0.5
```

```
  2.0
$
$ Set parameters for numerical schemes ------------------------------ $
$ - 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.
$
$ - Additional data for propagation schemes
$   No prop.    : -
$   First order : maximum CFL number for refraction.
$
$   0.7
$
$   UQ scheme   : Id., flag for soft bound., swell age in diffusion (s),
$                 maximum latitude (degr.) in correction of swell age.
$                 Swell age = 0 switches off all diffusion, including
$                 isotropic diffusion for growing components.
$
    0.7  F  1. 70.
$
$ - Xp, Xr and Xf for the dynamic integration scheme (if source terms
$   defined in compilation, not necessarily activated).
$
    0.15  1.00  0.05
$
$ 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.) 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.
$
     12     12
      1.     1.     4.
     -1.    -1.     4.
     -0.1 2.50  10  -10. 3 1 '(....)' 'UNIT' 'input'
$
$ 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
$
$ 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.
$
     1.75  1.50  0.25  0.00     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

4.3.3 The initial conditions program

Program	:	ww3_strt		(W3STRT)
Code	:	ww3_strt.ftn		
Input	:	ww3_strt.inp	:	Formatted input file for program.
		mod_def.ww3	:	Model definition file.
Output	:	standard out	:	Formatted output of program.
		restart.ww3	:	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.) Ym and
$    spread (degr.), Hmax (m)
$
    0.10  0.01  270. 2  1. 0.5 1. 0.5 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.) Ym and
$   spread (degr.)
$   alfa, sigA, sigB give default values if less than or equal to 0.
$
$   0.0  0.2  270.  3.3 0. 0. 1. 0.5 1. 1.
$
$ 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

4.3.4 The field preprocessor for the generic shell

Program	:	ww3_prep	(W3PREP)	
Code	:	ww3_prep.ftn		
Input	:	ww3_prep.inp	: Formatted input file for program.	
		mod_def.ww3	: Model definition file.	
		'user input'	: See example below.	(opt)
Output	:	standard out	: Formatted output of program.	
		level.ww3	: Water levels file.	(opt)
		current.ww3	: Current fields file.	(opt)
		wind.ww3	: Wind fields file.	(opt)
		ice.ww3	: Ice fields file.	(opt)

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 (w3fldg.ftn). For additional documentation and file formats reference if made to this routine.

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.
$   Format types :  AI    Transfer field 'as is'.
$                   LL    Field defined on longitude-latitude grid.
$                   F1    Arbitrary grid, longitude and latitude of
$                         each grid point given in separate file.
$                   F2    Like F1, composite of 2 fields.
$   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.
$
   19680605 120000
$
$ Additional input format type 'LL' ------------------------------- $
$ Grid range (degr.) and number of points for longitudes and latitudes,
$ respectively.
$
    -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 latitude-longitude information :
$ 1) Discrete size of input grid (NXI,NYI) and flag identifying closure
$    in longitudes.
$ 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'
$
$ 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),
$
   1. 1. 1. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0.
   1. 1. 1. 1. 1. 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.
   1. 1. 1. 1. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0.
$
```

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

end of example input file

4.3.5 The generic shell

Program	:	ww3_shel	(W3SHEL)	
Code	:	ww3_shel.ftn		
Input	:	ww3_shel.inp	: Formatted input file for program.	
		mod_def.ww3	: Model definition file.	
		restart.ww3	: Restart file.	
		nest.ww3	: Boundary conditions file.	(opt)
		level.ww3	: Water levels file.	(opt)
		current.ww3	: Current fields file.	(opt)
		wind.ww3	: Wind fields file.	(opt)
		ice.ww3	: Ice fields file.	(opt)
		track_i.ww3	: Output track information.	(opt)
Output	:	standard out	: Formatted output of program.	
		log.ww3	: Output log of wave model (see section 4.2).	
		test.ww3	: Test output of wave model.	(opt)
		restart*n*.ww3	: Restart file(s).	(opt)
		nest*n*.ww3	: Nesting file(s).	(opt)
		out_pnt.ww3	: Raw output of spectra.	(opt)
		out_grd.ww3	: Raw output of gridded fields.	(opt)
		track_o.ww3	: Raw output of spectra along tracks.	(opt)

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; four input lines.
$ 1) Water levels
$ 2) Currents
$ 3) Winds
$ 4) Ice concentrations (cannot be homogeneous).
$
   F T
   F F
   T T
   T
$
$ Time frame of calculations ---------------------------------------- $
```

```
$ - Starting time in yyyymmdd hhmmss format.
$ - Ending time in yyyymmdd hhmmss format.
$
   19680606 000000
   19680606 030000
$
$ 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  19680607 000000
     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).
$                               The raw data file is out_pnt.ww3,
$                               see w3iogo.ftn for additional doc.
$
   19680606 000000   3600  19680607 000000
$
$   -0.25 -0.25 'Land      '
     0.0   0.0  'Point 1   '
     2.0   1.0  'Point 2   '
     2.1   1.9  'Point 3   '
$    5.0   5.0  'Outside   '
     0.0   0.0  'STOPSTRING'
$
$ Type 3 : Output along  track.
$          Flags for formatted input files.
```

```
$                       The data files are track_i.ww3 and
$                       track_o.ww3, see w3iotr.ftn for ad. doc.
$
   19680606 000000   1800  19680606 023000
     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 010000   3600  19680606 030000
$
$ 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    0.0
   'CUR' 19680606 073125    2.0    25.
   'WND' 19680606 000000   20.    145.    2.0
$  'WND' 19680606 030000   20.    270.  -10.0
   'STP'
$
$ ---------------------------------------------------------------------- $
$ End of input file                                                      $
$ ---------------------------------------------------------------------- $
```

end of example input file

4.3.6 Gridded output post-processor

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

In its present form, WWATCH is not supplied with a graphical output options, because these tend to be at least to some level system dependent. The user could make his or her own graphical post-processors by reading the raw data files, but it is probably easier to generate graphical output from the transfer file as described in the example input below. The latter files are furthermore less likely to change with future releases of WWATCH than the former.

start of example input file

```
$ ------------------------------------------------------------------- $
$ WAVEWATCH III Grid output post-processing                            $
$------------------------------------------------------------------- $
$ Time, time increment and number of outputs
$
  19680606 000000  10800. 3
$
$ 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 F F F  T F F T F  T F F F F  F
$ T T T T T  T T T T T  T T T T T  T
$
$ 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

4.3.7 Point output post-processor

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

In its present form, WWATCH is not supplied with a graphical output options, because these tend to be at least to some level system dependent. The user could make his or her own graphical post-processors by reading the raw data files, but it is probably easier to generate graphical output from the transfer file as described in the example input below. The latter files are furthermore less likely to change with future releases of WWATCH than the former.

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. 4
$
$ Points requested ------------------------------------------------- $
$ Define points for which output is to be generated.
$
$ 1
  2
$ 3
$ 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   -1.  0.  33  F
$
$ The transfer file contains records with the following contents.
$
$ - File ID in quotes, nubmer 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.
```

```
$
$ 2  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.
$
$ 4 0. -1. 50   T T T T T T  0  F
$
$ The transfer file contains records with the following contents.
$
$ - File ID in quotes, nubmer of frequencies, directions and points,
$   flages 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.

4.3.8 Track output post-processor

Program	:	ww3_trck		(W3TRCK)
Code	:	ww3_trck.ftn		
Input	:	track_o.ww3	:	Raw track output data.
Output	:	standard out	:	Formatted output of program.
		track.ww3	:	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 presented is printed :

- Date and time in `yyyyddmm 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).

5 Installing the wave model

5.1 Introduction

WWATCH is written in ANSI standard FORTRAN-77, with a minimum of machine-dependent elements. For all machine dependent calls, dummy versions are available, so that WWATCH can be installed without modifications on most platforms. WWATCH utilizes its own preprocessor to process include files with PARAMETER and COMMON statements, 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 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.

5.2 Installing files

In its packed 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.

These files may be compressed with the standard UNIX commands `compress` or `gzip`, in which case the extension .Z or .gz is added. Such files can be unpacked with the standard UNIX commands `uncompress` or `gunzip`.

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. 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
```

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 makefiles.
inp	Input files.
obj	Object modules of all subroutines of WWATCH.
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 FORTRAN source codes of auxiliary programs, using the compiler as defined by the user in the setup file.

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 4.3.7).

The above source codes are stored in the directory aux and the executables are stored in the directory bin. A more detailed description (including instructions on running the executables) of these programs can be found in the documentation included in the above source code files. After the compilation of these three 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.
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.subs	Script to generate the part the of the makefile that controls compilation of all subroutines.
make_makefile.prog	Script to generate the part the of the makefile that controls linking of programs (depends on data 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. 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. The install program will furthermore attempt to generate a partial makefile makefile.subs in the directory ftn using the script make_makefile.subs in the directory bin. This makefile contains dependency information for all WWATCH subroutines.

If the installation of the input files is requested (file wwatch3.inp.tar), the install program will copy input files 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.

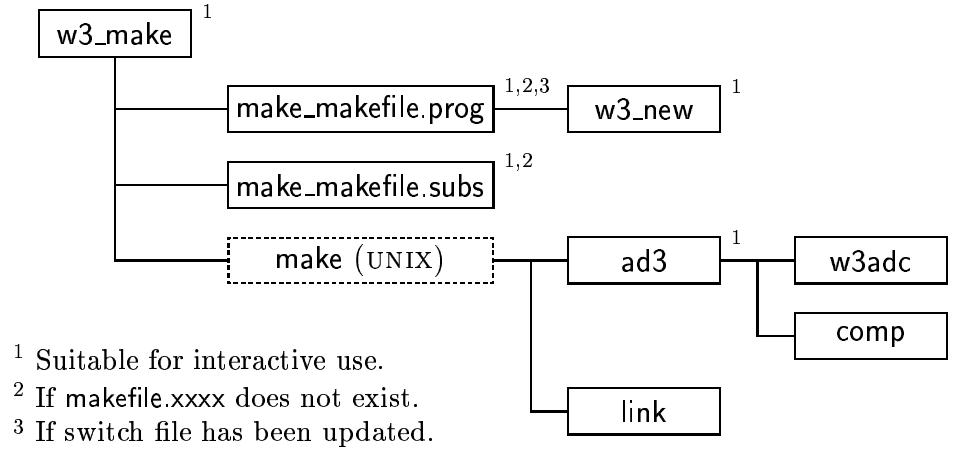

Figure 5.1: General layout of the compiler program w3_make.

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[3]. 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 many 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.prog and make_makefile.subs to generate a makefile. make_makefile.prog generates a

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

list of subroutines to be linked, based on the program switches in the file switch (see section 5.4). 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. 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 service routine STRACE. First go to the directory work and make a link to the source code of this routine by typing

```
ln3 strace
```

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 strace
```

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

```
ad3 strace 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/strace.o is generated. If requested during the initial set up, a source code and listing file (strace.f and strace.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 strace.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 strace.ftn have been removed, the link to the work directory 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
```

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

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. 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 :

DUM	Dummy to be used if WWATCH is to be installed on previously untried hardware.
ABS	Absoft compiler on Linux.
XLF	IBM xlf compiler.
NEXT	Intell based NeXT using Absoft compiler.
LNX	Red Hat Linux with fort77 or g77.
SGI	Silicone Graphics workstation.
HP	Hewlett Packard work stations.
YMP	Cray YMP, C90 or comparable using f77 compiler.

C90 Cray YMP, C90 or comparable using f90 compiler.

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 (most hardware).
DIST Distributed memory model (massively parallel hardware).

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 discrete grid dimensions. The maximum grid dimensions are set in dimx.cmn using PARAMETER statements. However, the grid dimensions do not have to be changed by altering the code, but can be changed by selecting one of the following compiler switches (see file main.cmn).

GRTST Slot for grids for general testing.
GRDn User-defined grids. Grids which are often used can be stored here. 9 slots are provided, addition of slots is discussed in section 5.5.
TPn Predefined grids for propagation tests (see documentation in test scripts in directory test).
TSn Predefined grids for source terms tests.

Selection of discrete spectral dimensions, defined by PARAMETER statements in dims.cmn.

SPTST Slot for spectra grid for general testing.
SPn User-defined spectral grids. Slots 1-4 available.
SPPn Predefined spectral grids for propagation tests (see documentation in test scripts in directory test).
SPSn Predefined grids for source terms tests.

Parameter statements defining the maximum number of output points in dimo.cmn.

OUTPn	Slots 0-5 to be used with the test grid and the grids for test cases. Presently set to 1 (minimum), 10 20, 50, 100 and 200, respectively.
GRDn	Settings directly hardwired to user-defined spatial grids.

Parameter statements defining the maximum number of input boundary points in dimo.cmn.

BPIn	Slots 0-2 to be used with the test grid and the grids for test cases. Presently set to 1 (minimum), 25 and 250, respectively.
GRDn	Settings directly hardwired to user-defined spatial grids.

Parameter statements defining the maximum number of input boundary points in dimo.cmn.

BPOn	Slots 0-2 to be used with the test grid and the grids for test cases. Presently set to 1 (minimum), 25 and 250, respectively.
GRDn	Settings directly hardwired to user-defined spatial grids.

Selection of propagation schemes:

PR0	No propagation scheme used.
PR1	First order propagation scheme.
PR2	ULTIMATE QUICKEST propagation scheme.
PRX	Experimental (user supplied).

Selection of input and dissipation:

ST0	No input and dissipation used.
ST1	WAM3 source term package.
ST2	Tolman and Chalikov source term package.
STX	Experimental (user supplied).

Selection of nonlinear interactions:

NL0	No nonlinear interactions used.
NL1	Discrete interaction approximation (DIA).
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.

5.4.2 Optional switches

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

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.
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.
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.
OMP2	Paralelization of inter-processor communication. NOT YET IMPLEMENTED.

Furthermore the following miscellaneous switches are available:

DSS0	Switch off frequency dispersion in diffusive dispersion correction.
S	Enable subroutine tracing in the main WWATCH subroutines by activating calls to the subroutine STRACE.
STAB2	Enable stability correction (2.63) - (2.66) for Tolman and Chalikov (1996) source term package.
RWND	Correct wind speed for current velocity.
SEED	Seeding of high-frequency energy.
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.
XW0	Swell diffusion only in ULTIMATE QUICKEST scheme.
XW1	Id. wave growth diffusion only.

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, adding new subroutines or new include files requires modification of the automated compilation scripts. If a new subroutine in a new file is added to WWATCH, the following steps are required to include the script in the automatic compilation:

1) Add the file name to the script make_makefile.subs and run this script to update the corresponding part of the makefile.
2) Add the file name to the script make_makefile.prog to assure that the file is included in the makefile under the correct conditions (will be invoked automatically by w3_make).

For details of inclusion, see the actual scripts. Adding a new include file requires the following actions:

1) Add the name of the include file to the scripts ad3 and ad3_test in the bin directory.
2) Add the identifier for the include file to make_makefile.subs and run this program.

3) Check if this include file needs to be touched if switches for the preprocessor are changed, and modify w3_new accordingly.

Adding a new switch to the compilation systems requires the following actions:

1) Put switch in required source code files.
2) Add a new 'keyword' and names of files to be touched in w3_new if necessary.
3) Update make_makefile.prog with the 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
```

where the `more` command is added to allow for on-screen inspection of the output. All intermetiate output files are placed in the work directory, and can be removed conveniently by typing

```
w3_clean
```

Also available are several test cases[4] 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 case 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.

[4] All scripts for running test cases presently expect that the model has been compiled as a shared memory model.

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 and the contents of the COMMON blocks (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; PARAMETER statements and COMMON blocks still have to be included, machine dependent code and compile-level program switches have to be selected, and test output may be activated. The arbitrary switch 'SWT' is included in the WWATCH files as comment of the form C/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 switches C/SWT can be found by typing

```
find_switch C/SWT
```

Inclusion of COMMON's and activation of switches is performed by the preprocessing program `w3adc`. This program is found in the file w3adc.f, which contains a ready to compile FORTRAN source code and a full documentation. 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

```
0 1
'w3wave.ftn'   'w3wave.f'
'DUM TS2 SPS2 PR1 ST1 NL1 BT1 WND1 CUR1 PNT1 BPI0 BPO0'
'##dimx.cmn'   'dimx.cmn'
'##dims.cmn'   'dims.cmn'
'##dimo.cmn'   'dimo.cmn'
'##spar.cmn'   'spar.cmn'
'##npar.cmn'   'npar.cmn'
'##sinp.cmn'   'sinp.cmn'
'##snl4.cmn'   'snl4.cmn'
'##sdis.cmn'   'sdis.cmn'
'##sbot.cmn'   'sbot.cmn'
'##tab1.cmn'   'tab1.cmn'
'##tab2.cmn'   'tab2.cmn'
'##trace.cmn'  'trace.cmn'
'##aux1.cmn'   'aux1.cmn'
'##auxt.cmn'   'auxt.cmn'
'##outp.dat'   'outp.dat'
```

Figure 6.1: Example input for W3ADC.

→ 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 COMMON file and name of COMMON 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 'C', except for empty switches, i.e., lines starting with 'C/'. A compress indicator 2 results in the subsequent removal of comment lines marked by '*',

whereas a value 3 results in the 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. For instance, C/C90/CMIC$ becomes CMIC$ is the C90 switch is selected, and is not removed by the file compression option.

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.

Note furthermore that many of the source code files have an include string `##docb` near the top of the file. This string is used internally at NCEP to provide additional documentation for the operational implementation of the model. In the distributed version, this string is simply replaced by 'C/' by ad3.

6.3 Program files

The WWATCH source code files are stored in files with the extension ftn. Separate subroutines are stored in separate files, to accommodate easy development of the program using '`make`' facilities, and to allow for switching on of test output in selected routines. The COMMON and DATA files to be included are stored in files with the extensions cmn and dat, respectively.

Below, the contents of all WWATCH source code files is described briefly. A source code file xxxx.ftn only contains the subroutine XXXX unless specified differently. The relation between the various subroutines is graphically depicted in Figs. 6.2 through 6.6. First, however, the include files will be described. The use of include files by the various subroutines can be traced using the shell script `find_switch`. For instance, typing

```
find_switch dimx.cmn
```

will provide a list of subroutine files that require this include file.

6.3.1 Include files

The WWATCH source code files require the following include files with PARAMETER statements, COMMON declarations and DATA statements:

dimx.cmn	PARAMETER statements defining discrete spatial grid.
dims.cmn	PARAMETER statements defining discrete spectral grid.
dimo.cmn	PARAMETER statements defining output arrays.
spar.cmn	Variables describing discrete spectrum.
npar.cmn	Numerical parameters.
sinp.cmn	Variables in input source term.
snl4.cmn	Variables in nonlinear interactions.
sdis.cmn	Variables in dissipation.
sbot.cmn	Variables in bottom friction.
mpp.cmn	Variables for distributed / shared memory model versions.
mpi.cmn	Varaibels for MPI communication calls.
tab1.cmn	Interpolation tables for solution of dispersion relation.
tab2.cmn	Interpolation tables for β in the input source term of Chalikov and Belevich.
trace.cmn	Variables governing subroutine tracing by subroutine STRACE.
aux1.cmn	Physical and mathematical constants.
auxt.cmn	Unit number of general and test data.
outp.dat	DATA statements used to identify output fields.

6.3.2 Main program elements

As discussed above, the actual wave model WWATCH consists of two main subroutines. The files with these subroutines are

w3init.ftn	The initialization routine W3INIT, which prepares the wave model for computations.
w3wave.ftn	The actual wave model W3WAVE.

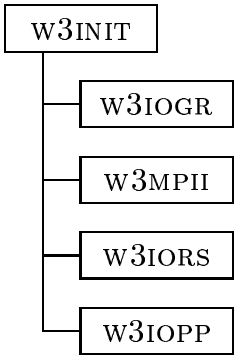

Figure 6.2: Subroutine structure for initialization routine without service and MPI routines).

6.3.3 Second level routines

The routines listed in the previous section mainly manage the wave model. Most of the actual calculations are performed by the so-called second level routines. The files with second level routines are.

w3xyp1.ftn	Propagation in physical space using the first order scheme.
w3xyp2.ftn	Id. using ULTIMATE QUICKEST scheme.
w3xypx.ftn	Id. dummy routine (slot for user-supplied routine).
w3ktp1.ftn	Propagation in spectral space (first order scheme).
w3ktp2.ftn	Id. using ULTIMATE QUICKEST scheme.
w3ktpx.ftn	Id. dummy routine (slot for user-supplied routine).
w3qck$n$.ftn	Routines performing ULTIMATE QUICKEST scheme in arbitrary spaces (1: regular grid. 2: irregular grid).
w3map$n$.ftn	Generation of auxiliary maps for the propagation routines n = 1,2,x.
w3apr1.ftn	Calculation of mean wave parameters used in WAM-3 source terms (entire grid).
w3apr2.ftn	Id. Tolman and Chalikov source terms.
w3aprx.ftn	Id. dummy routine.
w3spr1.ftn	Calculation of mean wave parameters used in WAM-3 source terms (single grid point).
w3spr2.ftn	Id. Tolman and Chalikov source terms.

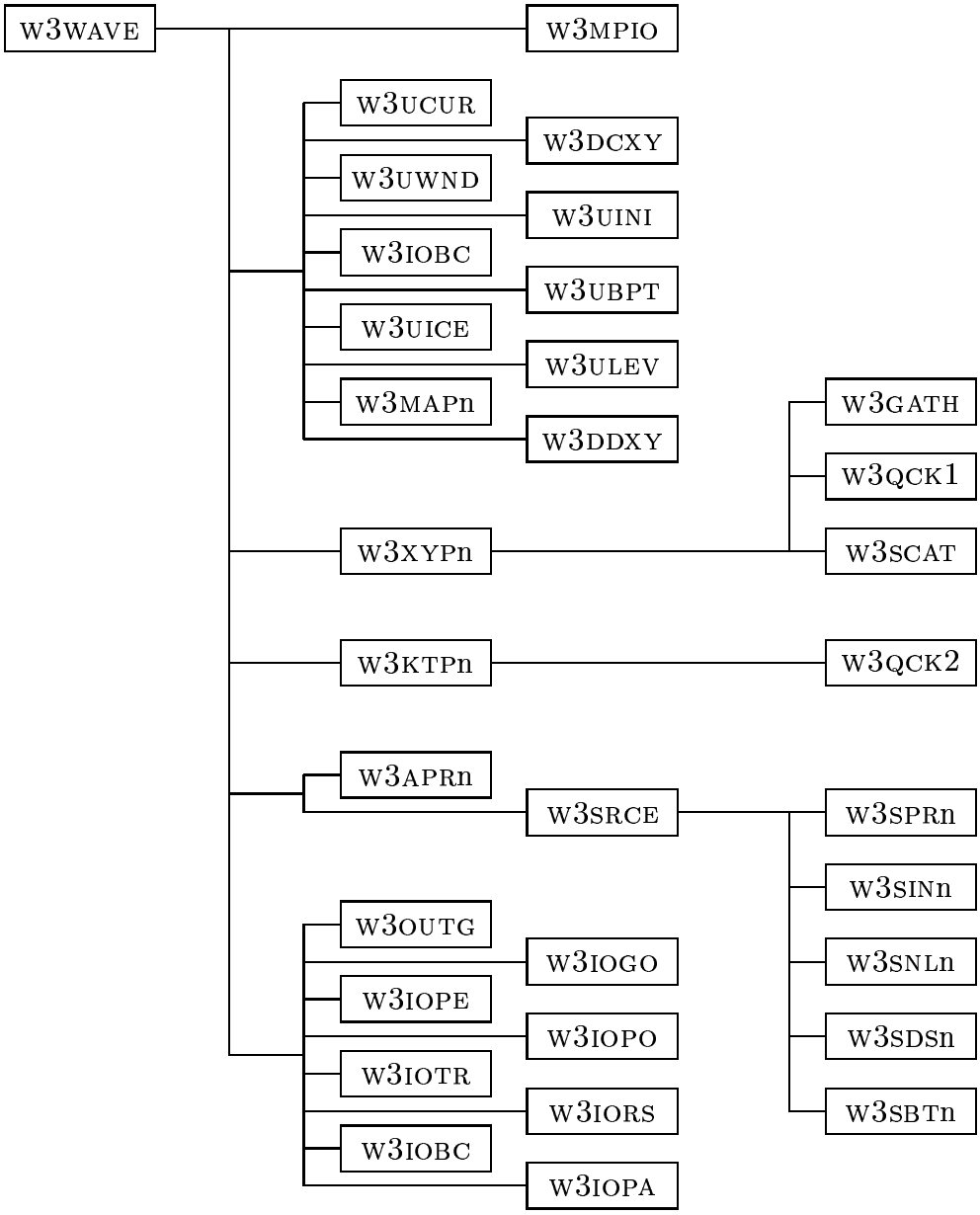

Figure 6.3: Subroutine structure for wave model routine without service and MPI routines.

w3sprx.ftn	Id. dummy routine.
w3srce.ftn	Integration of source terms.
w3sin1.ftn	Calculation of S_{in} (WAM-3).
w3sin2.ftn	Calculation of S_{in} (Tolman and Chalikov).
w3sinx.ftn	Calculation of S_{in} (dummy routine).
w3snl1.ftn	Calculation of S_{nl} (DIA).
w3snlx.ftn	Calculation of S_{nl} (dummy routine).
w3snp$n$.ftn	Preparations for nonlinear interaction; n = 1,x.
w3sds1.ftn	Calculation of S_{ds} (WAM-3).
w3sds2.ftn	Calculation of S_{ds} (Tolman and Chalikov).
w3sdsx.ftn	Calculation of S_{ds} (dummy routine).
w3sbt1.ftn	Calculation of S_{bot} (JONSWAP).
w3sbtx.ftn	Calculation of S_{bot} (dummy routine).
w3gath.ftn	Data transpose to gather data for spatial propagation in a single array.
w3scat.ftn	Corresponding scatter operation.
w3mpii.ftn	Corresponding MPI initializations.

6.3.4 Input field update routines

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 files

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

6.3.5 I/O and related routines

WWATCH requires a multitude of data files, as is illustrated in figure 4.1. Both reading and writing of every WWATCH data file is managed by a

single subroutine, which simplifies the construction of ad hoc pre- and post-processing programs. The following files with I/O routines, and routines preparing output data are available

w3iogr.ftn	Reading and writing of mod_def.ww3.
w3iors.ftn	Reading and writing of restart$n$.ww3.
w3outg.ftn	Calculation of gridded output parameters.
w3iogo.ftn	Reading and writing of out_grd.ww3.
w3iopp.ftn	Processing of requests for point output.
w3iope.ftn	Calculating point output data.
w3iopo.ftn	Reading and writing of out_pnt.ww3.
w3iotr.ftn	Generate track output in track_o.ww3.
w3iobc.ftn	Reading and writing of nest$n$.ww3.
w3iopa.ftn	Providing gridded output to main program.
w3mpio.ftn	MPI initializations for I/O.

6.3.6 Interpolation tables

The dispersion relation (2.1) and the Doppler equation (2.2) require implicit solutions when solved for known σ or ω. Due to the selected model equations, the model never requires implicit solutions of (2.2). Implicit solutions of (2.1), however, are required regularly, because the spectral grids are defined in terms of the discrete frequencies σ. To assure accurate and economic solutions of the dispersion relations, the following routines and files are available. The calculation of the wind-wave interaction parameter β in the Chalikov and Belevich input term is rather complicated and can be performed by a function. To speed up the model, and interpolation table for β is also available. Note that these routines only use their own include file (tab1.cmn or tab2.cmn), and can therefore easily be used by any other program.

wavnu1.ftn	Solve (2.1) for given σ using interpolation tables.
wavnu2.ftn	Iterative solution of (2.1) for given σ using a Newton-Raphson scheme.
distab.ftn	Generation of interpolation tables used by WAVNU1.
w3beta.ftn	Function to calculate β.
inptab.ftn	Generation of the interpolation table for β.

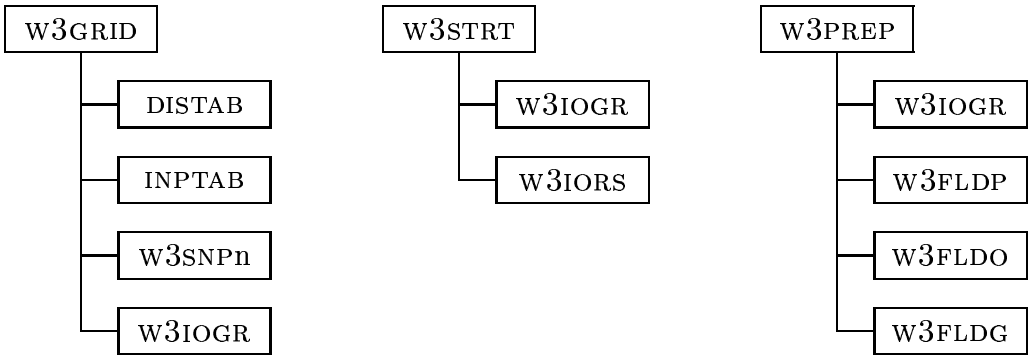

Figure 6.4: Subroutine structure of pre-processors

6.3.7 Service routines

Several auxiliary routines are available in the following files. Note that several of these files do not require WWATCH include files, so that they can easily be used by any other program.

w3s2xy.ftn	Convert data from storage grid (sea points only) to full spatial grid.
w3wrt$x$.ftn	Routines for optimization of writing to files.
strace.ftn	Routine to enable subroutine tracing.
nextln.ftn	Reading of input file to find next non-comment line.
fej5p.ftn	Calculation of energy density $F(f)$ of a onedimensional five-parameters JONSWAP spectrum.
extcde.ftn	Abort model with exit code (machine dependend.
w3mpix.ftn	Close pending MPI communication for abort.
tick21.ftn / dsec21.ftn / stme21.ftn / iymd21.ftn / mymd21.ftn	Routines for management of time in yyyymmdd and hhmmss formats.
timesd.ftn / datesd.ftn	Routines to get machine-dependent date and time of the model run.
ina2i.ftn / ina2r.ftn / outa2i.ftn outa2r.ftn	Auxiliary routines for array I/O.

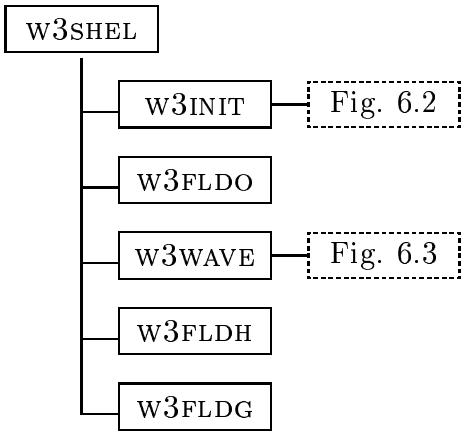

Figure 6.5: Subroutine structure for the generic program shell.

---

pr_arr.ftn / pr_blk.ftn / pr_mat.ftn
: Auxiliary routines for printed output of arrays.

prt1dm.ftn / prt1ds.ftn / pro2ds.ftn / angstr.ftn
: Print-plots of spectra and two-dimensional fields.

6.3.8 Main programs

Finally, WWATCH has several auxiliary pre- and post-processors, and a generic stand-alone shell (see section 4.3). These main programs and some additional routines are stored in the following files. Note that the routines managing the data flow to the shell (W3FLDP through W3FLDG) do not use WWATCH include files, and can therefore be used directly in other programs to generate input data for the generic shell.

ww3_grid.ftn	The grid preprocessor W3GRID.
ww3_strt.ftn	The initial conditions program W3STRT.
ww3_prep.ftn	Pre-processor W3PREP for the input fields for the generic shell.
ww3_shel.ftn	The generic program shell W3SHEL.

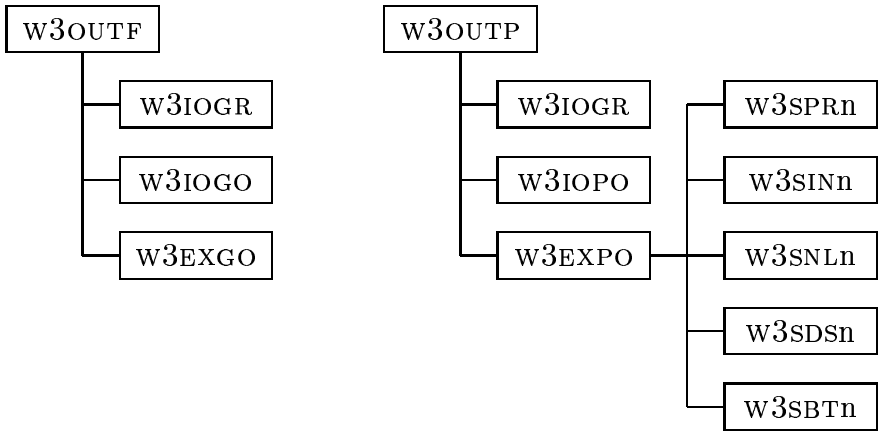

Figure 6.6: Subroutine structure for the postprocessors.

---

w3fldp.ftn	Prepare interpolation of input fields from arbitrary grids.
w3fldo.ftn	Opening and checking of data files for W3SHEL.
w3fldg.ftn	Reading and writing of data files for W3SHEL.
w3fldh.ftn	Management of homogeneous input fields in W3SHEL.
ww3_outf.ftn	The post-processing program for gridded fields of mean wave parameters W3OUTF.
w3exgo.ftn	Routine performing output requests in W3OUTF.
ww3_outp.ftn	The post-processing program output at selected locations W3OUTP.
w3expo.ftn	Routine performing output requests in W3OUTP.
ww3_trck.ftn	Converting unformatted direct access track output file to integer-packed formatted file.

6.4 Optimization

The source code of WWATCH is written in ANSI standard FORTRAN-77[5] and has been compiled and run on a variety of platforms ranging from PC's to supercomputers. Optimization for super-computers has been performed by structuring the code in long vector loops where possible. Optimization was performed for the Cray YMP and C90. Note that some compiler directives for vectorization have been used. Parallelization for shared memory machines using threading has been implemented using either Cray specific compiler directives CFPP$ and CMIC$ or more general OpenMP directives. Such parallelization takes place mainly in the loop calling the source term routine W3SRCE and the different propagation routines. Parallelization on Cray platforms furthermore required the inclusion of CFPP$ and CMIC$ compiler directives in nearly all subroutines (to avoid micro-tasking in most routines). Cray or OpenMP directives are activated by the corresponding preprocessor switches (C/YMP, C/C90 and OMP*n*, respectively). 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 (see section 6.3.6).

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. 6.7. Longitudes are denoted throughout the program by the counter IX, and latitudes by the counter IY. The user-defined size of the grid (NX,NY) may be smaller than the declared size (MX,MY). The closure of the grid in case of a global applications is handled

[5] with the exception of some recent MPI additions, which include some FORTRAN-90 elements.

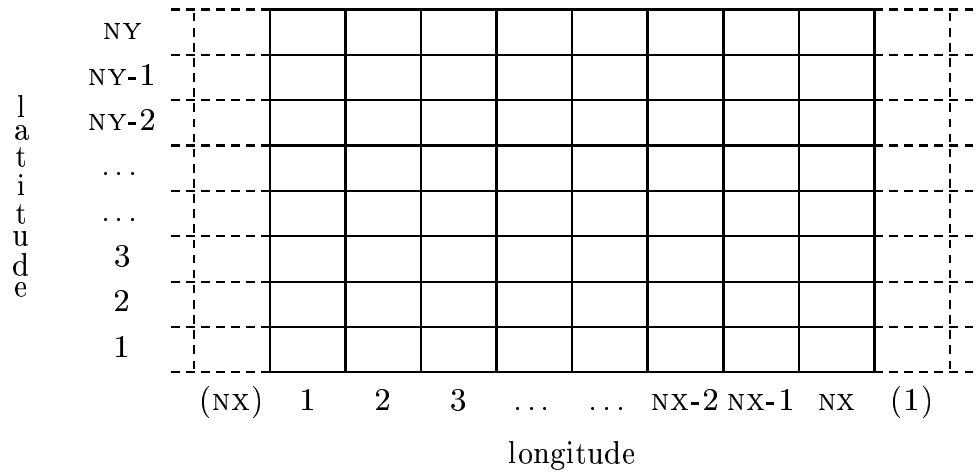

Figure 6.7: Layout of the spatial grid. Grid points are denoted as boxes, dotted boxes denoted repeated collumns for global model applications.

---

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

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

and are read row by row (see also chapter 4). Within the program, however, 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. 6.7) does not influence this one-dimensional counter IXY, the array ARRAY, its one-dimensional equivalent VARRAY and IXY are defined as

$$\text{ARRAY}(\text{MY},\text{MX}) \quad , \quad \text{VARRAY}(\text{MY} * \text{MX}) \quad ,$$

$$\text{IXY} = \text{IY} + (\text{IX} - 1) * \text{MY} \quad .$$

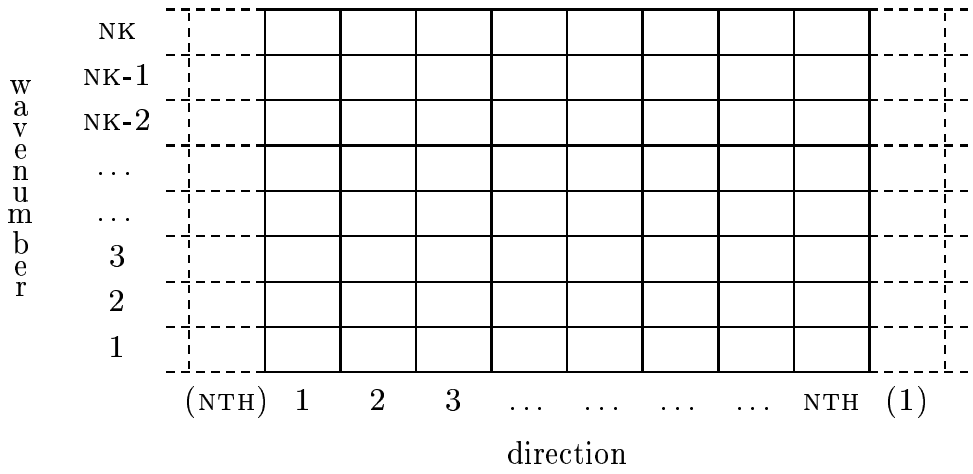

Figure 6.8: Layout of the spectral grid. Dotted boxes denoted repeated collumns for directional closure.

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. 6.8). Unlike with the spatial grid, however, the declared dimensions NK and NTH are the actual discrete dimensions of the spectrum, to assure optimum use of memory. As with the spatial grid, the internal description of the spectrum A is defined as

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

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 IX direction), and $\theta = 90^\circ$ corresponds to propagation from south to north (positive 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

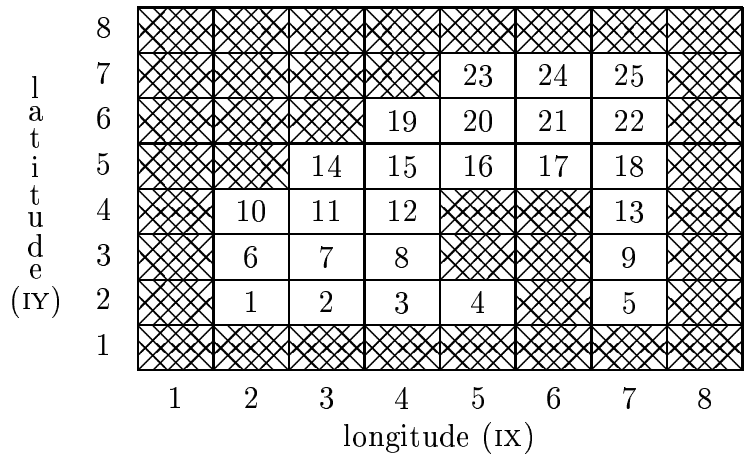

Figure 6.9: An example of the onedimensional storage grid for spectra. Hatched grid boxes denote land points. Numbers within the grid boxes show the grid counter ISEA of the storage grid.

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

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

An example of the layout of this storage grid in relation to the full grid of Fig. 6.7 is given in Fig. 6.9. 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.

$$\mathrm{MAPSF}(\mathrm{ISEA}, 1) = \mathrm{IX} \; ,$$

$$\mathrm{MAPSF}(\mathrm{ISEA}, 2) = \mathrm{IY} \; ,$$

$$\mathrm{MAPSF}(\mathrm{ISEA}, 3) = \mathrm{IXY} \; .$$

$$\mathrm{MAPFS}(\mathrm{IY}, \mathrm{IX}) = \mathrm{VMAPFS}(\mathrm{IXY}) = \mathrm{ISEA} \; ,$$

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

$$\text{ISEA} = \text{IAPROC} + (\text{JSEA} - 1)\text{NAPROC}\,,$$

$$\text{JSEA} = 1 + (\text{ISEA} - 1)/\text{NAPROC}\,,$$

$$\text{IAPROC} = 1 + \text{MOD}(\text{ISEA} - 1, \text{NAPROC})\,.$$

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

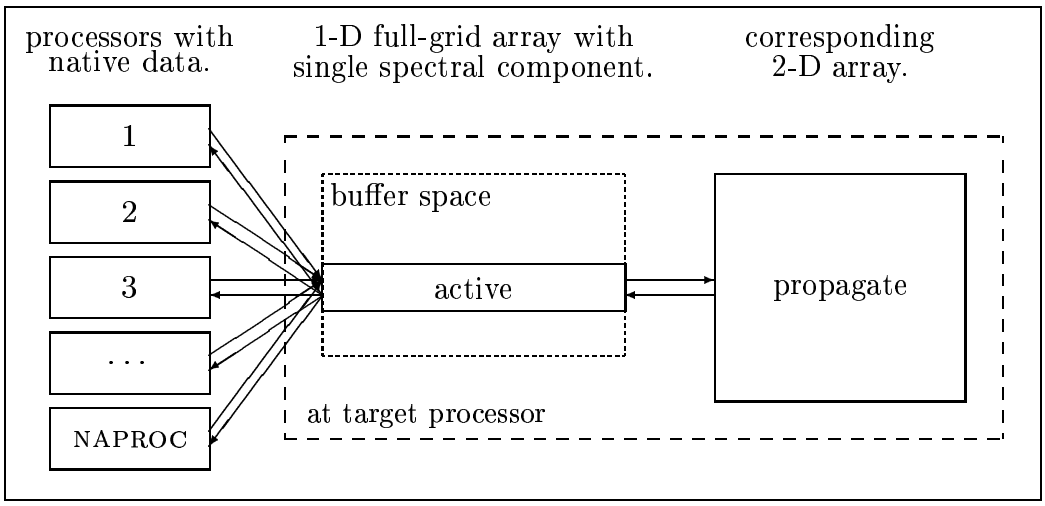

Figure 6.10: Data transpose in distributed memory model version. First, the data is moved from left to right in the figure during the gather operation. After the calculation is performed, the data is moved from right to left in the scatter operation.

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 (w3init.ftn).

The data transpose for the gather operation is implemented in two steps. 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 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 w3gath.ftn and w3scat.ftn, respectively). 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. 6.10.

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.

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 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 (as will be presented

elsewhere).

Finally it should be noted that only that MPI versions are only necessary and/or available for the main wave model and the initial conditions program. All other source codes remain single threaded. This may imply that the separate sets of object codes of individual subroutines are required for different executables.

6.5.3 Variables in include files

To assure that multiple grids can be used simultaneously in a specialized program shell, most data transfer to and from the wave model occurs through the parameter lists of its main routines W3INIT and W3WAVE[6]. For such applications it is assumed that spectral grids and source terms used are identical. To minimize the number of addresses to be transferred through the parameter list, the corresponding variables are stored in COMMON blocks. Furthermore, maximum array sizes are set in PARAMETER statements to allow for the use of local work arrays, which do not have to be passed through the parameter lists. Below, all variables in PARAMETER statements and all variables in named COMMON blocks are described briefly. The file name of the file containing the variable is given at the right margin of the start of each list. The second column of each list identifies the type of the parameter. I, R, L and C represent integer, real, logical and character, respectively and A identifies an array. PARAMETER statements might require modification by the user for custom applications. Minimum values as dictated by algorithms or by declaration requirements are given below. All other parameter statements are internal to the model and should not be modified.

Variables in PARAMETER statements (spatial grid). dimx.cmn

MX,MY	I	Maximum dimensions of the grid as shown in Fig. 6.7, MX $\geq$ 3, MX $\geq$ 3.
MSEA	I	Maximum number of sea points, MSEA $\geq$ 1.
MSEAL	I	Id., local for distributed memory version, MSEAL $\geq$ 1.

[6] This option is not yet included, and will need additional modifications to the code, in particular considering file names for parallel grids

Variables in PARAMETER statements (spectral grid). dims.cmn snl4.cmn

NTH,NK	I	Actual dimensions of the spectral grid as shown in Fig. 6.8, NTH ≥ 4, NK ≥ 3.
NSPEC	I	NSPEC = NTH*NK
NFR	I	Number of frequencies in the DIA, NFR = NK.
NFRADM	I	Size of expansion of spectrum for use in the DIA.
NFRMAX, NSPEX	I	Auxiliaries

Variables in PARAMETER statements (communication buffer). mpi.cmn

MPIBIF	I	Buffer depth for data transpose in distributed memory version. MPIBUF = 1 disables buffering. MPIBUF ≥ 1.

Variables in PARAMETER statements (output arrays). dimo.cmn

NOGRID	I	Number of field output options, do not change.
MOPTS	I	Maximum number of output points, MOPTS ≥ 1.
MBI	I	Maximum number of grid points with input boundary data, MBI ≥ 1.
MBI2	I	Maximum number of spectra in input boundary data file MBI2 ≥ 1.
MBO	I	Maximum number of points with output boundary data, MBO ≥ 1.
MBO2	I	Maximum number of spectra in output boundary data file(s) MBO2 ≥ 1.

Variables in PARAMETER statements (tables). tab$n$.cmn

NAR1D	I	Dimension of interpolation table for solution of dispersion relation.
DFAC	R	Maximum nondimensional water depth kd in the interpolation table fro dispersion relation.
SIGAMX	R	Maximum nondimensional frequency in the interpolation table for the wind-wave interaction parameter β of Chalikov and Belevich.

DRAGMX	R	Id. maximum drag coefficient.	
NRSIGA	I	Id., dimensions of interpolation table.	
NRDRAG	I	Id.	

Variables in PARAMETER statements (version numbers). w3init.ftn w3io*xx*.ftn

WWVER	C	Version number of the main program.	
VERGRD	C	Version number of file mod_def.ww3.	
VEROGR	C	Version number of file out_grd.ww3.	
VEROPT	C	Version number of file out_pnt.ww3.	
VERTRK	C	Version number of file track_o.ww3.	
VERINI	C	Version number of file restart.ww3.	
VERBPT	C	Version number of file nest.ww3.	

Variables in PARAMETER statements (programs). ww3_*xxxx*.ftn

NFL	I	Dimension of array with program flags in ww3_grid.ftn (do not change).	
MXI,MYI	I	Maximum dimension of input fields in ww3_prep.ftn.	
NHMAX	I	Maximum number of homogeneous fields in shell.	
NFO	I	Maximum number fields in parameter list output of W3WAVE in shell.	

Variables in PARAMETER statements (service routines). prt1ds.ftn prt1dm.ftn

NFRMAX	I	Maximum number of frequencies in print-plot of spectra.	
NFM2	I	Auxiliary.	

Variables in COMMON /CW3SPR/ (spectral description) spar.cmn

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 $F(f)$.

Variables in COMMON /CW3MPP/ (Distributed data parameters) mpp.cmn

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.
NSEALM	I	Number of spectra locally stored.
IAPPRO	IA	Map with processor number at which propagation for spectral component is performed.

Variables in COMMON /CW3MPI/ (MPI parameters) mpi.cmn

WW3_FIELD_VEC, WW3_SPEC_VEC

	I	Derived data types used in data transpose.
IRQSGn	IA	Handles for persistent non-blocking communication requests used in data transpose.
NRQSGn	I	Corresponding counters.
IBFLOC	I	Number of presently active buffer.
ISPLOC	I	Number ISP of the spectral component presently being propagated.
NSPLOC	I	Number of spectral components to be propagated at the present processor.
STORE	RA	Buffer array used in data transpose.
BSTAT	IA	Status indicator for buffers. 0: buffer presently not used. 1: buffer used for gathering data or active. 2: buffer used for scattering data or active.
BISPL	IA	Number ISP of spectral compnent for corresponding buffer number, ISP = BISPL(IBFLOC).
IRQGO	IA	Handles for persistent non-blocking communication requests used for output of mean wave parameters.
IRQPA	IA	Id. parameter list output in W3WAVE.
IRQPOn	IA	id. point output.
IRQRS	IA	Id. restart file.
IRQBPn	IA	id. boundary data output.
NRQxxx	I	Corresponding counters.

Variables in COMMON /CW3NPR/ (numerical parameters) npar.cmn

FACP	R	Composite constant in Eq. (3.57).
XR	R	X_r in Eq. (3.58).
XFILT	R	X_f in Eq. (3.59).
FXFM	R	First constant in Eq. (2.31).
FXPM	R	Second constant in Eq. (2.31).
XFT	R	Constant for f_{21} in Eq. (2.62).
XFC	R	Constant for f_{hf} in Eq. (2.62).
XSEED	R	X_{seed} in Eq. (3.60).

Variables in COMMON /CW3SIN/ (input source term) sinp.cmn

NITTIN	I	Number of iteration in initialization of u_* for Tolman and Chalikov input and dissipation.
SINC1	R	Proportionality constant in Eq. (2.26).
ZWND	R	Height of input wind for Tolman and Chalikov input.
CINXSI	R	χ in Eq. (2.40).
SWELLF	R	Swell regative input reduction factor in Eq. (2.48).
STABSH	R	c_0 in Eq. (2.63).
STABOF	R	$\mathcal{ST}_o$ in Eq. (2.63).
CNEG	R	c_1 in Eq. (2.64).
CPOS	R	c_2 in Eq. (2.65).
FNEG	R	f_1 in Eq. (2.64).
FPOS	R	f_2 in Eq. (2.65).

Variables in COMMON /CW3SNL/ (nonlinear interactions) snl4.cmn

SNLC1	R	Constant C in Eq. (2.20).
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) .
DELTHn	R	Relative angles of $\mathbf{k}_3$ and $\mathbf{k}_4$ in quadruplet.
DALn	R	μ-dependent factors in Eq. (2.20).
NFRHGH	I	Auxiliary frequency counter.
NFRCHG	I	Id.
NSPECX	I	Auxiliary spectral counter.

AF11	RA	Factor f^{11}.
IPnn	RA	Discrete addresses for quadruplet interpolation.
IMnn	RA	Id.
ICnn	RA	Id.
AWGn	R	Interpolation weights for spectrum.
SWGn	R	Interpolation weights for 'diagonal'.

Variables in COMMON /CW3SDS/ (dissipation source term) sdis.cmn

SDSC1	R	Composite constant in Eq. (2.28).
SDSAn	R	Constants a_n in Eq. (2.57).
SDSALN	R	α_r in Eq. (2.58).
SDSBn	R	Constants b_n in Eq. (2.51).
FPIMIN	R	Minimum value of $\tilde{\phi}_{p,i}$ allowed in Eq. (2.51).
XFH	R	Constant for f_h in Eq. (2.62).
XF1	R	Constant for f_1 in Eq. (2.62).
XF2	R	Constant for f_2 in Eq. (2.62).

Variables in COMMON /CW3SBT/ (bottom friction source term) sbot.cmn

SBTC1	R	Composite constant in Eq. (2.67).

Variables in COMMON /CW3TB1/ (interpolation tables) tab1.cmn

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.

Variables in COMMON /CW3TB2/ (interpolation tables) tab2.cmn

DSIGA	R	Nondimensional frequency increment.
DDRAG	R	Drag coefficient increment.
BETATB	RA	Interpolation table for the wind-wave interaction parameter β.

Variables in COMMON /CTRACE/ (subroutine tracing) trace.cmn

NDSTRC	I	Unit number for trace output.
ITRACE	I	Maximum number of trace outputs per subroutine.

Variables in COMMON /CW3PHS/ (physical and mathematical constants) aux1.cmn

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

Variables in COMMON /CW3TST/ (test info) auxt.cmn

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.