U. S. Department of Commerce
National Oceanic and Atmospheric Administration
National Weather Service
National Centers for Environmental Prediction
5830 University Research Court
College Park, MD 20740

Technical Note

User manual and system documentation of WAVEWATCH III ® version 6.07 †

The WAVEWATCH III ® Development Group ‡
(WW3DG)

Environmental Modeling Center
Marine Modeling and Analysis Branch

March 2019

To refer to this manual, please use the following citation:

The WAVEWATCH III® Development Group (WW3DG), (2019): User manual and system documentation of WAVEWATCH III® version 6.07. Tech. Note 333, NOAA/NWS/NCEP/MMAB, College Park, MD, USA, 465 pp. + Appendices.

---

† MMAB Contribution No. 333.

‡See Section 1.5 for WW3DG group description.

‡Code manager email: NCEP.List.WAVEWATCH@noaa.gov

This page is intentionally left blank.

Contents

This page is intentionally left blank.

1 Introduction

1.1 The WAVEWATCH III Modeling Framework

WAVEWATCH III® is a community wave modeling framework that includes the latest scientific advancements in the field of wind-wave modeling and dynamics.

The core of the framework consists of the WAVEWATCH III third-generation wave model, developed at the US National Centers for Environmental Prediction (NOAA/NCEP) in the spirit of the WAM model (Komen et al., 1994). The current framework evolved from earlier WAVEWATCH I & II model packages (Tolman, 1991, 1992), and differs from its predecessors in many important points such as governing equations, model structure, numerical methods and physical parameterizations.

WAVEWATCH III solves the random phase spectral action density balance equation for wavenumber-direction spectra. The implicit assumption of this equation is that properties of medium (water depth and current) as well as the wave field itself vary on time and space scales that are much larger than the variation scales of a single wave. The model includes options for shallow-water (surf zone) applications, as well as wetting and drying of grid points. Propagation of a wave spectrum can be solved using regular (rectilinear or curvilinear) and unstructured (triangular) grids, individually or combined into multi-grid mosaics.

Source terms for physical processes (source terms) include parameterizations for wave growth due to the actions of wind, exact and parametrized forms accounting for nonlinear resonant wave-wave interactions, scattering due to wave-bottom interactions, triad interactions, and dissipation due to whitecapping, bottom friction, surf-breaking, and interactions with mud and ice. The model includes several alleviation methods for the Garden Sprinkler Effect, and computes other transofrmation processes such as the effects of surface currents to wind and wave fields, and sub-grid blocking due to unresolved islands.

Inputs to WAVEWATCH III may be provided via external files or via coupling using the OASIS or ESMF/NUOPC frameworks. Input data is dynamically updated within the wave model driver, and may include ice coverage, mud, current fields, bottom properties for dissipation on a moveable

bed, and data for assimilation within a data assimilation placeholder module that may be developed by users.

WAVEWATCH III is written in ANSI standard FORTRAN 90, fully modular and fully allocatable. The model is set up for traditional one-way nesting, and also using a 'mosaic' or multiple-grid approach, where an arbitrary number of grids can be considered with full two-way interactions between all grids. Individual or multi-grid mosaics can be used as moving frame of reference that allows high-resolution modeling of hurricanes away from the coast.

Wave energy spectra are discretized using a constant directional increment (covering all directions), and a spatially varying wavenumber grid. First-, second- and third-order accurate numerical schemes are available to describe wave propagation. Source terms are integrated in time using a dynamically adjusted time stepping algorithm, which concentrates computational efforts in conditions with rapid spectral changes. WAVEWATCH III can optionally be compiled to include shared memory parallelisms using OpenMP compiler directives, and/or for a distributed memory environment using the Message Passing Interface.

1.2 About this manual

This is the user manual and system documentation of version 6.07 of the WAVEWATCH III ®wind-wave modeling framework. Although code management of the framework is primarily undertaken by NCEP/NOAA, the model development itself relies on a community of developers, the WAVEWATCH III Development Group (WW3DG) with membership indicated below. This manual is describes the wave modeling framework as follows.

- **Chapter 2**: Governing equations,
- **Chapter 3**: Numerical approaches,
- **Chapter 4**: Model structure and data flow,
- **Chapter 5**: Installing, compiling and running,
- **Chapter 6**: Details on the general code structure and implementation of different aspects.

A user wishing to install the framework may thus jump directly to Chapter 5, and then successively modify input files in example runs (eg, Chapter 4).

However this will not replace a thorough knowledge of WAVEWATCH III that can be obtained by following Chapters 2 through 5.

The format of a combined user manual and system documentation has been chosen to give users the necessary background to include new physical and numerical approaches in the framework according to their own specifications. This approach became more important as WAVEWATCH III developed into a wave modeling framework. By design, a user can apply his or her numerical and/or physical approaches, and thus develop a new wave model based on the WAVEWATCH III framework. In such an approach, optimization, parallelization, nesting, input and output service programs from the framework can be easily shared between actual models.

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. Note that a best practices guide for code development for WAVEWATCH III is now available (Tolman, 2010a, 2014a). Applications of the modeling framework are widely documented in the literature, some reviews of recent applications may be found in Tolman et al. (2002), Hanson et al. (2009), Chawla et al. (2013), Alves et al. (2014), Rogers et al. (2014), Li and Saulter (2014), Roland and Ardhuin (2014a), Zieger et al. (2018).

The present model version (6.07) is the new public version based on the previous official model release (version 5.16). The following are new features added and code-structure modifications made in WAVEWATCH III 6.07since the previous release.

- Preparing for next model version, adding optional instrumentation to code for profiling of memory use (model version 6.00).
- Separates Stokes drift spectrum calculation (US3D) from OUTG and provides new option to output surface Stokes drift partitioned into run-time defined frequencies (USSP) (model version 6.01).
- Adds a new module for ESMF interface (model version 6.02).
- Adds a capability to update restart file's total energy based on independent significant wave height analysis (model version 6.03).
- Adds domain decomposition for unstructured implicit schemes using PDLIB (Parallel Domain Decomposition Library) and ParMetis (model version 6.04).

- Updates the namelist options for the following programs: ww3_ounf, ww3_ounp, ww3_trnc, ww3_bounc, and ww3_shel (model version 6.05).
- Adding IC5 (the extended FS model) as a sea ice source term option (model version 6.06)
- Public release (model version 6.07)

Other additions include updates on source term parameterizations such IC2, IS2, ST4, REF1.

Up to date information on this model can be found (including bugs and bug fixes) on the WAVEWATCH III GitHub wiki page

https://github.com/NOAA-EMC/WW3/wiki

and at the NCEP WAVEWATCH III page,

http://polar.ncep.noaa.gov/waves/wavewatch/

and comments, questions and suggestions should be directed to the code managers, Ali Abdolali (ali.abdolali@noaa.gov) and Jose-Henrique Alves (henrique.alves@noaa.gov), or the general WAVEWATCH III users mailing group list

ncep.list.wwatch3.users@lstsrv.ncep.noaa.gov

NCEP will redirect questions regarding contributions from outside NCEP to the respective authors of the codes. You may subscribe to the WAVEWATCH III users mailing list at the following web page:

https://www.lstsrv.ncep.noaa.gov/mailman/listinfo/ncep.list.wwatch3.users

1.3 Licensing terms

Starting with model version 3.14, WAVEWATCH III is distributed under the following licensing terms:

start of licensing terms

Software, as understood herein, shall be broadly interpreted as being inclusive of algorithms, source code, object code, data bases and related documentation, all of which shall be furnished free of charge to the Licensee.

Corrections, upgrades or enhancements may be furnished and, if furnished, shall also be furnished to the Licensee without charge. NOAA, however, is not required to develop or furnish such corrections, upgrades or enhancements.

NOAA's software, whether that initially furnished or corrections or upgrades, are furnished as is. NOAA furnishes its software without any warranty whatsoever and is not responsible for any direct, indirect or consequential damages that may be incurred by the Licensee. Warranties of merchantability, fitness for any particular purpose, title, and non-infringement, are specifically negated.

The Licensee is not required to develop any software related to the licensed software. However, in the event that the Licensee does so, the Licensee is required to offer same to NOAA for inclusion under the instant licensing terms with NOAA's licensed software along with documentation regarding its principles, use and its advantages. This includes changes to the wave model proper including numerical and physical approaches to wave modeling, and boundary layer parameterizations embedded in the wave model The Licensee is encouraged but not obligated to provide pre-and post processing tools for model input and output. The software required to be offered shall not include additional models to which the wave model may be coupled, such as oceanic or atmospheric circulation models. The software provided by the Licensee shall be consistent with the latest model version available to the Licensee, and interface routines to the software provided shall conform to programming standards as outlined in the model documentation. The software offered to NOAA shall be offered as is, without any warranties whatsoever and without any liability for damages whatsoever. NOAA shall not be required to include a Licensee's software as part of its software. Licensee's offered software shall not include software developed by others.

A Licensee may reproduce sufficient software to satisfy its needs. All copies shall bear the name of the software with any version number as well as replicas of any applied copyright notice, trademark notice, other notices and credit lines. Additionally, if the copies have been modified, e.g. with deletions or additions, this shall be so stated and identified.

All of Licensee's employees who have a need to use the software may have access to the software but only after reading the instant license and stating,

in writing, that they have read and understood the license and have agreed to its terms. Licensee is responsible for employing reasonable efforts to assure that only those of its employees that should have access to the software, in fact, have access.

The Licensee may use the software for any purpose relating to sea state prediction.

No disclosure of any portion of the software, whether by means of a media or verbally, may be made to any third party by the Licensee or the Licensee's employees

The Licensee is responsible for compliance with any applicable export or import control laws of the United States.

end of licensing terms

The software will be distributed through our web site after the Licensee has agreed to the license terms.

1.4 Copyrights and trademarks

1.5 The WAVEWATCH III® Development Group (WW3DG)

The development of WAVEWATCH III ®relies on the efforts of a team of developers that have worked tirelessly to make this an effective community tool. With the expansion of physical and numerical parameterizations available, the list of contributors to this model keeps growing. The development group consists of a core group of developers that are involved in overall code development, debugging and optimization as well as a larger group that has

either made or continues to make contributions to physics packages and numerics. The following is a list of contributors (both past and present) of this development group (in alphabetic order):

Abdolali, Ali (UCAR at NOAA/NCEP/EMC, USA)
: Unstructured grids, Prallelization efficiency, Wave-Surge coupling, General code development support and code management for WAVEWATCH III.

Accensi, Mickael (LOPS, France)
: NetCDF for input and output (ww3_prnc, ww3_bounc, ww3_ounf, ww3_ounp, ww3_trnc), namelist input files, OASIS coupling, installing and compiling environment, and general code development support.

Alves, Jose-Henrique (SRG at NOAA/NCEP/EMC, USA)
: Shallow water physics packages, development of space-time wave-height extremes approach, General code development support and code management for WAVEWATCH III.

Ardhuin, Fabrice (LOPS, France, previously at SHOM)
: Various parameterization packages (ST3, ST4, BS1, BT4, IG1, REF1, IS2...), interface with unstructured grid schemes, tidal analysis, and some I/O aspects (estimation of fluxes, adaptation of NetCDF).

Babanin, Alexander (University of Melbourne, Australia)
: ST6 project leader, source functions (wind input, whitecapping dissipation, swell dissipation, negative input, physical constraints); sea ice source function IC5.

Barbariol, Francesco (ISMAR-CNR, Italy)
: Development of a space-time wave-height extremes approach.

Benetazzo, Alvise (ISMAR-CNR, Italy)
: Development of a space-time wave-height extremes approach.

Bidlot, Jean (ECMWF, UK)
: Updates to physics package ST3.

Booij, Nico (Delft University of Technology, The Netherlands, retired)
: Original design of source code pre-processor (`w3adc`), basic method

of documentation and other programming habits. Spatially varying wavenumber grid.

Boutin, Guillaume (LOPS, France)
Contribution to IS2 and IC2.

Bunney, Chris (Met Office, UK)
Additional spectral partitioning schemes and netCDF outputs for SMC grid.

Campbell, Tim (Naval Research Laboratory, USA)
Search and regrid utilities, irregular grids, regression testing shell script, ESMF interface module, and overall code development support.

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

Chawla, Arun (NOAA/NCEP/EMC, USA)
Support of code development at NCEP, GRIB packing, automated grid generation software (Chawla and Tolman, 2007, 2008).

Cheng, Sukun (while at Clarkson University, USA)
Original author of the code that was ported into WW3 (for model version 5) as the improved "IC3" parameterization for effect of sea ice on waves.

Collins, Clarence (while an NRL/ASEE post-doc, USA)
Origination of IC4 (sea ice source function).

Filipot, Jean-François (France Energy Marine, formerly at SHOM then LOPS, France).
Unification of whitecapping and breaking in ST4.

Flampouris, Stylianos S. (IMSG at NOAA/NCEP/EMC, USA)
Wave Data Assimilation, communication between DA systems and the model (ww3_uprstr)

Foreman, Mike (IOS, Canada)
Versatile tidal analysis package.

Ginis, Isaac (University of Rhode Island, USA)
Development of source code for sea-state dependent wind stress calculations (FLD1, FLD2).

Hara, Tetsu (University of Rhode Island, USA)
Development of source code for sea-state dependent wind stress calculations (FLD1, FLD2).

Hesser, Tyler J. (US Army Engineer Research and Development Center Coastal and Hydraulics Laboratory)
Unstrucutred grids, Shallow water physics.

Janssen, Peter (ECMWF, United Kingdom)
Original version of WAM-Cycle 4 package (ST3), canonical transform for the second order wave spectrum.

Leckler, Fabien (LOPS, France)
Breaking parameters from source terms and contributions to ST4.

Li, Jian-Guo (UK MetOffice, United Kingdom)
SMC grid, second order UNO schemes and rotated grids.

Lind, Kevin (DoD PETTT, USA)
Improvements to performance of some multi-grid functions.

Liu, Qingxiang (University of Melbourne, Australia)
Sea ice source function IC5; updates to source term package ST6.

Meixner, Jessica (NOAA/NCEP/EMC, USA)
Coupled modeling development, general code development support and code management for WAVEWATCH III.

Mentaschi, Lorenzo (European Commission, Joint Research Centre, JRC, formerly at the University of Genova, Italy)
Development of the Unresolved Obstacles Source Term (UOST).

Orzech, Mark (Naval Research Laboratory, USA)
Source terms for effects of mud (BT8, BT9).

Padilla–Hernández, Roberto (IMSG at NOAA/NCEP/EMC, USA)
Support of code development at NCEP, editing.

Perrie, William (Bedford Institute of Oceanography, Canada)
Two-Scale Approximations for non-linear interactions (NL4).

Rawat, Arshad (MIO, Mauritius and LOPS, France)
Contribution to second order spectrum and free infragravity wave sources (IG1).

Reichl, Brandon (NOAA/GFDL and Princeton University; Formerly at University of Rhode Island, USA)
Development and coding of source code for sea-state dependent wind stress calculations (FLD1, FLD2).

Rogers, W. Erick (Naval Research Laboratory, USA)
Irregular grids, effects of sea ice (e.g. IC1, IC2, IC3, IC4, IC5) and mud (BT8, BT9), ST6 package (source material and advice), adaptation/interfacing of conservative remapping software, tripole grid, regression tests.

Roland, Aron (T. U. Darmstadt, Germany)
Advection on unstructured (triangle-based) grids and meshing tools, Prallelization efficiency, Implementation of Implicit Solver.

Sevigny, Caroline (UQAR, Canada)
Contribution to ice scattering including ice break-up.

Shen, Hayley (Clarkson Univ.)
Supervised contributions by Zhao and Cheng on the "IC3" parameterization for effect of sea ice on waves.

Saulter, Andy (Met Office, UK)
Testing and support for code development at the Met Office.

Sikiric, Mathieu Dutour (IRB, Croatia)
Multi-grid computations with unstructured (triangle-based) grids, Prallelization efficiency, Implementation of Implicit Solver.

Szyszka, Mark (RPS Group, Australia)
Identifying several bugs in the code development process and providing fixes for Openmp issues.

Smith, Jane M. (US Army Engineer Research and Development Center Coastal and Hydraulics Laboratory)
Unstrucutred grids, Shallow water physics.

Tolman, Hendrik L. (DOC/NOAA/NWS/OSTI, USA).
General code architecture, original WAVEWATCH-I, II and III models. Ongoing model development.

Toulany, Bash (Bedford Institute of Oceanography, Canada)
Two-Scale Approximations for non-linear interactions (NL4).

Tracy, Barbara (US Army Corps of Engineers, ERDC-CHL, USA, retired)
Spectral partitioning.

Van Vledder, Gerbrant Ph. (Delft University of Technology, NL)
Webb-Resio-Tracy exact nonlinear interaction routines, as well as some of the original service routines.

Van Der Westhuysen, André (IMSG at NOAA/NCEP/EMC, USA)
Support of code development at NCEP, wave system tracking, addition of triad interactions.

Young, Ian (University of Melbourne, Australia) ST6 source functions (wind input, whitecapping dissipation).

Zhao, Xin (while at Clarkson University, USA)
Original author of the code that was ported into WW3 (model version 4) as the "IC3" parameterization for effect of sea ice on waves.

Zieger, Stefan (Bureau of Meteorology, Australia)
ST6 source term package, code and testing.

1.6 Acknowledgments

The WAVEWATCH III wind wave model was started by Hendrik Tolman with the development of the WAVEWATCH model at Delft University and WAVE- WATCH II at NASA, Goddard Space Flight Center in the early 1990s. Subsequently a research program started under the auspices of the

National Ocean Partnership Program (NOPP) to support developments in wind wave modeling has helped transition WAVEWATCH III from being a task undertaken by a single person or group to a community modeling framework. This program provided the resources to significantly expand the capability of the modeling system as well as move it to a community development paradigm. We are thankful to all our partners in the scientific community who have sustained this effort well past the end of the NOPP program. We are also extremely grateful to the larger user community who have tirelessly worked with us to identify bugs and other issues in the model.

WAVEWATCH III Development Team, February 2019

This page is intentionally left blank.

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 $\mathbf{k}$. Equations given here follow the geometrical optics approximation, which is exact in the limit when scales of variation of depths and currents are much larger than those of an individual wave[1]. Diffraction, scattering and interference effects that are neglected by this approximation can be added a posteriori as source terms in the wave action equation. Under this approximation of slowly varying current and depth, 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 over the scales of individual waves) 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)

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

[1] Even with a factor 5 change in wave height over half a wavelength, the geometrical optics approximation can provide reasonable results as was shown over submarine canyons (Magne et al., 2007)

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

For irregular wind waves, the (random) variance of the sea surface is described using the surface elevation variance density spectrum F . In the wave modeling community this is usually called the 'energy spectrum'. This spectrum F is a function of all independent phase parameters, i.e., $F(\mathbf{k},\sigma,\omega)$, and furthermore varies in space and time at scales larger than those of individual waves, e.g., $F(\mathbf{k},\sigma,\omega;\mathbf{x},t)$. However, for waves shorter than 3 times the dominant wind sea frequency (Leckler et al., 2015), the energy is very close to the linear dispersion relation 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. The effect of non-linear contributions of bound waves can be added in WAVEWATCH III as post-processing, following the method of Janssen (2009).

Within WAVEWATCH III 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 WAVEWATCH III, however, consists of the more traditional frequency-direction spectrum $F(f_r,\theta)$. The different spectra can be calculated from $F(k,\theta)$ using straightforward Jacobian transformations

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

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

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

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

In cases without currents, the variance (energy) of a wave packet 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 without scattering, effects of nonlinear wave propagation (i.e., wave-wave interactions) and partial wave reflections arise in S. Propagation and source terms will be discussed separately in the following sections.

2.2 Propagation

In a numerical model, an Eulerian form of the balance equation (2.7) is needed. This balance equation can either be written in the form of a transport equation (with velocities outside the derivatives), or in a conservation form (with velocities inside the derivatives). The former form is valid for the vector wavenumber spectrum $N(\mathbf{k};\mathbf{x},t)$ only, whereas valid equations of the latter form can be derived for arbitrary spectral formulations, as long as the corresponding Jacobian transformation as described above is well behaved (e.g., Tolman and Booij, 1998). Furthermore, the conservation equation conserves total wave energy/action, unlike the transport equation. This is an important feature of an equation when applied in a numerical model. The balance equation for the spectrum $N(k,\theta;\mathbf{x},t)$ as used in WAVEWATCH III 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 $\mathbf{c}_g = (c_g \sin\theta, c_g \cos\theta$, s is a coordinate in the direction θ and m is a coordinate perpendicular to s. Equation (2.8) is valid for Cartesian coordinates. For large-scale applications, this equation is usually transferred to spherical coordinates, defined by longitude λ and latitude ϕ, but maintaining the definition of the local variance (i.e., per unit surface, as in WAMDIG, 1988)

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

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

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

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

where R is the radius of the earth and U_ϕ and U_λ are current components. Equation (2.15) includes a correction term for propagation along great circles, using a Cartesian definition of θ where $\theta = 0$ corresponds to waves traveling from west to east. WAVEWATCH III can be run using either Cartesian or Spherical coordinates. Note that unresolved obstacles such as islands can be included in the equations. In WAVEWATCH III this is done at the level of the numerical scheme, as is discussed in section 3.4.7. Also, depth variations at the scale of the wavelength can be introduced by a scattering source term described in section 2.3.19.

Finally, both Cartesian and spherical coordinates can be discretized in many ways, using quadrangles (rectangular, curvilinear or SMC grids) and triangles. That aspect is treated in chapter 3.

2.3 Source terms

2.3.1 General concepts

In deep water, the net source term S is generally considered to consist of three parts, an atmosphere-wave interaction term S_{in}, which is usually a positive

energy input but can also be negative in the case of swell, a nonlinear wave-wave interactions term $S_{\rm nl}$ and a wave-ocean interaction term that is generally dominated by wave breaking $S_{\rm ds}$. The input term $S_{\rm in}$ is dominated by the exponential wind-wave growth term, and this source term generally describes this dominant process only. For model initialization, and to provide more realistic initial wave growth, a linear input term $S_{\rm ln}$ can also be added in WAVEWATCH III.

In shallow water additional processes have to be considered, most notably wave-bottom interactions $S_{\rm bot}$ (e.g., Shemdin et al., 1978). In extremely shallow water, an additional breaking term ($S_{\rm db}$) should be considered, if not well represented in $S_{\rm ds}$ (see Filipot and Ardhuin, 2012a). Triad wave-wave interactions ($S_{\rm tr}$) may also be considered, but present parameterizations have limited accuracy. Also available in WAVEWATCH III are source terms for scattering of waves by bottom features ($S_{\rm sc}$), wave-ice interactions ($S_{\rm ice}$), reflection off shorelines or floating objects such as icebergs which can include sources of infragravity wave energy ($S_{\rm ref}$), and a general purpose slot for additional, user defined source terms ($S_{\rm user}$).

This defines the general source terms used in WAVEWATCH III as

$$S = S_{\rm ln} + S_{\rm in} + S_{\rm nl} + S_{\rm ds} + S_{\rm bot} + S_{\rm db} + S_{\rm tr} + S_{\rm sc} + S_{\rm ice} + S_{\rm ref} + S_{\rm user}\,. \quad (2.16)$$

Other source terms could be easily added. Those 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 explicit treatment of the nonlinear interactions defines third-generation wave models. Therefore, the options for the calculation of S_{nl} will be discussed first, starting in section 2.3.2. S_{in} and S_{ds} represent separate processes, but are often interrelated, because the balance of these two source terms governs the integral growth characteristics of the wave energy. Several combinations of these basic source terms are available, and are described in section 2.3.7 and following. The description of linear input starts in section 2.3.12, and section 2.3.13 and following describe available additional processes, mostly related to shallow water and sea ice.

A third-generation wave model effectively integrates the spectrum only up to a cut-off frequency f_{hf} (or wavenumber k_{hf}), that is ideally equal to the highest discretization frequency. In practice the source terms parameterization or the time step used may not allow a proper balance to be obtained,

and thus f_{hf} may be taken within the model frequency range. Above the cut-off frequency a parametric tail is applied (e.g., WAMDIG, 1988)

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

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

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

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

Before actual source term parameterizations are described, the definition of the wind requires some attention. In cases with currents, one can either consider the wind to be defined in a fixed frame of reference, or in a frame of reference moving with the current. Both definitions are available in WAVEWATCH III, 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.

The treatment of partial ice coverage (ice concentration) in the source terms follows the concept of a limited air-sea interface. This means that the momentum transferred from the atmosphere to the waves is limited. Therefore, input and dissipation terms are scaled by the fraction of ice concentration. The nonlinear wave-wave interaction term can be used in areas of open water and ice (Polnikov and Lavrenov, 2007). The scaling is implemented so that it is independent of the source term selected.

2.3.2 S_{nl}: Discrete Interaction Approximation (DIA)

Switch:	`NL1`
Origination:	WAM model
Provided by:	H. L. Tolman

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

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

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

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

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

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

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

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

Recommended (default) values for the constants are $c_1 = 5.5$, $c_2 = 5/6$ and $c_3 = 1.25$ (Hasselmann and Hasselmann, 1985). The overbar notation

	λ_{nl}	C
ST6	0.25	$3.00\ 10^7$
WAM-3	0.25	$2.78\ 10^7$
ST4 (Ardhuin et al.)	0.25	$2.50\ 10^7$
Tolman and Chalikov	0.25	$1.00\ 10^7$

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

denotes straightforward averaging over the spectrum. For an arbitrary parameter z the spectral average is given as

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

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

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

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

where $\hat{k}$ is defined as

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

The shallow water correction of Eq. (2.21) is valid for intermediate depths only. For this reason the mean relative depth $\bar{k}d$ is not allowed to become smaller than 0.5 (as in WAM). All above constants can be reset by the user in the input files of the model (see Section 4.4.3).

2.3.3 S_{nl}: Full Boltzmann Integral (WRT)

Switch:	`NL2`
Origination:	Exact-NL model
Provided by:	G. Ph. van Vledder

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

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

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

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

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

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

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

in which the function T is given by

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

in which

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

The delta functions in Eq. (2.29) determine a region in wavenumber space along which the integration should be carried out. The function θ determines a section of the integral which is not defined due to the assumption that $\mathbf{k}_1$ is closer to $\mathbf{k}_3$ than $\mathbf{k}_2$. The crux of the Webb method consists of using a local coordinate system along a so-named locus, that is, the path in $\mathbf{k}$ space given by the resonance conditions for a given combination of $\mathbf{k}_1$ and $\mathbf{k}_3$. To that end the (k_x,k_y) coordinate system is replaced by a (s,n) coordinate system, where s (n) is the tangential (normal) direction along the locus. After some transformations, the transfer integral can then be written as a closed line integral along the closed locus

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

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

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

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

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

where n_k and n_θ are the discrete number of wavenumbers and directions in the computational grid, respectively. Note that although the spectrum is defined in terms of the vector wavenumber $\mathbf{k}$, the computational grid in a

wave model is more conveniently defined in terms of the absolute wavenumber and wave direction (k, θ) to assure directional isotropy of the calculations. Taking all wavenumbers $\mathbf{k}_1$ into account produces the complete source term due to nonlinear quadruplet wave-wave interactions. Details of the efficient computation of a locus for a given combination of the wavenumbers $\mathbf{k}_1$ and $\mathbf{k}_3$ can be found in Van Vledder (2000, 2002a,b).

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

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

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

An important feature of most algorithms for the evaluation of the Boltzmann integral is that the integration space can be pre-computed. This is also the case for the subroutine version of the WRT method used in WAVEWATCH III. In the initialization phase of the wave model the integration space, consisting of the discretized paths of all loci, together with the interaction coefficients and gradient terms, are computed and stored in a binary data file. For each water depth such a data file is generated and stored in the current directory. The names of these data files consist of a keyword, "quad", followed by the keyword "*xxxx*", with *xxxx* the water depth in meters, or 9999 for deep water. The extension of the binary data file is "bqf" (Bi-

nary Quadruplet File, BQF). If a BQF file exists, the program checks if this BQF file has been generated with the proper spectral grid. If this is not the case, the existing BQF file is overwritten with the correct BQF file. During a wave model run with various depths, the optimal BQF is used, by looking at the nearest water depths for which a valid BQF file has been generated. In addition, the result is rescaled using the ratio of the depth scaling factors (2.21) for the target depth and the depth corresponding to the BQF file.

2.3.4 S_{nl}: Generalized Multiple DIA (GMD)

Switch:	`NL3`
Origination:	WAVEWATCH III
Provided by:	H. L. Tolman

The GMD has been developed as an extension to the DIA. Its development is documented in a set of Technical notes (Tolman, 2003a, 2005, 2008a, 2010b), reports (Tolman and Krasnopolsky, 2004; Tolman, 2009a, 2011a), and papers (Tolman, 2004, 2013a). As part of the development of the GMD, a holistic genetic optimization technique was developed (Tolman and Grumbine, 2013). A package to perform this optimization within WAVEWATCH III was first provided by Tolman (2010c). The most recent version of this package is version 1.5 (Tolman, 2014b).

The GMD expands on the DIA in three ways. First, the definition of the representative quadruplets is expanded. Second, the equations are developed for arbitrary depths, including the description of strong interactions in extremely shallow water (e.g., Webb, 1978). Third, multiple representative quadruplets are used.

The GMD allows for arbitrary configurations of the representative quadruplet, by expanding on the resonance conditions (2.19) as

$$\left.\begin{array}{rcl} \sigma_1 & = & a_1\,\sigma_r \\ \sigma_2 & = & a_2\,\sigma_r \\ \sigma_3 & = & a_3\,\sigma_r \\ \sigma_4 & = & a_4\,\sigma_r \\ \theta_{12} & = & \theta_1 \pm \theta_{12} \end{array}\right\} \quad , \tag{2.34}$$

where $a_1 + a_2 = a_3 + a_4$ to satisfy the general resonance conditions (2.26),

Table 2.2: One, two, or three parameter definitions of the representative quadruplet in the GMD. $\mathbf{k}_d$ or (σ_d, θ_d) represents the discrete spectral grid point for which the discrete interaction contributions are evaluated. All quadruplets are aligned with the discrete directions by taking $\mathbf{k}_1 + \mathbf{k}_2 // \mathbf{k}_d$.

parameters	a_1	a_2	a_3	a_4	θ_{12}	σ_r
(λ)	1	1	$1+\lambda$	$1-\lambda$	0	σ_d
(λ, μ)	$1+\mu$	$1-\mu$	$1+\lambda$	$1-\lambda$	implied*	σ_d
$(\lambda, \mu, \theta_{12})$	$1+\mu$	$1-\mu$	$1+\lambda$	$1-\lambda$	free	$\frac{\sigma_d}{1+\mu}$

* assuming $\mathbf{k}_1 + \mathbf{k}_2 = \mathbf{k}_3 + \mathbf{k}_4 = 2\mathbf{k}_d$

σ_r is a reference frequency, and θ_{12} is the angular gap between the wavenumbers $\mathbf{k}_1$ and $\mathbf{k}_2$. The latter parameter can either be implicit to the quadruplet definition, or can be an explicitly tunable parameter. With this, a one- (λ), two- (λ, μ) or three-parameter $(\lambda, \mu, \theta_{12})$ quadruplet definition have been constructed as outlined in Table 2.2. Note that, unlike in the DIA, all quadruplets are evaluated for the actual water depth and frequency.

In the GMD, the discrete interactions are computed for arbitrary depths. Somewhat surprisingly, interactions computed for the $F(f, \theta)$ spectrum and converted to the native WAVEWATCH III spectrum $N(k, \theta)$ using a Jacobian transformation proved more easily optimizable than computing the interaction contributions for the latter spectrum directly. Furthermore, a two-component scaling function was introduced with a 'deep' scaling function for the traditionally represented weak interactions in intermediate to deep water, and a 'shallow' scaling function representing strong interactions in extremely shallow water. With these modifications, the discrete interac-

tion contributions (2.20) of the DIA become

$$\begin{pmatrix} \delta S_{nl,1} \\ \delta S_{nl,2} \\ \delta S_{nl,3} \\ \delta S_{nl,4} \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \\ 1 \\ 1 \end{pmatrix} \left(\frac{1}{n_{q,d}} C_{\mathrm{deep}} B_{\mathrm{deep}} + \frac{1}{n_{q,s}} C_{\mathrm{shal}} B_{\mathrm{shal}} \right) \times \left[\begin{array}{l} \dfrac{c_{g,1}F_1}{k_1\sigma_1} \dfrac{c_{g,2}F_2}{k_2\sigma_2} \left(\dfrac{c_{g,3}F_3}{k_3\sigma_3} + \dfrac{c_{g,4}F_4}{k_4\sigma_4} \right) \\ - \dfrac{c_{g,3}F_3}{k_3\sigma_3} \dfrac{c_{g,4}F_4}{k_4\sigma_4} \left(\dfrac{c_{g,1}F_1}{k_1\sigma_1} + \dfrac{c_{g,2}F_2}{k_2\sigma_2} \right) \end{array} \right] \quad (2.35)$$

where B_{deep} and B_{shal} are the deep and shallow water scaling functions

$$B_{\mathrm{deep}} = \frac{k^{4+m}\sigma^{13-2m}}{(2\pi)^{11}\, g^{4-m}\, c_g^2} \quad , \quad (2.36)$$

$$B_{\mathrm{shal}} = \frac{g^2\, k^{11}}{(2\pi)^{11}\, c_g} (kd)^n \quad , \quad (2.37)$$

with m and n as tunable parameters, C_{deep} and C_{shal} in Eq. (2.35) are the corresponding deep and shallow water tunable proportionality constants, and $n_{q,d}$ and $n_{q,s}$ are the number of representative quadruplets with deep and shallow water scaling, respectively, representing the feature of the GMD that multiple representative quadruplets can be used.

In the namelists SNL3 and ANL3 the user defines the number of quadruplets, and per quadruplet λ, μ, θ_{12}, C_{deep} and C_{shal}. Values of m and n are defined once, and used for all quadruplets. Finally relative depth below which deep water scaling is not used and above which shallow water scaling is not used are defined. Examples of some of the GMD configurations from Tolman (2010b) are included in the example input file ww3_grid.inp in Section 4.4.3. The default setting is to reproduce the traditional DIA.

Note that the GMD is significantly more complex that the DIA formulation, and requires evaluation of the quadruplet layout for every spectral frequency (compared to a single layout used for the DIA). For effective computation, quadruplet layouts are pre-computed and stored in memory for a set of nondimensional depths. Even with these and other optimizations, the GMD is roughly twice as expensive to compute for a single representative quadruplet than the DIA when using the one-parameter quadruplet layout. Using the two- or three-parameter quadruplet layout, the GMD has four

rather than two quadruplet realizations, making the GMD per quadruplet four times as expensive as the traditional DIA. Using multiple representative quadruplets is linearly additive in computational costs. For more in depth assessment of computational costs of a model including the GMD, see Tolman (2010b) and Tolman (2013a).

2.3.5 S_{nl}: The Two-Scale Approximation (TSA) and the Full Boltzmann Integral (FBI)

Switch:	`NL4 with INDTSA=1 for TSA or 0 for FBI`
Origination:	Full Boltzmann Integral
Provided by:	B. Toulany, W. Perrie, D. Resio & M. Casey

The Boltzmann integral describes the rate of change of action density of a particular wavenumber due to resonant interactions among four wavenumbers. The wavenumbers must satisfy a resonance:

$$\mathbf{k}_1 + \mathbf{k}_2 = \mathbf{k}_3 + \mathbf{k}_4. \tag{2.38}$$

The Two-Scale Approximation (TSA) for calculating the nonlinear interactions that is implemented in WAVEWATCH III is based on papers by Resio and Perrie (2008) (hereafter RP08), Perrie and Resio (2009), Resio et al. (2011) and Perrie et al. (2013a). A description of TSA with respect to the Boltzmann integral is similar to the description for the WRT method. Here, we focus on the TSA derivation and the differences with the WRT method.

Starting from RP08 Eq. (2), the integral of the transfer rate from wavenumber $\mathbf{k}_3$ to wavenumber $\mathbf{k}_1$, denoted $T(\mathbf{k}_1, \mathbf{k}_3)$, satisfies:

$$\frac{\partial n(\mathbf{k}_1)}{\partial t} = \int\int T(\mathbf{k}_1, \mathbf{k}_3) d\mathbf{k}_3 \tag{2.39}$$

which can be re-written (as in RP08) as:

$$\begin{aligned} T(\mathbf{k}_1,\mathbf{k}_3) &= 2\oint [n_1 n_3(n_4-n_2)+n_2 n_4(n_3-n_1)]C(\mathbf{k}_1,\mathbf{k}_2,\mathbf{k}_3,\mathbf{k}_4) \\ &\quad \vartheta(|\mathbf{k}_1-\mathbf{k}_4|-|\mathbf{k}_1-\mathbf{k}_3|)\left|\frac{\partial W}{\partial \eta}\right|^{-1} ds \\ &\equiv 2\oint N^3 C\vartheta \left|\frac{\partial W}{\partial \eta}\right|^{-1} ds \end{aligned} \tag{2.40}$$

as a line integral on contour s and where the function W is given by

$$W=\omega_1+\omega_2-\omega_3-\omega_4 \tag{2.41}$$

where ϑ is the Heaviside function and $\mathbf{k}_2=\mathbf{k}_2(s,\mathbf{k}_1,\mathbf{k}_3)$. Here, n_i is the action density at $\mathbf{k}_i$ and function W is given by $W=\omega_1+\omega_2-\omega_3-\omega_4$ requiring that the interactions conserve energy on s, which is the locus of points satisfying $W=0$ and η is the local orthogonal to the locus s. Note that Eq.(2.40) is similar to Eq. (2.31) of WRT in section 2.3.3 with coupling coefficient C equal to the WRT coupling coefficient G divided by 2.

TSA and FBI For FBI, as well as for WRT, we numerically compute the discretized form of Eq.(2.40) as a finite sum of many line integrals (around locus s) of $T(\mathbf{k}_1,\mathbf{k}_3)$ for all discrete combinations of $\mathbf{k}_1$ and $\mathbf{k}_3$. The line integral is determined by dividing the locus into a finite number of segments, each with the length ds. A complete 'exact' computation is expensive, requiring 10^3-10^4 times DIAs run time.

The methodology for TSA is to decompose a directional spectrum into a *parametric* (*broadscale*) spectrum and a (*local-scale*) nonparametric *residual* component. The residual component allows the decomposition to retain the same number of degrees of freedom as the original spectrum, a prerequisite for the nonlinear transfer source term in 3G models. As explained in the cited literature, this decomposition leads to a representation of the nonlinear wave-wave interactions in terms of the *broadscale* interactions, *local-scale* interactions, and the *cross terms*: the interactions between the broadscale and local-scale components of the spectrum. This method allows the broadscale interactions and certain portions of the local-scale interactions to be *pre-computed*. TSA's accuracy is dependent on the accuracy of the parameterization used to represent the broadscale component.

We begin by decomposing a given action density spectrum n_i into the *parametric* broadscale term $\hat{n}_i$ and a *residual* local-scale (or '*perturbation*-scale') term n'_i. The broadscale term $\hat{n}_i$ is assumed to have a JONSWAP-type form, depending on only a few parameters,

$$n'_i = n_i - \hat{n}_i \tag{2.42}$$

TSA's accuracy depends on $\hat{n}_i$, in that if the number of degrees of freedom used for $\hat{n}_i$ approaches the number of degrees of freedom in a given wave spectrum n_i, the local-scale n'_i becomes quite small, and thus, TSA is very accurate. However, it is time-consuming to set up large multi-dimensional sets of pre-computed matrices for $\hat{n}_i$. Therefore an optimal TSA formulation must minimize the number of parameters needed for $\hat{n}_i$. However, even for complicated multi-peaked spectra n_i, a small set of parameters can be used to let $\hat{n}_i$ capture *most* of the spectra so that the residual n'_i, can be small (RP08; Perrie and Resio (2009)).

RP08 describe the partitioning of n_i so that the transfer integral T in Eq. (2.40) consists of the sum of *broadscale* terms $\hat{n}_i$, denoted B, *local-scale* terms n'_i, denoted L, and *cross-scale terms* of $\hat{n}_i$ and n'_i, denoted X. Thus the nonlinear transfer term can be represented as,

$$S_{nl}(f,\theta) = B + L + X \tag{2.43}$$

where B depends on JONSWAP-type parameters x_i and can be pre-computed,

$$S_{nl}(f,\theta)_{broadscale} = B(f,\theta,x_1,\ldots,x_n). \tag{2.44}$$

TSA needs to find accurate efficient approximations for $L + X$. If all terms in Eq. (2.43) are computed as in FBI, this might result in an $8\times$ increase in the computations, compared to B in Eq. (2.43). While this approach can provide a means to examine the general problem of bimodal wave spectra, for example in mixed seas and swells, by subtracting the interactions for a single spectral region from the interactions for the sum of the two spectral regions, it does not provide the same insight as the use of the *split* density function, where the cross-interaction terms can be examined algebraically.

In any case, to simplify Eq. (2.43), terms involving n'_2 and n'_4 are neglected assuming that these local-scale terms are deviations about the broad-scale terms, $\hat{n}_2$ and $\hat{n}_4$, which are supposed to capture most of the spectra, whereas n'_2 and n'_4, with their positive/negative differences and products tend

to cancel. TSA's ability to match the FBI (or WRT) results for test spectra is used to justify the approach. Moreover, the broadscale terms $\hat{n}_2$ and $\hat{n}_4$, tend to have much longer lengths along locus s and therefore should contribute more to the net transfer integral. Thus, RP08 show that

$$S_{nl}(k_1) = B + L + X = B + \int\int\oint N_*^3 C \left|\frac{\partial W}{\partial n}\right|^{-1} ds k_3 d\theta_3 dk_3, \qquad (2.45)$$

where N_*^3 is what's left from all the cross terms, after neglecting terms involving n_2' and n_4',

$$N_*^3 = \hat{n}_2\hat{n}_4(n_3' - n_1') + n_1'n_3'(\hat{n}_4 - \hat{n}_2) + \hat{n}_1 n_3'(\hat{n}_4 - \hat{n}_2) + n_1'\hat{n}_3(\hat{n}_4 - \hat{n}_2), . \quad (2.46)$$

and they use known scaling relations, with specific parameterizations, for example for f^{-4} or f^{-5} based spectra. To implement this formulation, we generally fit each peak separately.

It should be noted that to speed up the computation, a pre-computed set of multi-dimensional arrays, for example the grid geometry arrays and the gradient array, which are functions of spectral parameters, number of segments on the locus and depth, are generated and saved in a file with filename '**grd_dfrq_nrng_nang_npts_ndep.dat**', for example, 'grd_1.1025-_35_36_30_37.dat', etc.

The flow chart for TSA's main subroutine W3SNL4 in w3snl4md.ftn is as follows:

```
          /
         |
         |*** It's called from:
         |    -----------------
         |    (1) W3SRCE in w3srcemd.ftn; to calc. & integrate source
         |        term at single pt
         |    (2) GXEXPO in gx_outp.ftn;  to perform point output
         |    (3) W3EXPO in ww3_outp.ftn; to perform point output
         |    (4) W3EXNC in ww3_ounp.ftn; to perform point output
         |*** It can also be called from:
         |    (5) W3IOGR in w3iogrmd.ftn; to perform I/O of "mod_def.ww3"
         |
W3SNL4 -->|
         |
         |*** It calls:
         |    ---------
         |               /
```

```
|              |
|              |*** It's called from:
|              |    -----------------
|              |    W3SNL4 in w3snl4md.ftn; main TSA subr.
|              |*** It can also be called from: subr W3IOGR
|              |    W3IOGR in w3iogrmd.ftn; I/O of mod_def.ww3
|              |
|              |*** It calls:
|              |    ---------
|              |--> wkfnc (function)
|              |--> cgfnc (function)
|(1)           |
|--> INSNL4 -->|                /
|              |               |--> shloxr (uses func wkfnc)
|              |--> gridsetr -->|--> shlocr
|              |               |--> cplshr
|              |                \
|(2)            \
|--> optsa2
|
|                /
|(3)             |    if (ialt=2)
|--> snlr_tsa -->|--> interp2
|                |
|                 \
|
|
|                /
|(4)             |    if (ialt=2)
|--> snlr_fbi -->|--> interp2
|                |
|                 \
|
 \
```

2.3.6 S_{nl}: Nonlinear Filter

Switch:	`NLS`
Origination:	WAVEWATCH III
Provided by:	H. L. Tolman

When the DIA of Eqs. (2.19) and (2.20) is applied with a quadruplet where

λ_{nl} is small enough so that the resulting quadruplet is *not* resolved by the discrete spectral grid, then the resulting numerical form of the DIA corresponds to a simple diffusion tensor. If this tensor is filtered so that it is applied to the high-frequency tail of the spectrum only, then a conservative filter results, which retains all conservation properties of the nonlinear interactions (Tolman, 2008a, 2011b). This filter can be used as a part of a parameterization of nonlinear interactions. For instance, it was shown to be effective in removing high-frequency spectral noise in some GMD configurations in Figs. 5 and 6 of Tolman (2011b). Since it is essential that the quadruplet is not resolved by the spectral grid, the free parameter of the filter defining the quadruplet is the relative offset of quadruplets 3 and 4 in the discrete frequency grid (α_{34}, $0 < \alpha_{34} < 1$), from which λ_{nl} is computed as

$$\lambda_{nl} = \alpha_{34}(X_\sigma - 1), \tag{2.47}$$

where X_σ is the increment factor for the discrete frequency grid, typically $X_\sigma = 1.1$ [Eq. (3.1)]. Using the native spectral description of WAVEWATCH III, the change in spectral density δN_i at quadruplet component i, is written in the form of a discrete diffusion equation as (Tolman, 2011b, page 294)

$$\begin{pmatrix} \delta N_3 \\ \delta N_1 \\ \delta N_4 \end{pmatrix} = N_1 \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + N_1 \frac{S\Delta t}{N_1} \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix}, \tag{2.48}$$

with

$$S = \frac{C_{nlf}\, k^4 \sigma^{12}}{(2\pi)^9\, g^4\, c_g} \left[\frac{N_1^2}{k_1^2} \left(\frac{N_3}{k_3} + \frac{N_4}{k_4} \right) - 2 \frac{N_1}{k_1} \frac{N_3}{k_3} \frac{N_4}{k_4} \right], \tag{2.49}$$

where C_{nlf} is the proportionality constant of the DIA used in the filter. The DIA results in changes S for two mirror-image quadruplets (S_a and S_b). A JONSWAP style filter (Φ) is applied to localize the smoother at higher frequencies only, with

$$\Phi(f) = \exp \left[-c_1 \left(\frac{f}{c_2 f_p} \right)^{-c_3} \right], \tag{2.50}$$

where c_1 through c_3 are tunable parameters. The latter three parameters need to be chosen such that $\Phi(f_p) \approx 0$, $\Phi(f > 3f_p) \approx 1$ and that $\Phi \approx 0.5$ for

frequencies moderately larger than f_p. This can be achieved by setting

$$c_1 = 1.25, \quad c_2 = 1.50, \quad c_3 = 6.00. \tag{2.51}$$

Accounting for the redistribution of the changes $S_{a,b}$ over the neighboring discrete spectral grids points, the effective nondimensional strengths ($\tilde{S}_{a,b}$) of the interactions for both quadruplets become

$$\tilde{S}_a = \Phi(f) M_1 S_a \Delta t / N_1, \quad \tilde{S}_b = \Phi(f) M_1 S_b \Delta t / N_1, \tag{2.52}$$

where N_1 is the action density at the center component of the quadruplet, and M_1 is a factor accounting for the redistribution of the contribution over the discrete spectral grid (for details, see Tolman, 2011b). To convert this DIA into a stable diffusive filter, $|\tilde{S}_{a,b}|$ should be limited to $\tilde{S}_{\max} \approx 0.5$ (e.g., Fletcher, 1988). The maximum change is distributed over the two quadruplets using

$$\tilde{S}_{m,a} = \frac{|\tilde{S}_a| \tilde{S}_{\max}}{|\tilde{S}_a| + |\tilde{S}_b|}, \quad \tilde{S}_{m,b} = \frac{|\tilde{S}_b| \tilde{S}_{\max}}{|\tilde{S}_a| + |\tilde{S}_b|}, \tag{2.53}$$

and the normalized changes $\tilde{S}_a$ and $\tilde{S}_b$ are limited as

$$-\tilde{S}_{m,a} \le \tilde{S}_a \le \tilde{S}_{m,a}, \quad -\tilde{S}_{m,b} \le \tilde{S}_b \le \tilde{S}_{m,b}. \tag{2.54}$$

With this, the free parameters of the conservative nonlinear filter are α_{34} in Eq. (2.47), C_{nlf} in Eq. (2.49), $\tilde{S}_{\max}$ in Eq. (2.53), and c_1 through c_3 in Eq. (2.50). All these parameters can de adjusted by the user through the namelist SNLS in ww3_grid.inp (parameters A34 , FHFC, DNM, FC1, FC2 and FC3, respectively). Note that this filter is applied in addition to a parameterization of S_{nl}, but does not replace it. Hence, it is used on concert with a full parameterization of S_{nl}, described in the preceding sections.

2.3.7 $S_{in} + S_{ds}$: WAM cycle 3

Switch:	`ST1`
Origination:	WAM model
Provided by:	H. L. Tolman

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

$$\mathcal{S}_{in}(k,\theta) = C_{in}\frac{\rho_a}{\rho_w}\max\left[0,\left(\frac{28\,u_*}{c}\cos(\theta-\theta_w)-1\right)\right]\,\sigma\,N(k,\theta)\;, \tag{2.55}$$

$$u_* = u_{10}\sqrt{(0.8+0.065u_{10})10^{-3}}\,, \tag{2.56}$$

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

$$\mathcal{S}_{ds}(k,\theta) = C_{ds}\,\hat{\sigma}\,\frac{k}{\hat{k}}\left(\frac{\hat{\alpha}}{\hat{\alpha}_{PM}}\right)^2 N(k,\theta)\;, \tag{2.57}$$

$$\hat{\sigma} = \left(\overline{\sigma^{-1}}\right)^{-1}\;, \tag{2.58}$$

$$\hat{\alpha} = E\,\hat{k}^2\,g^{-2}\,, \tag{2.59}$$

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[2] $m=4.5$ and by

$$f_{hf} = \max\left[\,2.5\,\hat{f}_r\;,\;4\,f_{PM}\,\right]\;, \tag{2.60}$$

$$f_{PM} = \frac{g}{28\,u_*}\,, \tag{2.61}$$

where f_{PM} is the Pierson and Moskowitz (1964) frequency, estimated from the wind friction velocity u_*. The shape and attachment point of this tail is hardcoded to the present model. The tunable parameters C_{in}, C_{ds} and α_{PM} are preset to their default values, but can be redefined by the user in the input files of the model. Alternative f_{PM} and α_{PM} values proposed by Alves et al. (2003) may also be used.

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

2.3.8 $S_{in} + S_{ds}$: Tolman and Chalikov 1996

Switch:	`ST2`
Origination:	WAVEWATCH III
Provided by:	H. L. Tolman

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

$$\mathcal{S}_{in}(k,\theta) = \sigma\,\beta\,N(k,\theta)\,, \tag{2.62}$$

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

$$10^4\beta = \begin{cases} -a_1\tilde{\sigma}_a^2 - a_2 & , \quad & \tilde{\sigma}_a & < -1 \\ a_3\tilde{\sigma}_a(a_4\tilde{\sigma}_a - a_5) - a_6 & , \quad -1 \le & \tilde{\sigma}_a & < \Omega_1/2 \\ (a_4\tilde{\sigma}_a - a_5)\tilde{\sigma}_a & , \quad \Omega_1/2 \le & \tilde{\sigma}_a & < \Omega_1 \\ a_7\tilde{\sigma}_a - a_8 & , \quad \Omega_1 \le & \tilde{\sigma}_a & < \Omega_2 \\ a_9(\tilde{\sigma}_a - 1)^2 + a_{10} & , \quad \Omega_2 \le & \tilde{\sigma}_a & \end{cases} \tag{2.63}$$

where

$$\tilde{\sigma}_a = \frac{\sigma\,u_\lambda}{g}\cos(\theta - \theta_w) \tag{2.64}$$

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

$$\lambda_a = \frac{2\pi}{k|\cos(\theta - \theta_w)|}\,. \tag{2.65}$$

The parameters $a_1 - a_{10}$ and Ω_1, Ω_2 in Eq. (2.63) depend on the drag coefficient C_λ at the height $z = \lambda_a$:

$$
\begin{array}{ll}
\Omega_1 = 1.075 + 75C_\lambda & \Omega_2 = \ 1.2 + 300C_\lambda \\
a_1 = 0.25 + 395C_\lambda, & a_3 = \ (a_0 - a_2 - a_1)/(a_0 + a_4 + a_5) \\
a_2 = 0.35 + 150C_\lambda, & a_5 = \ a_4\Omega_1 \\
a_4 = 0.30 + 300C_\lambda, & a_6 = \ a_0(1 - a_3) \\
a_9 = 0.35 + 240C_\lambda, & a_7 = \ (a_9(\Omega_2 - 1)^2 + a_{10})/(\Omega_2 - \Omega_1) \\
a_{10} = -0.05 + 470C_\lambda, & a_8 = \ a_7\Omega_1 \\
& a_0 = \ 0.25a_5^2/a_4
\end{array}
\tag{2.66}
$$

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

$$u_z = \frac{v_*}{\kappa} \ln\left(\frac{z}{z_0}\right) \ , \tag{2.67}$$

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)

$$C_r = \kappa^2 \left[R - \ln(C)\right]^2 \ , \tag{2.68}$$

where

$$R = \ln\left(\frac{z_r g}{\chi\sqrt{\alpha}u_r^2}\right) \ , \tag{2.69}$$

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

$$C_r = 10^{-3}\left(0.021 + \frac{10.4}{R^{1.23} + 1.85}\right) \ . \tag{2.70}$$

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)

$$\alpha = 0.57\left(\frac{u_*}{c_p}\right)^{3/2} \ . \tag{2.71}$$

As the latter equation depends on the drag coefficient, Eqs. (2.69) through (2.71) 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. (2.71) 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. (2.67) the roughness parameter z_0 becomes

$$z_0 = z_r \, \exp\left(-\kappa C_r^{-1/2}\right) \; , \tag{2.72}$$

and the wind velocity and drag coefficient at height λ become

$$u_\lambda = u_r \frac{ln(\lambda_a/z_0)}{ln(z_r/z_0)}, \tag{2.73}$$

$$C_\lambda = C_r \left(\frac{u_a}{u_\lambda}\right)^2 , \tag{2.74}$$

Finally, Eq. (2.71) 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)

$$f_{p,i} = \frac{\int\int \, f^{-2} \, c_g^{-1} \, \max\left[\, 0 \, , \, \mathcal{S}_{wind}(k,\theta) \,\right] df \, d\theta}{\int\int \, f^{-3} \, c_g^{-1} \, \max\left[\, 0 \, , \, \mathcal{S}_{wind}(k,\theta) \,\right] df \, d\theta} \; , \tag{2.75}$$

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

$$\tilde{f}_p \; = \; 3.6 \, 10^{-4} \; + \; 0.92 \tilde{f}_{p,i} \; - \; 6.3 \, 10^{-10} \; \tilde{f}_{p,i}^{-3} \; . \tag{2.76}$$

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

$$\mathcal{S}_{i,m} = \begin{cases} \mathcal{S}_i & \text{for } \beta \geq 0 \quad \text{or} \quad f > 0.8 f_p \\ X_s \mathcal{S}_i & \text{for } \beta < 0 \quad \text{and} \quad f < 0.6 f_p \\ \mathcal{X}_s \mathcal{S}_i & \text{for } \beta < 0 \quad \text{and} \quad 0.6 f_p < f < 0.8 f_p \end{cases} \quad , \tag{2.77}$$

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 (2.62), 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 drag coefficient that follows from Eq. (2.71) becomes unrealistically high for hurricane strength wind speeds, leading to unrealistically high wave growth rates. To alleviate this, the drag coefficient at the reference height C_r can be capped with a maximum allowed drag coefficient $C_{r,max}$, either as a simple hard limit

$$C_r = \min(C_r, C_{r,max}) \quad , \tag{2.78}$$

or with a smooth transition

$$C_r = C_{r,max} \tanh(C_r / C_{r,max}) \quad . \tag{2.79}$$

Selection of the capped drag coefficient occurs at the compile stage of the code. The cap level and cap type can be set by the user. Defaults settings are $C_{r,max} = 2.5\ 10^{-3}$ and Eq. (2.78).

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,

$$\mathcal{S}_{ds,l}(k,\theta) = -2\ u_*\ h\ k^2 \phi\ N(k,\theta)\ , \tag{2.80}$$

$$h = 4 \left(\int_0^{2\pi} \int_{f_h}^{\infty} F(f,\theta)\ df\ d\theta \right)^{1/2} \quad . \tag{2.81}$$

$$\phi = b_0 + b_1 \tilde{f}_{p,i} + b_2 \tilde{f}_{p,i}^{-b_3}\ . \tag{2.82}$$

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. (2.82) 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

$$b_2 = \tilde{f}_{p,i,\min}^{b_3} \left(\phi_{\min} - b_0 - b_1 \tilde{f}_{p,i,\min} \right) , \tag{2.83}$$

$$b_3 = 8 . \tag{2.84}$$

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

$$\tilde{f}_a = \frac{\phi_{\min} - b_0}{b_1} \quad , \quad \tilde{f}_b = \max \left\{ \tilde{f}_a - 0.0025 \, , \, \tilde{f}_{p,i,\min} \right\} \quad , \tag{2.85}$$

$$b_2 = \tilde{f}_b^{b_3} \left[\phi_{\min} - b_0 - b_1 \tilde{f}_b \right] , \tag{2.86}$$

$$b_3 = \frac{b_1 \tilde{f}_b}{\phi_{\min} - b_0 - b_1 \tilde{f}_b} . \tag{2.87}$$

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

$$\mathcal{S}_{ds,h}(k,\theta) = -a_0 \left(\frac{u_*}{g} \right)^2 f^3 \, \alpha_n^B \, N(k,\theta) \quad , \tag{2.88}$$

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

$$\alpha_n = \frac{\sigma^6}{c_g \, g^2 \alpha_r} \int_0^{2\pi} N(k,\theta) \, d\theta \quad , \tag{2.89}$$

where α_n is Phillips' nondimensional high-frequency energy level normalized with α_r, and where a_0 through a_2 and α_r are empirical constants. This parameterization implies that $m = 5$ in the parametric tail, which has been preset in the model. Note that in the model Eq. (2.89) is solved assuming a deep water dispersion relation, in which case α_n is evaluated as

$$\alpha_n = \frac{2 \, k^3}{\alpha_r} F(k) \quad . \tag{2.90}$$

The two constituents of the dissipation source term are combined using a simple linear combination, defined by the frequencies f_1 and f_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 2.3: Suggested constants in the source term package of Tolman and Chalikov. KC denotes Kahma and Calkoen (1992, 1994). First line represents default model settings.

$$\mathcal{S}_{ds}(k,\theta) = \mathcal{A}\mathcal{S}_{ds,l} + (1-\mathcal{A})\,\mathcal{S}_{ds,h}\;, \tag{2.91}$$

$$\mathcal{A} = \begin{cases} 1 & \text{for} \quad f < f_l \quad , \\ \frac{f-f_2}{f_1-f_2} & \text{for} \quad f_1 \le f < f_2 \quad , \\ 0 & \text{for} \quad f_2 \le f \quad . \end{cases} \tag{2.92}$$

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)

$$N(k_i,\theta) = (1-\mathcal{B})\,N(k_i,\theta) + \mathcal{B}N(k_{i-1},\theta)\left(\frac{f_i}{f_{i-1}}\right)^{-m-2}\;, \tag{2.93}$$

where i is a discrete wavenumber counter, and $\mathcal{B}$ 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

$$\left.\begin{array}{rcl} f_{hf} & = & 3.00\,f_{p,i} \\ f_1 & = & 1.75\,f_{p,i} \\ f_2 & = & 2.50\,f_{p,i} \\ f_h & = & 2.00\,f_{p,i} \end{array}\right\}\;. \tag{2.94}$$

Furthermore, $f_{p,i,\min} = 0.009$ and $\alpha_r = 0.002$ are preset in the model. All other tunable parameters have to be provided by the user. Suggested and default values are given in Table 2.3.

Test results of these source terms in a global model implementation (Tolman, 2002f) suggested that (i) the model tuned in the classical way to

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

$$\frac{u_e}{u} = \left(\frac{c_o}{1 + C_1 + C_2} \right)^{1/2} , \tag{2.95}$$

$$C_1 = c_1 \tanh \left[\max(0, f_1 \{ \mathcal{ST} - \mathcal{ST}_o \}) \right] , \tag{2.96}$$

$$C_2 = c_2 \tanh \left[\max(0, f_2 \{ \mathcal{ST} - \mathcal{ST}_o \}) \right] , \tag{2.97}$$

$$\mathcal{ST} = \frac{hg}{u_h^2} \frac{T_a - T_s}{T_0} , \tag{2.98}$$

where $\mathcal{ST}$ is a bulk stability parameter, and T_a, T_s and T_0 are the air, sea and reference temperature, respectively. Furthermore, $f_1 \leq 0$, c_1 and c_2 have opposite signs and $f_2 = f_1 c_1 / c_2$. Following Tolman (2002f), default settings of $c_0 = 1.4$, $c_1 = -0.1$, $c_2 = 0.1$, $f_1 = -150$ and $\mathcal{ST}_o = -0.01$ in combination with the tuning to stable stratification wave growth data ('KC stable' parameter values in Table 2.3) are used. Note that this effective wind speed was derived for winds at 10 m height. The wind correction can be switched off by the user during compilation of the model, and default parameter settings can be redefined by the user in the program input files. This stability correction is activated with the `STAB2` switch. The air-sea temperature differences need to be provided by the user, e.g. using ww3_prep.

2.3.9 $S_{in} + S_{ds}$: WAM cycle 4 (ECWAM)

Switch:	`ST3`
Origination:	WAM model
Provided by:	F. Ardhuin

The wind-wave interaction source terms described here are based on the wave growth theory of Miles (1957), modified by Janssen (1982). The pressure-slope correlations that give rise to part of the wave generation are parameterized following Janssen (1991).

This parameterization was further extended by Abdalla and Bidlot (2002) to take into account a stronger gustiness in unstable atmospheric conditions. This effect is included in the present parameterization and is activated with the optional `STAB3` switch. The formula used in `STAB3` is not described herein. For that, the reader is referred to Abdalla and Bidlot (2002). If `STAB3` is used, the air-sea temperature differences should be provided by the user, e.g. using ww3_prep.

Efforts have been made to make the present implementation as close as possible to the one in the ECWAM model (Bidlot et al., 2005), in particular the stress lookup tables were verified to be identical. Later modifications include the addition of a negative part in the wind input to represent swell dissipation.

The source term reads (Janssen, 2004)

$$\mathcal{S}_{in}(k,\theta) = \frac{\rho_a}{\rho_w}\frac{\beta_{\max}}{\kappa^2}\mathrm{e}^{Z}Z^4\left(\frac{u_\star}{C}+z_\alpha\right)^2\cos^{p_{in}}(\theta-\theta_u)\sigma N\left(k,\theta\right)+S_{out}(k,\theta), \tag{2.99}$$

where ρ_a and ρ_w are the air and water densities, $\beta_{\max}$ is a non-dimensional growth parameter (constant), κ is von Kármán' constant, and p_{in} is a constant that controls the directional distribution of $\mathcal{S}_{in}$. In the present implementation the air/water density ratio ρ_a/ρ_w is constant. We define $Z = \log(\mu)$ where μ is given by Janssen (1991) Eq. (16), and corrected for intermediate water depths, so that

$$Z = \log(kz_1) + \kappa/\left[\cos\left(\theta-\theta_u\right)\left(u_\star/C+z_\alpha\right)\right], \tag{2.100}$$

where z_1 is a roughness length modified by the wave-supported stress τ_w, and z_α is a wave age tuning parameter[3]. The roughness z_1 is defined as,

[3]Although this tuning parameter z_α is not well described in WAM-Cycle4 documentation, it has an important effect on wave growth. Essentially it shifts the wave age of the long waves, which typically increases the growth, and even generates waves that travel faster than the wind. This accounts for some gustiness in the wind and should possibly be resolution-dependent. For reference, this parameter was not properly set in early versions of the SWAN model, as discovered by R. Lalbeharry.

$$U_{10} = \frac{u_\star}{\kappa} \log\left(\frac{z_u}{z_1}\right) \tag{2.101}$$

$$z_1 = \alpha_0 \frac{\tau}{\sqrt{1-\tau_w/\tau}}, \tag{2.102}$$

where $\tau = u_\star^2$, and z_u is the height at which the wind is specified. These two equations provide an implicit functional dependence of $u_\star$ on U_{10} and τ_w/τ. This relationship is then tabulated (Janssen, 1991; Bidlot et al., 2007).

An important part of the parameterization is the calculation of the wave-supported stress τ_w,

$$\tau_w = \left| \int_0^{k_{\max}} \int_0^{2\pi} \frac{\mathcal{S}_{in}(k',\theta)}{C} \left(\cos\theta, \sin\theta\right) \mathrm{d}k' \mathrm{d}\theta + \tau_{\mathrm{hf}}(u_\star,\alpha) \left(\cos\theta_u, \sin\theta_u\right) \right|, \tag{2.103}$$

which includes the resolved part of the spectrum, up to $k_{\max}$, as well as the stress supported by shorter waves, τ_{hf}. Assuming a f^{-X} diagnostic tail beyond the highest frequency, τ_{hf} is given by

$$\begin{aligned} \tau_{\mathrm{hf}}(u_\star,\alpha) &= \frac{u_\star^2}{g^2} \frac{\sigma_{\max}^X 2\pi\sigma}{2\pi C_g(k_{\max})} \int_0^{2\pi} N\left(k_{\max},\theta\right) \max\left\{0, \cos\left(\theta-\theta_u\right)\right\}^3 d\theta \\ &\quad \times \frac{\beta_{\max}}{\kappa^2} \int_{\sigma_{\max}}^{0.05*g/u_\star} \frac{\mathrm{e}^{Z_{\mathrm{hf}}} Z_{\mathrm{hf}}^4}{\sigma^{X-4}} \mathrm{d}\sigma \end{aligned} \tag{2.104}$$

where the second integral is a function of $u_\star$ and the Charnock coefficient α only, which is easily tabulated. In practice the calculation is coded with $X = 5$, and the variable Z_{hf} is defined by,

$$Z_{\mathrm{hf}}(\sigma) = \log(kz_1) + \min\left\{\kappa/\left(u_\star/C + z_\alpha\right), 20\right\}. \tag{2.105}$$

This parameterization is sensitive to the spectral level at $k_{\max}$. A higher spectral level will lead to a larger value of $u_\star$ and thus positive feedback on the wind input via z_1. This sensitivity is exacerbated by the sensitivity of the high-frequency spectral level to the presence of swell via the dissipation term.

A linear damping of swells was introduced in the operational ECWAM model in September 2009. It takes the form given by Janssen (2004)

Par.	WWATCH var.	namelist	WAM4	BJA	Bidlot 2012
z_u	ZWND	SIN3	10.0	10.0	10.0
α_0	ALPHA0	SIN3	0.01	0.0095	0.0095
$\beta_{\max}$	BETAMAX	SIN3	1.2	1.2	1.2
p_{in}	SINTHP	SIN3	2	2	2
z_α	ZALP	SIN3	0.0110	0.0110	0.0080
s_1	SWELLF	SIN3	0.0	0.0	1.0

Table 2.4: Parameter values for WAM4, BJA and the 2012 update in the ECWAM model. Source term parameterizations that can be reset via the SIN3 and SDS3 namelist. BJA is generally better than WAM4. The default parameters in ST3 corresponds to BJA. Please note that the names of the variables only apply to the namelists. In the source term module the names are slightly different, with a doubled first letter, in order to differentiate the variables from the pointers to these variables.

$$S_{out}(k,\theta) = 2 s_1 \kappa \frac{\rho_a}{\rho_w} \left(\frac{u_\star}{C}\right)^2 \left[\cos(\theta - \theta_u) - \frac{\kappa C}{u_\star \log(k z_0)}\right] \tag{2.106}$$

where s_1 is set to 1 when this damping is used and 0 otherwise. For $s_1 = 0$ the parameterization is the WAM4 or BJA parameterization (see Table 2.4).

Due to the increase in high-frequency input compared to WAM3, the dissipation function was adapted by Janssen (1994) from the WAM3 dissipation, and later reshaped by Bidlot et al. (2005). That later modification is referred to as "BJA" for Bidlot, Janssen and Abdallah. A more recent modification, strongly improved the model results for Pacific swells, at the price of an underestimation of the highest sea states. This corresponds to the ECMWF WAM model contained in the IFS version CY38R1 (Bidlot, 2012). Note that these parameters were optimized for use of neutral winds from the operational ECMWF analysis. Using these with other wind products may require a re-tuning of these coefficients. For example, with NCEP or CFSRR winds, the value of BETAMAX should probably be reduced or ZWND increased.

The generic form of the WAM4 dissipation term is,

Par.	WWATCH var.	namelist	WAM4	BJA	Bidlot 2012
$C_{\rm ds}$	SDSC1	SDS3	-4.5	-2.1	-1.33
p	WNMEANP	SDS3	-0.5	0.5	0.5
$p_{\rm tail}$	WNMEANPTAIL	SDS3	-0.5	0.5	0.5
δ_1	SDSDELTA1	SDS3	0.5	0.4	0.5
δ_2	SDSDELTA2	SDS3	0.5	0.6	0.5

Table 2.5: Parameter values for WAM4, BJA and the update by Bidlot (2012). Source term parameterizations that can be reset via the SDS3 namelist. BJA is generally better than WAM4. Please note that the names of the variables only apply to the namelists. In the source term module the names are slightly different, with a doubled first letter, in order to differentiate the variables from the pointers to these variables.

$$S_{ds}\left(k,\theta\right)^{\rm WAM} = C_{ds}\overline{\alpha}^2\overline{\sigma}\left[\delta_1\frac{k}{\overline{k}} + \delta_2\left(\frac{k}{\overline{k}}\right)^2\right]N\left(k,\theta\right) \tag{2.107}$$

where C_{ds} is a non-dimensional constant δ_1 and δ_2 are weight parameters,

$$\overline{k} = \left[\frac{\int k^p N\left(k,\theta\right)\mathrm{d}\theta}{\int N\left(k,\theta\right)\mathrm{d}\theta}\right]^{1/p} \tag{2.108}$$

with p a constant power. Similarly, the mean frequency is defined as

$$\overline{\sigma} = \left[\frac{\int \sigma^p N\left(k,\theta\right)\mathrm{d}\theta}{\int N\left(k,\theta\right)\mathrm{d}\theta}\right]^{1/p}, \tag{2.109}$$

so that the mean steepness is $\overline{\alpha} = E\overline{k}^2$.

The mean frequency also occurs in the definition of the maximum frequency of prognostic integration of the source terms. Since the definition of that frequency may be different from that of the source term it is defined with another exponent $p_{\rm tail}$.

Unfortunately these parameterizations are sensitive to swell. An increase in swell height typically reduces dissipation at the windsea peak because the mean wavenumber $\overline{k}$ and thus the mean steepness $\overline{\alpha}$ are reduced. For $p < 2$, as in the WAM-Cycle 4 and BJA parameterizations, this sensitivity is much

larger and opposite to the expected effect of short wave modulation by long waves.

The source term code was generalized to allow the use of WAM4, BJA or others ECWAM parameterization, via a simple change of the parameters in the namelists SIN3 and SDS3, see Tables 2.4 and 2.5. At present, the default values of the namelist parameters correspond to BJA (Bidlot et al., 2005).

2.3.10 $S_{\rm in} + S_{\rm ds}$: Ardhuin et al. 2010 ...

Switch:	`ST4`
Origination:	WAVEWATCH III
Provided by:	F. Ardhuin

This family of parameterizations uses a positive part of the wind input taken from WAM cycle 4 with an ad hoc reduction of $u_\star$, implemented in order to allow a balance with a saturation-based dissipation that uses different options for a cumulative term. This correction also reduces the drag coefficient at high winds. This is done by reducing the wind input for high frequencies and high winds. Many different adjustments can be made by changing the namelist parameters. A few successful combinations are given by tables 2.6 and 2.7, with results described by (Rascle and Ardhuin, 2013; Stopa et al., 2016). Further calibration to any particular wind field should be done for best performance. Guidance for this is given by Stopa (2018). We also note that the particular set of parameters T400 corresponds to setting IPHYS=1 in the ECWAM code cycle 45R2, with a few differences related to the fact that precomputed stress tables have now been removed from ECWAM.

The reduction of $u_\star$ in eq. (2.99) is obtained by replacing it with $u'_\star(k)$ defined for each frequency as

$$\left(u'_\star\right)^2 = \left| u_\star^2 \left(\cos\theta_u, \sin\theta_u\right) - |s_u| \int_0^k \int_0^{2\pi} \frac{S_{in}\left(k',\theta\right)}{C} \left(\cos\theta, \sin\theta\right) \mathrm{d}k' \mathrm{d}\theta, \right| \tag{2.110}$$

where the sheltering coefficient $|s_u| \sim 1$ can be used to tune the stresses at high winds, which would be largely overestimated for $s_u = 0$. For $s_u > 0$ this sheltering is also applied within the diagnostic tail in eq. (2.104),

which requires the estimation of a 3-dimensional look-up table for the high frequency stress, the third parameter being the energy level of the tail.

The `STAB3` switch, described above for use with `ST3`, may also be used with `ST4`. If `STAB3` is used, the air-sea temperature differences should be provided by the user, e.g. using ww3_prep.

The swell dissipation parameterization of Ardhuin et al. (2009a) is activated by setting s_1 to a non-zero integer value, and is given by a combination of the viscous boundary layer value,

$$\mathcal{S}_{\mathrm{out,vis}}\left(k,\theta\right) = -s_5 \frac{\rho_a}{\rho_w} \left\{ 2k\sqrt{2\nu\sigma} \right\} N\left(k,\theta\right), \tag{2.111}$$

with the turbulent boundary layer expression

$$\mathcal{S}_{\mathrm{out,tur}}\left(k,\theta\right) = -\frac{\rho_a}{\rho_w} \left\{ 16 f_e \sigma^2 u_{\mathrm{orb},s}/g \right\} N\left(k,\theta\right), \tag{2.112}$$

giving the full term

$$\mathcal{S}_{\mathrm{out}}\left(k,\theta\right) = r_{vis}\mathcal{S}_{\mathrm{out,vis}}\left(k,\theta\right) + r_{tur}\mathcal{S}_{\mathrm{out,tur}}\left(k,\theta\right), \tag{2.113}$$

where the two weights r_{vis} and r_{tur} are defined from a modified air-sea boundary layer significant Reynolds number $\mathrm{Re} = 2u_{\mathrm{orb},s}H_s/\nu_a$

$$r_{\mathrm{vis}} = 0.5(1 - \tanh((\mathrm{Re} - \mathrm{Re}_c)/s_7), \tag{2.114}$$
$$r_{\mathrm{tur}} = 0.5(1 + \tanh((\mathrm{Re} - \mathrm{Re}_c)/s_7). \tag{2.115}$$

The significant surface orbital velocity is defined by

$$u_{\mathrm{orb},s} = 2\left[\iint \sigma^3 \, N(k,\theta) \, dkd\theta\right]^{1/2}. \tag{2.116}$$

The first equation (2.111) is the linear viscous decay by Dore (1978), with ν_a the air viscosity and s_5 is an $O(1)$ tuning parameter. A few tests have indicated that a threshold $\mathrm{Re}_c = 2\times10^5 \times (4\ \mathrm{m}/H_s)^{(1-s_6)}$ provides reasonable result with $s_6 = 0$, although it may also be a function of the wind speed, and we have no explanation for the dependence on H_s. With $s_6 = 1$, a constant threshold close to 2×10^5 provides similar – but less accurate – results.

Par.	WWATCH var.	namelist	T471	T471f	T400/$I_{\text{phys}} = 1$	T405	T500	T601
z_u	ZWND	SIN4	10.0	10.0	10.0	10.0	10.0	10.0
α_0	ALPHA0	SIN4	0.0095	0.0095	**0.0062**	0.0095	0.0095	0.0095
$\beta_{\max}$	BETAMAX	SIN4	1.43	**1.33**	**1.42**	**1.55**	**1.52**	**2.0**
p_{in}	SINTHP	SIN4	2	2	2	2	2	**1**
z_α	ZALP	SIN4	0.006	0.006	0.008	0.006	0.006	0.006
s_u	TAUWSHELTER	SIN4	0.3	0.3	**0.25**	**0.0**	**1.0**	**0.5**
s_1	SWELLF	SIN4	0.66	0.66	0.66	0.8	0.8	0.66
s_2	SWELLF2	SIN4	-0.018	-0.018	-0.018	-0.018	-0.018	-0.018
s_3	SWELLF3	SIN4	0.022	0.022	0.022	**0.015**	**0.015**	0.022
Re_c	SWELLF4	SIN4	1.5×10^5	1.5×10^5	1.5×10^5	$\mathbf{10^5}$	$\mathbf{10^5}$	1.5×10^5
s_5	SWELLF5	SIN4	1.2	1.2	1.2	1.2	1.2	1.2
s_6	SWELLF6	SIN4	0.	0.	**1.0**	0.	0.	0.
s_7	SWELLF7	SIN4	3.6×10^5	3.6×10^5	3.6×10^5	**0.0**	**0.0**	3.6×10^5
z_r	Z0RAT	SIN4	0.04	0.04	0.04	0.04	0.04	0.04
$z_{0,\max}$	Z0MAX	SIN4	1.002	1.002	1.002	**0.002**	1.002	1.002

Table 2.6: Parameter values for T471, T471f, T400, T405, T500, and T601 source term parameterizations that can be reset via the SIN4 namelist. Please note that the names of the variables only apply to the namelists. In the source term module the names are slightly different, with a doubled first letter, in order to differentiate the variables from the pointers to these variables, and the SWELLFx are combined in one array SSWELLF. Values highlighted in bold are different from the default values set by ww3_grid.

TEST471 generally provides the best results at global scale when using ECMWF winds, with the only serious problem being a low bias for $H_s >$ 8 m. TEST451f corresponds to a retuning for CSFR wind reanalysis from NCEP/NCAR (Saha et al., 2010), and has almost no bias all the way to $H_s = 15$ m. Simulations and papers prepared before March 2012, used slightly different values, *e.g.* TEST441 and TEST441f can be recovered by setting SWELLF7 to 0, and TEST471 also used $s_u = 1$ and a few other adjustements (see manual of version 4.18). TEST405 is slightly superior for short fetches, and TEST500 is intermediate in terms of quality but it also includes depth-induced breaking in the same formulation, and thus may be more appropriate for depth-limited conditions.

Eq. (2.112) is a parameterization for the nonlinear turbulent decay. When comparing model results to observations, it was found that the model tended to underestimate large swells and overestimate small swells, with regional biases. This defect is likely due, in part, to errors in the generation or nonlinear evolution of theses swells. However, it was chosen to adjust f_e as a function of the wind speed and direction,

$$f_e = s_1 f_{e,GM} + [|s_3| + s_2 \cos(\theta - \theta_u)]\, u_\star / u_{\rm orb}, \tag{2.117}$$

where $f_{e,GM}$ is the friction factor given by Grant and Madsen's (1979) theory for rough oscillatory boundary layers without a mean flow, using a roughness length adjusted to r_z times the roughness for the wind z_1. The coefficient s_1 is an $O(1)$ tuning parameter, and the coefficients s_2 and s_3 are two other adjustable parameters for the effect of the wind on the oscillatory air-sea boundary layer. When $s_2 < 0$, wind opposing swells are more dissipated than following swells. Further, if $s_3 > 0$, $\mathcal{S}_{out}$ is applied to the entire spectrum and not just the swell.

The dissipation term is parameterized from the wave spectrum saturation, following the general ideas of Phillips (1985), which were initially explored in a numerical modeling framework by Alves and Banner (2003). However, using a saturation parameterization only makes sense when the spectrum is smooth. In general, going from the spectral space to the physical space requires integrating the saturation over a finite spectral band in wavenumber and direction to compute the breaking probability and then deconvolve this integral to obtain a spectral dissipation rate. Because such operations would be too time consuming we have implemented two approaches. One uses on integration over directions only (Ardhuin et al., 2010), while the other uses

an integration over frequency bands (Filipot and Ardhuin, 2012a). These are activated by different namelist parameters, see e.g. T471 for the first approach and T500 for the second.

Because the directional wave spectra were too narrow when using a saturation spectrum integrated over the full circle (Ardhuin and Boyer, 2006), (Ardhuin et al., 2010) restricted over a sector of half-width Δ_θ,

$$B'\left(k,\theta\right) = \int_{\theta-\Delta_\theta}^{\theta+\Delta_\theta} \sigma k^3 cos^{\mathrm{sB}}\left(\theta-\theta'\right) N(k,\theta')\mathrm{d}\theta'. \tag{2.118}$$

As a result, a sea state with two systems of same energy but opposite direction will typically produce less dissipation than a sea state with all the energy radiated in the same direction.

Based on recent analysis by Veras Guimarães et al. (2018) and Peureux et al. (2019), this saturation is enhanced by a factor that represents the effect of long waves on short waves

$$1 + M_\theta\sqrt{\mathrm{mss}_\theta} + N_\theta\sqrt{\mathrm{nss}_\theta}. \tag{2.119}$$

where M_θ is twice the modulation transfer function for short wave steepness, with $M_\theta = 8$ when following the simplified theory by Longuet-Higgins and Stewart (1960) and using the root mean square enhancement of B over a long wave cycle. N_θ is an additional straining factor due to the instability of the wave action envelope of short waves propagating in the direction close to that of the long wave (Peureux et al., 2019). The squared slopes mss_θ is the mean square slope in direction θ, wheras nss_θ is a slope of long waves propagating in a narrow window $\pm\delta_\theta$, around the short wave direction θ.

We finally define our dissipation term as the sum of the saturation-based term and a cumulative breaking term $S_{\mathrm{bk,cu}}$,

$$\begin{aligned}\mathcal{S}_{ds}(k,\theta) &= \sigma\frac{C_{\mathrm{ds}}^{\mathrm{sat}}}{B_r^2}\left[\delta_d \max\left\{B\left(k\right)-B_r,0\right\}^2\right. \\ &\quad \left.+\left(1-\delta_d\right)\max\left\{B'\left(k,\theta\right)-B_r,0\right\}^2\right] N(k,\theta) \\ &\quad +\mathcal{S}_{\mathrm{bk,cu}}(k,\theta)+\mathcal{S}_{\mathrm{turb}}(k,\theta).\end{aligned} \tag{2.120}$$

where

$$B\left(k\right) = \max\left\{B'(k,\theta),\theta\in[0,2\pi[\right\}. \tag{2.121}$$

The combination of an isotropic part (the term that multiplies δ_d) and a direction-dependent part (the term with $1-\delta_d$) was intended to allow some control of the directional spread in resulting spectra.

The cumulative breaking term $\mathcal{S}_{\rm bk,cu}$ represents the smoothing of the surface by big breakers with celerity C' that wipe out smaller waves of phase speed C. Due to uncertainties in the estimation of this effect in various observations, we use the theoretical model of Ardhuin et al. (2009b). Briefly, the relative velocity of the crests is the norm of the vector difference, $\Delta_C = |\mathbf{C}-\mathbf{C}'|$, and the dissipation rate of short wave is simply the rate of passage of the large breaker over short waves, i.e. the integral of $\Delta_C \Lambda(\mathbf{C}) d\mathbf{C}$, where $\Lambda(\mathbf{C})d\mathbf{C}$ is the length of breaking crests per unit surface that have velocity components between C_x and $C_x + dC_x$, and between C_y and $C_y + dC_y$ (Phillips, 1985). Here Λ is inferred from breaking probabilities. Based on Banner et al. (2000, figure 6, $b_T = 22\,(\varepsilon - 0.055)^2$), and taking their saturation parameter ε to be of the order of $1.6\sqrt{B'(k,\theta)}$, the breaking probability of dominant waves is approximately

$$P = 56.8 \left(\max\{ \sqrt{B'(k,\theta)} - \sqrt{B'_r}, 0\} \right)^2 . \tag{2.122}$$

However, because they used a zero-crossing analysis, for a given wave scale, there are many times when waves are not counted because the record is dominated by another scale: in their analysis there is only one wave at any given time. This tends to overestimate the breaking probability by a factor of 2 (Filipot et al., 2010), compared to the present approach in which it is considered that several waves (of different scales) may be present at the same place and time. This effect is corrected simply dividing P by 2.

With this approach the spectral density of crest length (breaking or not) per unit surface $l(\mathbf{k})$ such that $\int l(\mathbf{k}) \mathrm{d}k_x \mathrm{d}k_y$, we take

$$l(\mathbf{k}) = 1/(2\pi^2 k), \tag{2.123}$$

and the spectral density of breaking crest length per unit surface is $\Lambda(\mathbf{k}) = l(\mathbf{k})P(\mathbf{k})$. Assuming that any breaking wave instantly dissipates all the energy of all waves with frequencies higher than a factor $r_{\rm cu}$ or more, the cumulative dissipation rate is simply given by the rate at which these shorter waves are taken over by larger breaking waves, times the spectral density, namely

$$\mathcal{S}_{\rm bk,cu}(k,\theta) = -C_{\rm cu} N\,(k,\theta) \int_{f'<r_{\rm cu} f} \Delta_C \Lambda(\mathbf{k}') \mathrm{d}\mathbf{k}', \tag{2.124}$$

where $r_{\rm cu}$ defines the maximum ratio of the frequencies of long waves that will wipe out short waves. This gives the source term,

$$\mathcal{S}_{\rm bk,cu}(k,\theta) = \frac{-14.2 C_{\rm cu}}{\pi^2} N\left(k,\theta\right) \int_0^{r_{\rm cu}^2 k} \int_0^{2\pi} \max\left\{\sqrt{B(f',\theta')} - \sqrt{B_r}, 0\right\}^2 {\rm d}\theta' {\rm d}k' \quad (2.125)$$

We shall take $r_{\rm cu} = 0.5$, and $C_{\rm cu}$ is a tuning coefficient expected to be of order 1, which also corrects for errors in the estimation of l.

Finally, the wave-turbulence interaction term of Teixeira and Belcher (2002) and Ardhuin and Jenkins (2006), is given by

$$\mathcal{S}_{\rm ds}^{\rm TURB}\left(k,\theta\right) = -2 C_{\rm turb} \sigma \cos(\theta_u - \theta) k \frac{\rho_a u_\star^2}{g \rho_w} N\left(k,\theta\right). \quad (2.126)$$

The coefficient $C_{\rm turb}$ is of order 1 and can be used to adjust for ocean stratification and wave groupiness.

All relevant source term parameters can be set via the namelists SIN4 and SDS4 to yield parameterizations TEST441b, TEST405, both described by Ardhuin et al. (2010) or TEST500 described by Filipot and Ardhuin (2012b) (see Tables 2.6 and 2.7). Please note that the DIA constant C has been slightly adjusted in TEST441b, $C = 2.5 \times 10^7$. TEST441f corresponds to a re-tuned wind input formulation when using NCEP/NCAR winds.

Par.	WWATCH var.	namelist	T471	T400/$I_{\text{phys}} = 1$	T405	T500	T601
f_{FM}	FXFM3	SDS4	2.5	2.5	2.5	**9.9**	5
	SDSC1	SDS4	0	0	0	**1.0**	0
$C_{\text{ds}}^{\text{sat}}$	SDSC2	SDS4	-2.2×10^{-5}	-2.2×10^{-5}	-2.2×10^{-5}	**0.0**	-2.2×10^{-5}
$C_{\text{ds}}^{\text{BCK}}$	SDSBCK	SDS4	0	0	0	**0.185**	0
$C_{\text{ds}}^{\text{HCK}}$	SDSHCK	SDS4	0	0	0	**1.5**	0
Δ_θ	SDSDTH	SDS4	80	80	80	80	80
δ_θ	SDSSTRAINA	SDS4	0	0	0	0	**15**
M_θ	SDSSTRAIN	SDS4	0	0	0	0	**10**
N_θ	SDSSTRAIN2	SDS4	0	0	0	0	**20**
B_r	SDSBR	SDS4	0.0009	0.0009	**0.00085**	0.0009	0.0009
C_{cu}	SDSCUM	SDS4	-0.40344	**0.0**	**0.0**	-0.40344	-0.40344
s_{B}	SDSCOS	SDS4	2.0	2.0	**0.0**	2.0	2.0
B_0	SDSC4	SDS4	1.0	1.0	1.0	1.0	1.0
p^{sat}	SDSP	SDS4	2.0	2.0	2.0	2.0	2.0
C_{turb}	SDSC5	SDS4	0.0	0.0	0.0	0.0	**1.0**
δ_d	SDSC6	SDS4	0.3	0.3	0.3	0.3	0.3
C	NLPROP	SNL1	2.5×10^7	$\mathbf{2.7 \times 10^7}$	$\mathbf{2.7 \times 10^7}$	2.5×10^7	2.5×10^7

Table 2.7: Same as Table 2.6, for the SDS4 and SNL1 namelists. Bold values are different from the default values set by ww3_grid.

2.3.11 $S_{in} + S_{ds}$: Rogers et al. 2012 & Zieger et al. 2015

Switch:	`ST6`
Origination:	AUSWEX, Lake George
Provided by:	A. Babanin, I. Young, M. Donelan, E. Rogers, S. Zieger, Q. Liu

This version implements observation-based physics for deep-water source/sink terms. These include wind input source term, and sink terms due to negative wind input, whitecapping dissipation and wave-turbulence interactions (swell dissipation). The wind input and whitecapping dissipation source terms are based on measurements taken at Lake George, Australia; wave-turbulence dissipation on laboratory experiments and field observations of swell decay; negative input on laboratory testing. Constraint is imposed on the total wind energy input through the wind stress, known independently.

Wind input. Apart from first direct field measurements of the wind input under strong wind forcing, the Lake George experiment revealed a number of new physical features for wind-wave exchange, previously not accounted for: (i) full air-flow separation that leads to a relative reduction of wind input for conditions of strong winds/steep waves; (ii) dependence of the wave growth rate on wave steepness, which signifies nonlinear behavior of the wind-input source function; (iii) enhancement of input in the presence of wave breaking (Donelan et al., 2006; Babanin et al., 2007) (the last feature was not implemented in here). Following Rogers et al. (2012), this input source term is formulated as

$$\mathcal{S}_{in}(k,\theta) = \frac{\rho_a}{\rho_w}\,\sigma\,\gamma(k,\theta)\,N(k,\theta), \tag{2.127}$$

$$\gamma(k,\theta) = G\,\sqrt{B_n}\,W, \tag{2.128}$$

$$G = 2.8 - \Big(1 + \tanh(10\sqrt{B_n}W - 11)\Big), \tag{2.129}$$

$$B_n = A(k)\,N(k)\sigma\,k^3, \tag{2.130}$$

$$W = \left(\frac{U_s}{c} - 1\right)^2. \tag{2.131}$$

In (2.127)−(2.131) ρ_a and ρ_w are densities of air and water, respectively, U_s is the scaling wind speed, c refers to wave phase speed, σ is radian frequency

and k is wavenumber. The spectral saturation (2.130), introduced by Phillips (1984), is a spectral measure of steepness ak. The omni-directional action density is obtained by integration over all directions: $N(k) = \int N(k,\theta)d\theta$. The inverse of the directional spectral narrowness $A(k)$ is defined as $A^{-1}(k) = \int_0^{2\pi}[N(k,\theta)/N_{\max}(k)]d\theta$, where $N_{\max}(k) = \max\{N(k,\theta)\}$, for all directions $\theta \in [0, 2\pi]$ (Babanin and Soloviev, 1987).

Donelan et al. (2006) parameterized the growth rate (2.128) in terms of winds 10 m above the mean surface. Wave models, however, typically employ friction velocity $u_\star = \tau/\rho_a$ to assure a consistent fetch law across different wind speeds (Komen et al., 1994, p. 253). Therefore, Rogers et al. (2012) advocated using an approximation

$$U_s = U_{10} \simeq \Upsilon u_*, \text{ and } \Upsilon = 28 \tag{2.132}$$

by following Komen et al. (1984).

$$W_1 = \max^2\left\{0, \frac{U}{c}\cos(\theta - \theta_w) - 1\right\}, \tag{2.133}$$

$$W_2 = \min^2\left\{0, \frac{U}{c}\cos(\theta - \theta_w) - 1\right\}. \tag{2.134}$$

The directional distribution of W is implemented as the sum of favorable winds (2.133) and adverse winds (2.134), so that they complement one another (i.e. $W = \{W_1 \cup W_2\}$, see *Negative Input* later this section):

$$W = W_1 - a_0\, W_2. \tag{2.135}$$

Wind input constraint. One important part of the input is the calculation of the momentum flux from the atmosphere to the ocean, which must agree with the flux received by the waves. At the surface, the stress $\vec{\tau}$ can be written as the sum of the viscous and wave-supported stress: $\vec{\tau} = \vec{\tau}_v + \vec{\tau}_w$. The wave-supported stress $\vec{\tau}_w$ is used as the principal constraint for the wind input and cannot exceed the total stress $\vec{\tau} \le \vec{\tau}_{tot}$. Here the total stress is determined by the flux parameterization: $\vec{\tau}_{tot} = \rho_a u_\star |u_\star|$. The wave-supported stress τ_w can be calculated by integration over the wind-momentum-input function:

$$\vec{\tau}_w = \rho_w g \int_0^{2\pi}\int_0^{k_{max}} \frac{S_{in}(k',\theta)}{c}\Big(\cos\theta, \sin\theta\Big)dk'd\theta. \tag{2.136}$$

Computation of the wave-supported stress (2.136) includes the resolved part of the spectrum up to the highest discrete wavenumber k_{max}, as well as the stress supported by short waves. To account for the latter, an f^{-5} diagnostic tail is assumed beyond the highest frequency in the energy density spectrum. In order to satisfy the constraint and in the case of $\vec{\tau} > \vec{\tau}_{tot}$, a wavenumber dependent factor L is applied to reduce energy from the high frequency part of the spectrum: $S_{in}(k') = L(k')\, S_{in}(k')$ with

$$L(k') = \min\Big\{1, \exp\big(\mu\,[1 - U/c]\big)\Big\}. \tag{2.137}$$

The reduction (2.137) is a function of wind speed and phase speed and follows an exponential form designed to reduce energy from the discrete part of the spectrum. The strength of reduction is controlled by coefficient μ, which has a greater impact at high frequencies and only little impact on the energy-dominant part of the spectrum. The value of μ is dynamically calculated by iteration at each integration time step (Tsagareli et al., 2010).

The drag coefficient is given by

$$C_d \times 10^4 = 8.058 + 0.967 U_{10} - 0.016 U_{10}^2, \tag{2.138}$$

which was selected and implemented as switch `FLX4`. The parameterization was proposed by Hwang (2011) and accounts for saturation, and further decline for extreme winds, of the sea drag at wind speeds in excess of 30 m s^{-1}. To prevent $u_\star$ from dropping to zero at very strong winds ($U_{10} \geq 50.33$m s^{-1}) expression (2.138) was modified to yield $u_\star = 2.026$m s^{-1}. *Important!* In `ST6`, bulk adjustment to any uniform bias in the wind input field is done in terms of the wind stress parameter $u_\star$ rather than U_{10}. In order to achieve that, the factor in expression $C_d \times 10^4$ on the left hand side of (2.138) was substituted with $C_d \times$ FAC and added as the FLX4 namelist parameter `CDFAC` (see *Bulk Adjustment* at the end of this section). The viscous drag coefficient,

$$C_v \times 10^3 = 1.1 - 0.05 U_{10}, \tag{2.139}$$

was parameterized by Tsagareli et al. (2010) as a function of wind speed using data from Banner and Peirson (1998).

Negative Input. Apart from the positive input, `ST6` also has a negative input term in order to attenuate the growth of waves in those parts of the wave spectrum where an adverse component of the wind stress is present

(2.133–2.134). The growth rate for adverse winds is negative (Donelan, 1999) and is applied after the constraint of the wave-supported stress τ_w is met. The value of a_0 (in 2.135) is a tuning parameter in the parameterization of the input and is adjustable through the SIN6 namelist parameter `SINA0`.

Whitecapping Dissipation. For dissipation due to wave breaking, the Lake George field study revealed a number of new features: (i) the threshold behavior of wave breaking (Babanin et al., 2001). The waves do not break unless they exceed a generic steepness in which case the wave breaking probability depends on the level of excedence above this threshold steepness. For waves below the critical threshold, whitecapping dissipation is zero. (ii) the cumulative dissipative effect due to breaking and dissipation of short waves affected by longer waves (Donelan, 2001; Babanin and Young, 2005; Moon et al., 2006; Young and Babanin, 2006; Babanin et al., 2010), (iii) nonlinear dissipation function at strong winds (Moon et al., 2006; Babanin et al., 2007), (iv) bimodal distribution of the directional spreading of the dissipation (Young and Babanin, 2006; Babanin et al., 2010) (the last feature was not implemented in `ST6`). Following Rogers et al. (2012), the whitecapping dissipation term is implemented as:

$$\mathcal{S}_{ds}(k,\theta) = \Big[T_1(k,\theta) + T_2(k,\theta)\Big]\; N(k,\theta), \tag{2.140}$$

where T_1 is the inherent breaking term, expressed as the traditional function of the wave spectrum, and T_2, expressed as an integral of the wave spectrum below wavenumber k, accounts for the cumulative effect of short-wave breaking or dissipation due to longer waves at each frequency/wavenumber. The inherent breaking term T_1 is the only breaking-dissipation term if this frequency is at or below the spectral peak. Once the peak moves below this particular frequency, T_2 becomes active and progressively more important as the peak downshifts further.

The threshold spectral density F_{T} is calculated as

$$F_{\mathrm{T}}(k) = \frac{\varepsilon_{\mathrm{T}}}{A(k)\,k^3}, \tag{2.141}$$

where k is the wavenumber and with $\varepsilon_{\mathrm{T}} = 0.035^2$ being an empirical constant (Babanin et al., 2007; Babanin, 2011). Let the level of exceedence above the critical threshold spectral density (at which stage wave breaking is predominant) be defined as $\Delta(k) = F(k) - F_{\mathrm{T}}(k)$. Furthermore, let $\mathcal{F}(k)$ be a

generic spectral density used for normalization, then the inherent breaking component can be calculated as

$$T_1(k) = a_1 A(k) \frac{\sigma}{2\pi} \left[\frac{\Delta(k)}{\mathcal{F}(k)} \right]^{p_1} . \tag{2.142}$$

The cumulative dissipation term is not local in frequency space and is based on an integral that grows towards higher frequencies, dominating at smaller scales:

$$T_2(k) = a_2 \int_0^k A(k) \frac{c_g}{2\pi} \left[\frac{\Delta(k)}{\mathcal{F}(k)} \right]^{p_2} dk. \tag{2.143}$$

The dissipation terms (2.142)−(2.143) depend on five parameters: a generic spectral density $\mathcal{F}(k)$ used for normalization, and four coefficients a_1, a_2, p_1, and p_2. The coefficients p_1 and p_2 control the strength of the normalized threshold spectral density $\Delta(k)/\mathcal{F}(k)$ of the dissipation terms. Namelist parameter `SDSET` changes between the spectral density $F(k)$ and threshold spectral density $F_{\rm T}(k)$ for normalization in (2.142)–(2.143). According to Babanin et al. (2007) and Babanin (2009), the directional narrowness parameter is set to unity $A(k) \approx 1$ in Eqs. (2.141)−(2.143).

Rogers et al. (2012) calibrated the dissipation terms based on duration-limited academic tests. Calibration coefficients used in `ST6` and listed in Table 2.8 differ somewhat from those of Rogers et al. (2012) mainly due to the fact that the wave-supported stress $\vec{\tau}_w$ is implemented in the form of vector components and the non-breaking swell dissipation (2.144) was previously not accounted for in Rogers et al. (2012).

Swell Dissipation. In the absence of wave breaking, other mechanisms of wave attenuation are present. Here, they are referred to as swell dissipation and parameterized in terms of the interaction of waves with oceanic turbulence (Babanin, 2011). This mechanism, however, remains active for the wind-generated waves too. Its contribution across the spectrum is small, if the spectrum is above the wave-breaking threshold, but it is dominant at the front face of the spectrum, or even at the peak in case of the full Pierson-Moscowitz development.

$$\mathcal{S}_{swl}(k,\theta) = -\frac{2}{3} b_1 \sigma \sqrt{B_n}\, N(k,\theta). \tag{2.144}$$

Parameter	WWATCH var.	namelist	vers. 4.18	vers. 5.16	vers. 6.07
F_{T}	SDSET	SDS6	T	T	T
a_1	SDSA1	SDS6	6.24E-7	3.74E-7	4.75E-6
p_1	SDSP1	SDS6	4	4	4
a_2	SDSA2	SDS6	8.74E-6	5.24E-6	7.00E-5
p_2	SDSP2	SDS6	4	4	4
$\Upsilon^{\dagger}$	SINWS	SIN6	n/a	n/a	32.0
$N_{hf}{}^{\dagger}$	SINFC	SIN6	n/a	n/a	6.0
a_0	SINA0	SIN6	0.04	0.09	0.09
b_1 is constant	CSTB1	SWL6	n/a	F	F
b_1, B_1	SWLB1	SWL6	0.25E-3	0.0032	0.0041
FAC	CDFAC	FLX4	1.00E-4	1.00E-4	1.0
C	NLPROP	SNL1	3.00E7	3.00E7	3.00E7

† In WW3 version 4.18 and 5.16, $\Upsilon = 28.0$ and $N_{hf} = 6.0$ were hard-coded in `ST6` module.

Table 2.8: Summary of calibration parameters for `ST6` when it is applied with the `DIA` nonlinear solver (section 2.3.2). Values tabulated represent default model settings. Abbreviation "n/a" indicates that the variable is not applicable in that release of the code.

By making coefficient b_1 in Eq. (2.144) dependent on steepness the large gradient in the spatial bias in wave height can be reduced:

$$b_1 = B_1 \, 2\sqrt{E} \, k_p. \tag{2.145}$$

In Eq. (2.145), B_1 is a scaling coefficient, E is the total sea surface variance Eq. (2.23) and k_p is the peak wavenumber. Eq. (2.145) can be flagged through the SWL6 namelist parameter `CSTB1`.The value for the coefficient B_1 in Eq. (2.145) and/or b_1 in Eq. (2.144) is customizable through the SWL6 namelist parameter `SWLB1` (see Table 2.8).

Updates since vers. 6.07 Following Rogers et al. (2012), the scaling wind speed $U_s = 28u_*$ (2.132) were adopted in vers. 4.18 and vers. 5.16 (Table 2.8). Such configurations of `ST6` have been proven skillful for different spatial scales and under different weather conditions (e.g. Zieger et al., 2015; Liu et al., 2017). Zieger et al. (2015, their Fig. 5), however, also suggested `ST6` (vers. 4.18 and vers. 5.16) was inclined to overestimate the energy level of the high-frequency tail of the spectrum, indicating an inaccurate balance of different source terms in this specific frequency range.

Rogers (2014, unpublished work) found that using $U_s = 32u_*$ (i.e., $\Upsilon = 32$) could improve model skills in estimating tail level in the `ST6` implementation in SWAN [see also Rogers (2017)]. Following this, Liu et al. (2019) carried out a thorough recalibration of `ST6` with WW3, and the new set of parameters (i.e., a_0, a_1, a_2, B_1) is summarized in the last column of Table 2.8. The updated `ST6` package not only performs well in predicting commonly-used bulk wave parameters (e.g., significant wave height and wave period) but also yields a clearly-improved estimation of high-frequency energy level (in terms of saturation spectrum and mean square slope). In the duration-limited test, the omnidirectional frequency spectrum $E(f)$ from the recalibrated `ST6` shows a clear transition behavior from the power law of approximately f^{-4} to the power law of about f^{-5}, comparable to previous field studies (Forristall, 1981).

Apart from applying `ST6` with the `DIA` (section 2.3.2) nonlinear solver, Liu et al. (2019) also made an attempt to run `ST6` with the `GMD` parameterization of S_{nl} (section 2.3.4). As a first step, only the `GMD` configuration with 5 quadruplets and a three-parameter quadruplet definition (i.e., `G35` in Tolman, 2013a) was adopted. The tunable parameters of the `GMD` which were specifically optimized for `ST6` by using the holistic genetic optimization

ST6	a_0	a_1	a_2	B_1
	0.05	4.75×10^{-6}	7.00×10^{-5}	6.00×10^{-3}
GMD	λ	μ	θ_{12} (°)	C_{deep}
	0.127	0.000	3.0	4.88×10^{7}
	0.127	0.097	21.0	1.26×10^{8}
	0.233	0.098	26.5	6.20×10^{7}
	0.283	0.237	24.7	2.83×10^{7}
	0.355	0.183	–	1.17×10^{7}

Table 2.9: Summary of calibration parameters for ST6 when it is applied with the GMD nonlinear solver (or specifically, G35)

technique designed by Tolman and Grumbine (2013), and the corresponding tunable parameters of ST6 are summarized in Table 2.9[4],[5]. Liu et al. (2019) demonstrated that in the duration-limited test, the GMD-simulated $E(f)$ is in excellent agreement with that from the exact solutions of S_{nl}(e.g., WRT; section 2.3.3) (see also Tolman, 2013a)]. In a 1-yr global hindcast, the DIA-based model overestimates the low-frequency wave energy (wave period $T > 16$ s) by 90%. Such model errors are reduced significantly by the GMD to ∼20%. It is noteworthy that the computational expense of the GMD approach presented in Table 2.9, however, is about 5 times larger than that of the DIA.

To flexibly control the high-frequency extent of the prognostic frequency region, we introduced

$$f_{hf} = N_{hf}/T_{m0,-1} \tag{2.146}$$

in vers. 6.07, where f_{hf} is the cut-off high-frequency limit (2.17), $T_{m0,-1}$ is the mean wave period defined in (2.235). N_{hf} is set through a namelist parameter SINFC (Table 2.8), and a negative N_{hf} (say, $N_{hf} = -1$) means *the high-frequency spectral tail evolves freely without any prescribed slope.*

[4]For the fifth quadruplet layout of GMD shown in Table 2.9, the three-parameter (λ, μ, θ_{12}) quadruplet definition degrades to a two-parameter (λ, μ) form, and θ_{12} is implied by the value of μ (Tolman, 2013a). Accordingly, a combination of ST6+GMD and the conservative nonlinear high-frequency filter of Tolman (2011b) [see also section 2.3.6] might be necessary to stabilize the model integration, particularly for high-resolution spectral grid (say, $\Delta\theta < 5°$).

[5]Unlike Table 2.8, here we only show important parameters of ST6 and GMD. The reader is refered to bin/README.UoM for the detailed set up of namelist variables for this specific model configuration.

Dominant Wave Breaking Probability Following Babanin et al. (2001, their Fig. 12), the dominant wave breaking probability b_T can be estimated from the wave spectrum $F(f,\theta)$ according to the following parametric form

$$b_T = 85.1\Big[(\epsilon_p - 0.055)(1 + H_s/d)\Big]^{2.33}, \tag{2.147}$$

where H_s is the significant wave height, d is the water depth, ϵ_p is the significant steepness of the spectral peak, given by

$$\begin{aligned} \epsilon_p &= H_p k_p/2 \\ H_p &= 4\Big[\int_0^{2\pi}\int_{0.7f_p}^{1.3f_p} F(f,\theta)\mathrm{d}f\mathrm{d}\theta\Big]^{1/2} \end{aligned} \tag{2.148}$$

where k_p and f_p are the peak wavenumber and frequency, respectively. Note that only the contribution from wind seas is considered when we calculate H_s, H_p and f_p (k_p) from $F(f,\theta)$ (Eqs. 2.147 and 2.148). Following Janssen et al. (1989) and Bidlot (2001), we consider spectral components as wind seas when

$$\frac{c(k)}{U_{10}\cos(\theta - \theta_u)} < \beta_w, \tag{2.149}$$

where $c(k)$ is the phase velocity, U_{10} is the wind speed at 10 m above the sea surface, θ_u is the wind direction and β_w is the constant wind forcing parameter. We implemented β_w as a tunable parameter (the MISC namelist parameter `BTBET`), and used $\beta_w = 1.2$ by default.

Bulk Adjustments. The source term `ST6` has been calibrated with flux parameterization `FLX4`. Bulk adjustment to the wind filed can be achieved by re-scaling the drag parameterization `FLX4` through the FLX4 namelist parameter `CDFAC=1.0E-4`[6]. This has a similar effect to tuning variable β_{max} in `ST4` source term package, equations (2.99) and (2.104), which is customizable through namelist parameter `BETAMAX` (see section 2.3.9–2.3.10). Ardhuin et al. (2011a) and Rascle and Ardhuin (2013) listed different sets of values that allow us to adjust to different wind fields. When optimizing the wave model, it is recommended to only re-tune parameters a_0, b_1 and FAC. Again, FAC can potentially eliminate a bias in the wind field, which

[6]This was changed to `CDFAC=1.0` since vers. 6.07 as the magnitude 10^{-4} was hard-coded in `FLX4` module. (Table 2.8)

typically changes with the selection of the reanalysis product. This reduction was tested for extreme wind conditions such as hurricanes (Zieger et al., 2015). In a global hindcast, the coefficient for the negative input can be used to tune the bulk of the wave heights in scatter comparisons, whereas the scaling coefficient for swell dissipation primarily affects large sea states. When the discrete interaction approximation (`DIA`) is used to compute the four-wave interaction, the default value for the proportionality constant changes to $C = 3.00 \times 10^7$.

Limitations of the code: For vers. 4.18 and vers. 5.16, in cases where the minimum time step for the dynamical source term integration is much smaller than the overall time step (i.e. less than 1/15th), the model becomes unstable. The issue has been solved since vers. 6.07.

2.3.12 S_{ln}: Cavaleri and Malanotte-Rizzoli 1981

Switch:	`LN1`
Origination:	Pre-WAM
Provided by:	H. L. Tolman

A linear input source term is useful to allow for the consistent spin-up of a model from quiescent conditions, and to improve initial wave growth behavior. The parameterization of Cavaleri and Malanotte-Rizzoli (1981) is available in WAVEWATCH III, with a filter for low-frequency energy as introduced by Tolman (1992). The input term can be expressed as

$$\mathcal{S}_{lin}(k,\theta) = 80 \left(\frac{\rho_a}{\rho_w}\right)^2 g^{-2} k^{-1} \max\left[0, u_* \cos(\theta - \theta_w)\right]^4 \; G \quad , \tag{2.150}$$

where ρ_a and ρ_w are the densities of air and water, respectively, and where G is the filter function

$$G = \exp\left[-\left(\frac{f}{f_{filt}}\right)^{-4}\right] \quad . \tag{2.151}$$

In Tolman (1992) the filter frequency f_{filt} was given as the Pierson-Moskowitz frequency f_{PM}, which in turn was estimated as in Eq. (2.61). In the present implementation, the filter can be related to both f_{PM} and the cut-off frequency of the prognostic part of the spectrum f_{hf} as defined in Eq. (2.17)

$$f_{filt} = \max\left[\alpha_{PM} f_{PM}, \alpha_{hf} f_{hf}\right] \quad , \tag{2.152}$$

where the constants α_{PM} and α_{hf} are user-defined. Default values of these constants are set to $\alpha_{PM} = 1$ and $\alpha_{hf} = 0.5$. Addition of the dependency on f_{hf} assures consistent growth behavior at all fetches, without the possibility of low-frequency linear growth to dominate at extremely short fetches.

2.3.13 S_{bot}: JONSWAP bottom friction

Switch:	`BT1`
Origination:	JONSWAP experiment
Provided by:	H. L. Tolman

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

$$\mathcal{S}_{bot}(k,\theta) = 2\Gamma \, \frac{n - 0.5}{gd} \, N(k,\theta) \, , \tag{2.153}$$

where Γ is an empirical constant, which is estimated as $\Gamma = -0.038\ \mathrm{m^2 s^{-3}}$ for swell (Hasselmann et al., 1973), and as $\Gamma = -0.067\ \mathrm{m^2 s^{-3}}$ for wind seas (Bouws and Komen, 1983). n is the ratio of phase velocity to group velocity given by (2.6). The default value for $\Gamma = -0.067$ can be redefined by the user by changing the SBT1 namelist parameter `GAMMA`.

2.3.14 S_{bot}: SHOWEX bottom friction

Switch:	`BT4`
Origination:	Crest model
Provided by:	F. Ardhuin

A more realistic parameterization for sandy bottoms is based on the eddy viscosity model by Grant and Madsen (1979) and a roughness parameterization that includes the formation of ripples and transition to sheet flow. The

Par.	WWATCH var.	namelist	SHOWEX	Tolman (1994)
A_1	RIPFAC1	BT4	0.4	1.5
A_2	RIPFAC2	BT4	-2.5	-2.5
A_3	RIPFAC3	BT4	1.2	1.2
A_4	RIPFAC4	BT4	0.05	0.0
σ_d	SIGDEPTH	BT4	0.05	user-defined
A_5	BOTROUGHMIN	BT4	0.01	0.0
A_6	BOTROUGHFAC	BT4	1.00	0.0

Table 2.10: Parameter values for the SHOWEX bottom friction (default values) and the original parameter values used by Tolman (1994). Source term parameters can be modified via the BT4 namelist. Please note that the name of the variables only apply to the namelists. In the source term module the seven variables are contained in the array SBTCX.

parameterization of Tolman (1994), was adjusted by Ardhuin et al. (2003) to field measurements from the DUCK'94 and SHOWEX experiments on the North Carolina continental shelf. The parameterization has been adapted to WAVEWATCH III by also including a sub-grid parameterization for the variability of the water depth, as given by Tolman (1995a). This parameterization is activated by the switch BT4.

The source term can be written as

$$\mathcal{S}_{bot}(k,\theta) = -f_e u_b \frac{\sigma^2}{2g\sinh^2(kd)} N(k,\theta)\,, \tag{2.154}$$

where f_e is a dissipation factor that is a function of the r.m.s. bottom orbital displacement amplitude a_b and the Nikuradse roughness length k_N, and u_b is the r.m.s. of the bottom orbital velocity.

The present bed roughness parameterization (2.155)–(2.161) contains seven empirical coefficients listed in Table 2.10.

The roughness k_N is decomposed in a ripple roughness k_r and a sheet flow

roughness k_s,

$$k_r = a_b \times A_1 \left(\frac{\psi}{\psi_c} \right)^{A_2}, \tag{2.155}$$

$$k_s = 0.57 \frac{u_b^{2.8}}{\left[g\,(s-1) \right]^{1.4}} \frac{a_b^{-0.4}}{(2\pi)^2}. \tag{2.156}$$

In Eqs. (2.155) and (2.156) A_1 and A_2 are empirical constants, s is the sediment specific density, ψ is the Shields number determined from u_b and the median sand grain diameter D_{50},

$$\psi = f'_w u_b^2 / \left[g\,(s-1)\,D_{50} \right], \tag{2.157}$$

with f'_w the friction factor of sand grains (determined in the same way as f_e with D_{50} instead of k_r as the bottom roughness), and ψ_c is the critical Shields number for the initiation of sediment motion under sinusoidal waves on a flat bed. We use an analytical fit (Soulsby, 1997)

$$\psi_c = \frac{0.3}{1 + 1.2 D_*} + 0.055 \left[1 - \exp\left(-0.02 D_* \right) \right], \tag{2.158}$$

$$D_* = D_{50} \left[\frac{g\,(s-1)}{\nu^2} \right]^{1/3}, \tag{2.159}$$

where ν is the kinematic viscosity of water.

When the wave motion is not strong enough to generate vortex ripples, i.e. for values of the Shields number less than a threshold $\psi_{\rm rr}$, k_N is given by a relic ripple roughness $k_{\rm rr}$. The threshold is

$$\psi_{\rm rr} = A_3 \psi_c. \tag{2.160}$$

Below this threshold, k_N is given by

$$k_{\rm rr} = \max\left\{ A_5 \mathrm{m}, A_6 D_{50}, A_4 a_b \right\} \text{ for } \psi < \psi_{\rm rr}. \tag{2.161}$$

2.3.15 S_{mud}: Dissipation by viscous mud (D&L)

Switch:	`BT8`
Origination:	NRL/SWAN
Provided by:	M. Orzech and E. Rogers

Two formulations for wave damping by viscous fluid mud have been implemented in WAVEWATCH III based on earlier implementations in a SWAN code at NRL. As with wave damping by ice (Sect. 2.4.1), both rely on the concept of complex wave number (Eq. (2.182)). Both treat the mud layer as a viscous fluid, and both assume that the mud depth is comparable to its Stokes' boundary layer thickness. The first formulation (Dalrymple and Liu (1978); hereafter D&L) is a numerical solution. The second formulation (Ng (2000); hereafter Ng) is an analytical, asymptotic solution, so calculations tend to be much faster than with D&L. For the range of mud characteristics used by Rogers and Holland (2009), which are based on field measurements (and estimates), the methods produce very similar results.

In each case, the mud-induced dissipation is added to contributions from other source/sink terms in Eq. (2.8).

$$S_{mud} = 2k_i C_{g,mud}, \tag{2.162}$$

where $k_i = imag(k_{mud})$ and $C_{g,mud}$ is the mud-modified wave group velocity.

The above follows from exponential decay of a single wave train with initial amplitude a_0:

$$a = a_0 e^{-k_i x}. \tag{2.163}$$

Both methods operate by solving for a modified dispersion relation, where the wavenumber being solved for, k_{mud}, is a complex number. The D&L method uses an iterative procedure for this dispersion relation. For details, see Section 2 and Appendix B of Dalrymple and Liu (1978). Descriptions specific to `BT9` (Ng) are given in the following section.

To activate viscous mud effects with the (D&L) routines, the user specifies `BT8` in the switch parameter file.

In the case where any of the new ice and mud source functions are activated with the switches `IC1`, `IC2`, `IC3`, `BT8`, or `BT9`, ww3_shel will anticipate instructions for 8 new fields (5 for ice, then 3 for mud). These are given prior to the "water levels" information. The new fields can also be specified as homogeneous field using ww3_shel.inp. The mud parameters are mud density (kg/m^3), mud thickness (m), and mud viscosity (m^2/s), in that order.

The user is referred to the regression tests ww3_tbt1.1 ww3_tbt2.1 for examples of how to use the new mud source functions.

Limitations of the code: In the case of ww3_multi, the interface for the necessary mud and ice forcing fields has only been implemented when using the namelist type of input file. In the case of mud, though the k_r is calculated,

its effect is not passed back to the main program. The only effect is via k_i (dissipation). Full implementation of k_r, already possible with IC3, and will be available in a future version of the model.

Limitations of the physics: 1) Both models (`BT8`, `BT9`) neglect elasticity in the mud layer. 2) Non-Newtonian response of the mud (e.g. as a thixotropic fluid) is not available. 3) Mud thickness should be interpreted not as the total mud thickness, but rather as the thickness of the fluidized mud layer. This value is notoriously difficult to determine in practice (Rogers and Holland (2009)). Fortunately, since WAVEWATCH III supports nonstationary and non-uniform input for the mud parameters, it is possible to address items (2) and (3) via coupling with a numerical model of the mud layer: no additional changes to the WAVEWATCH III code are required for this.

2.3.16 S_{mud}: Dissipation by viscous mud (Ng)

Switch:	`BT9`
Origination:	NRL/SWAN
Provided by:	M. Orzech and E. Rogers

To activate viscous mud effects with the Ng routines, the user specifies `BT9` in the switch parameter file. The Ng method computes k_i as:

$$k_i \approx D_{mud} \equiv \frac{\delta_m (B'_r + B'_i) {k_1}^2}{sinh 2k_1 d + 2k_1 d}, \tag{2.164}$$

Here, δ_m is the Stokes boundary layer thickness for mud, d is water depth, and k_1 is leading order term of the real part of the mud-modified wave number k_{mud}, respectively, in a Taylor expansion about the mud-water interface, and D_{mud} is the leading order term in the full expansion of k_i. B' is a complex coefficient affecting the depth profile of the velocities. For additional details, see Section 2.3.15 and Ng (2000).

2.3.17 S_{db}: Battjes and Janssen 1978

Switch:	`DB1 / MLIM`
Origination:	Pre-WAM
Provided by:	J. H. G. M. Alves

The implementation in WAVEWATCH III of depth-induced breaking algorithms is intended to extend the applicability of the model to within shallow water environments, where wave breaking, among other depth-induced transformation processes, becomes important.

For this reason the approach of Battjes and Janssen (1978, henceforth denoted as BJ78), which is based on the assumption that all waves in a random field exceeding a threshold height, defined as a function of bottom topography parameters, will break. For a random wave field, the fraction of waves satisfying this criterion is determined by a statistical description of surf-zone wave heights (i.e., a Rayleigh-type distribution, truncated at a depth-dependent wave-height maximum).

The bulk rate δ of spectral energy density dissipation of the fraction of breaking waves, as proposed by BJ78, is estimated using an analogy with dissipation in turbulent bores as

$$\delta = 0.25\, Q_b\, f_m\, H_{\max}^2, \tag{2.165}$$

where Q_b is the fraction of breaking waves in the random field, f_m is the mean frequency and $H_{\max}$ is the maximum individual height a component in the random wave field can reach without breaking (conversely, above which all waves would break). In BJ78 the maximum wave height $H_{\max}$ is defined using a Miche-type criterion (Miche, 1944),

$$\bar{k} H_{\max} = \gamma_M \tanh(\bar{k} d), \tag{2.166}$$

where γ_M is a constant factor. This approach also removes energy in deep-water waves exceeding a limiting steepness. This can potentially result in double counting of dissipation in deep-water waves. Alternatively, $H_{\max}$ can be defined using a McCowan-type criterion, which consists of simple constant ratio

$$H_{\max} = \gamma\, d, \tag{2.167}$$

where d is the local water depth and γ is a constant derived from field and laboratory observation of breaking waves. This approach will exclusively represent depth-induced breaking. Although more general breaking criteria for $H_{\max}$ as a simple function of local depth exist (e.g., Thornton and Guza, 1983), it should be noted that the coefficient γ refers to the maximum height of an individual breaking wave within the random field. McCowan (1894)

calculated the limiting wave-height-to-depth ratio for a solitary wave propagating on a flat bottom to be 0.78, which is still used presently as a conservative criteria in engineering applications. The average value found by Battjes and Janssen (1978) was $\gamma = 0.73$. More recent analyses of waves propagating over reefs by Nelson (1994, 1997) suggest a ratio of 0.55.

The fraction of breaking waves Q_b is determined in terms of a Rayleigh-type distribution truncated at $H_{\max}$ (i.e., all broken waves have a height equal to H_{max}), which results in the following expression:

$$\frac{1 - Q_b}{-\ln Q_b} = \left(\frac{H_{rms}}{H_{\max}} \right)^2 , \tag{2.168}$$

where H_{rms} is the root-mean-square wave height. In the current implementation, the implicit equation (2.168) is solved for Q_b iteratively. With the assumption that the total spectral energy dissipation δ is distributed over the entire spectrum so that it does not change the spectral shape (Eldeberky and Battjes, 1996) the following depth-induced breaking dissipation source function is obtained

$$\mathcal{S}_{db}(k,\theta) = -\alpha \frac{\delta}{E} F(k,\theta) = -0.25\, \alpha\, Q_b\, f_m \frac{H_{\max}^2}{E} F(k,\theta), \tag{2.169}$$

where E is the total spectral energy, and $\alpha = 1.0$ is a tunable parameter. The user can select between Eqs. (2.166) and (2.167), and adjust γ and α. Defaults are Eq. (2.167), $\gamma = 0.73$ and $\alpha = 1.0$.

2.3.18 S_{tr}: Triad nonlinear interactions (LTA)

Switch:	`TR1`
Origination:	SWAN
Provided by:	A. Van der Westhuysen

Nonlinear triad interactions are modelled using the LTA model of Eldeberky (1996). This stochastic model is based on the Boussinesq-type deterministic equations of Madsen and Sorensen (1993). These deterministic equations are ensemble averaged, and the hierarchy of spatial evolution equations truncated by a zero-fourth-order-cumulant assumption, yielding a

set of equations for the spectral and bispectral evolution in one-dimension. The bispectrum appearing in the spectral evolution equation is split up into a biamplitude and a biphase. The biphase corresponding to the self interaction of the peak frequency σ_p is parameterised as a function of the local Ursell number by

$$\beta(\sigma_p,\sigma_p) = -\frac{\pi}{2} + \frac{\pi}{2}\tanh\left(\frac{0.2}{Ur}\right), \tag{2.170}$$

in which the spectrally based Ursell number Ur is given by

$$Ur = \frac{g}{8\sqrt{2}\pi^2}\frac{H_s {T_{m01}}^2}{d^2} \ . \tag{2.171}$$

The biamplitude is obtained by spatially integrating the evolution equation for the bispectrum, by which the biamplitude is rendered a spatially local function. This result in a expression for the biamplitude which has a spatially slowly-varying component and a fast-oscillating component, of which the latter is neglected. Using the derived expressions for the biphase and biamplitude, the spectral evolution equation (a one-equation model) can be solved. To reduce the computational cost even further, the complete set of all interacting triads are represented by only the set of *self sum interactions*, that is, triads in which a component of frequency σ interacts with a component of the same frequency to exchange energy flux with a component of frequency $\sigma+\sigma=2\sigma$. The final expression for the effect of triad interactions on a component with frequency σ is made up of two contributions—one adding energy flux to σ (transferred flux arriving from $1/2\sigma$) and one subtracting energy flux from σ (transfer going to 2σ). The expression implemented, adapted for radian frequencies, reads:

$$S_{\rm nl3}(\sigma,\theta) = S^{-}_{\rm nl3}(\sigma,\theta) + S^{+}_{\rm nl3}(\sigma,\theta), \tag{2.172}$$

with

$$S^{+}_{\rm nl3}(\sigma,\theta) = \max[0, \alpha_{\rm EB} 2\pi c c_g J^2 |\sin\beta| \left\{E^2(\sigma/2,\theta) - 2E(\sigma/2,\theta)E(\sigma,\theta)\right\}], \tag{2.173}$$

and

$$S^{-}_{\rm nl3}(\sigma,\theta) = -2S^{+}_{\rm nl3}(2\sigma,\theta) \ . \tag{2.174}$$

Because of a Jacobian in the transfer of the energy flux from σ to 2σ, the flux density arriving at 2σ is half that leaving σ (hence the factor 2 appearing in Eq. (2.174)). The interaction coefficient J, describing self interaction in the nonlinearity range $0 \leq Ur \leq 1$, is given by (Madsen and Sorensen, 1993):

$$J = \frac{k_{\sigma/2}^2 (gd + 2c_{\sigma/2}^2)}{k_\sigma d (gd + \frac{2}{15} gd^3 k_\sigma^2 - \frac{2}{5}\sigma^2 d^2)} \quad . \tag{2.175}$$

The LTA formulation is implemented along each propagation direction of the directional spectrum, yielding an isotropic, directionally decoupled representation of triad interaction. The value of the proportionality coefficient is set at $\alpha_{\rm EB} = 0.05$. The results produced by the LTA are furthermore quite sensitive to the choice of the frequency up to which the interactions are calculated, denoted here as $f_{max,EB}$. (Eldeberky, 1995) recommends that the interactions be computed up to a frequency of 2.5 times the mean frequency ($f_{max,EB} = 2.5 f_{m01}$).

2.3.19 S_{bs}: Bottom scattering

Switch:	`BS1`
Origination:	CREST model
Provided by:	F. Ardhuin

Waves propagating over a sloping bottom are partially reflected. In the limit of small variation in water depth Δd with respect to the mean water depth d, the reflection coefficient is proportional to the bottom spectrum Kreisel (1949) and leads to a redistribution of wave energy in direction. This process may be formulated as a source term, which leads to accurate reflection coefficients when considering the evolution of the spectrum over scales larger than the bottom auto-correlation length, with reasonable accuracy up to $\Delta d/d \simeq 0.6$ (Ardhuin and Magne, 2007). The source term reads,

$$\mathcal{S}_{\rm bs}(\mathbf{k}) = \frac{\pi}{2} \int_0^{2\pi} \frac{k'^2 M^2(\mathbf{k},\mathbf{k}')}{\sigma\sigma' \left(k' C_g' + \mathbf{k}\cdot\mathbf{U}\right)} F^B(\mathbf{k}-\mathbf{k}') \left[N(\mathbf{k}') - N(\mathbf{k})\right] {\rm d}\theta', \tag{2.176}$$

with the coupling coefficient

$$M(\mathbf{k},\mathbf{k}') \simeq M_b(\mathbf{k},\mathbf{k}') = \frac{g\mathbf{k}\cdot\mathbf{k}'}{\cosh(kd)\cosh(k'd)} \tag{2.177}$$

where the effect of bottom-induced current and elevation changes are neglected, as appropriate for low to moderate current velocity relative to the intrinsic phase speed, i.e. $U/C < 0.3$. For larger Froude numbers, in particular in near-blocking conditions, the present implementation is not expected to be accurate. In Eq. (2.176), k and k' are related by the resonance condition, $\omega = \omega'$, i.e. $\sigma + \mathbf{k}\cdot\mathbf{U} = \sigma' + \mathbf{k}'\cdot\mathbf{U}$, where $\mathbf{U}$ is the phase advection velocity (see, e.g., WISE Group, 2007).

The bottom spectrum $F^B(\mathbf{k})$ is specified in the file bottom_spectrum.inp. This spectrum may be determined from multi-beam bathymetric data. In the absence of detailed bathymetric data, the sand dune spectrum may be parameterized based on the work of Hino (1968). Recent observations generally confirm the earlier data on sand dune spectra (Ardhuin and Magne, 2007), with a non-dimensional constant spectrum for large k, i.e. $F^B(\mathbf{k}) \sim k^{-4}$.

The bottom spectrum is double-sided for simplicity of calculation and normalized such that the bottom variance (in square meters) is

$$< d^2 >= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F^B(k_x,k_y)\mathrm{d}k_x\mathrm{d}k_y. \tag{2.178}$$

In the present implementation this bottom spectrum is assumed to be the same at all grid points.

The source term is computed according to different methods depending on the value of the current. For zero current, the interactions only involves waves of the same frequency and the interaction is always the same and linear in terms of the directional spectrum. In this case the interaction is expressed as a matrix problem. Namely, the directional spectrum F is a vector of NTH components, and the source term is a vector of same size that is obtained by the matrix multiplication $S = MF$ with M a square (NTH,NTH) array. This array is a function of the bottom spectrum and the non-dimensional depth kd. This scattering matrix is precomputed and diagonalized as a preprocessing step for a finite number of wavenumber magnitudes (Ardhuin and Herbers, 2002). The cost of this preprocessing increases linearly with the number of discrete wavenumbers.

For non-zero current, the interaction pattern depends on the current magnitude and direction (magnitude only for an isotropic bottom spectrum), and

a precomputation of the scattering matrix would increase the overhead cost by at least one order of magnitude. In the present implementation, the interaction integration with non-zero current is recomputed at every source term call.

2.3.20 S_{uo}: Unresolved Obstacles Source Term

Switch:	`UOST`
Origination:	WAVEWATCH III
Provided by:	L. Mentaschi

Unresolved bathymetric and coastal features, such as cliffs, shoals and small islands, are a major source of local error in spectral wave models. Their dissipating effects can be accumulated over long distances, and neglecting them can compromise the simulation skill on large portions of the domain (Tolman, 2001; Tolman et al., 2002; Tuomi et al., 2014; Mentaschi et al., 2015a). In WAVEWATCH III two approaches are available for subscale modelling the dissipation due to unresolved obstacles: a) a propagation-based approach established for regular grids, described in section 3.4.7 (Tolman, 2003b; Chawla and Tolman, 2008); and b) the Unresolved Obstacles Source Term (UOST, Mentaschi et al., 2018a, 2015b) described here. In addition to supporting virtually any type of mesh, UOST takes into account the directional/spatial layout of the unresolved features, which can improve the model skill with respect to the established approach (Hardy et al., 2000; Mentaschi et al., 2018b).

UOST relies on the hypothesis that any mesh can be considered as a set of polygons, called cells, and that the model estimates the average value of the unknowns in each cell. Given a cell (let us call it A, figure 2.1ab) UOST estimates, for each spectral component, the effect of a) the unresolved features located in A (Local Dissipation, LD); b) the unresolved features located upstream of A, and projecting their shadow on A (Shadow Effect, SE). For the estimation of SE, an upstream polygon A' is defined for each cell/spectral component, as the intersection between the joint cells neighboring A (cells B, C and D in figure 2.1ab), and the flux upstream of A. For each cell or upstream polygon, and for each spectral component, two different transparency coefficients are estimated. 1) The overall transparency coefficient α; and 2) a layout-dependent transparency β, defined as the average transparency of

cell sections starting from the cell upstream side.

The source term can be expressed as:

$$S_{uo} = S_{ld} + S_{se} \ , \tag{2.179}$$

$$S_{ld} = -\psi_{ld}(\mathbf{k}) \frac{1-\beta_l(\mathbf{k})}{\beta_l(\mathbf{k})} \frac{c_g(\mathbf{k})}{\Delta L} N \ , \tag{2.180}$$

$$S_{se} = -\psi_{se}(\mathbf{k}) \left[\frac{\beta_u(\mathbf{k})}{\alpha_u(\mathbf{k})} - 1 \right] \frac{c_g(\mathbf{k})}{\Delta L} N \ , \tag{2.181}$$

where S_{ld} and S_{se} are the local dissipation and the shadow effect, N is the spectral density, $\mathbf{k}$ is the wave vector, c_g is the group velocity, ΔL is the path length of the spectral component in the cell, and the ψ factors model the reduction of the dissipation in presence of local wave growth. The subscripts l and u of α and β indicate that these coefficients can be referred, respectively, to the cell and to the upstream polygon. For a more detailed explanation on the theoretical framework of UOST, the reader is referred to (Mentaschi et al., 2015b, 2018a).

Automatic generation of mesh parameters. An open-source python package (alphaBetaLab, https://github.com/menta78/alphaBetaLab) was developed for the automatic estimation, from real-world bathymetry, of the upstream polygons, of the transparency coefficients α_l, β_l, α_u, β_u, and of the other parameters needed by UOST. alphaBetaLab considers the cells as free polygons, and estimates the transparency coefficients from the cross section of the unresolved obstacles versus the incident spectral component (figure 2.1cd). This involves, that it can be applied to any type of mesh, including unstructured triangular and SMC meshes (as of August 2018 only regular and triangular meshes are handled, but support for SMC meshes will be soon added). We need to mention that while UOST would be able to modulate the energy dissipation with the spectral frequency, only the direction is currently considered in alphaBetaLab. For more details on the algorithms implemented in alphaBetaLab, the user is referred to Mentaschi et al. (2018a). Mentaschi et al. (2018c) provides the documentation of the software and of its architecture, along with use guidance and illustrative examples.

Time step settings. In WAVEWATCH III the source terms are applied at the end of each global time step. Therefore, to work properly on a given cell, UOST needs a global time step lower or equal to the critical CFL

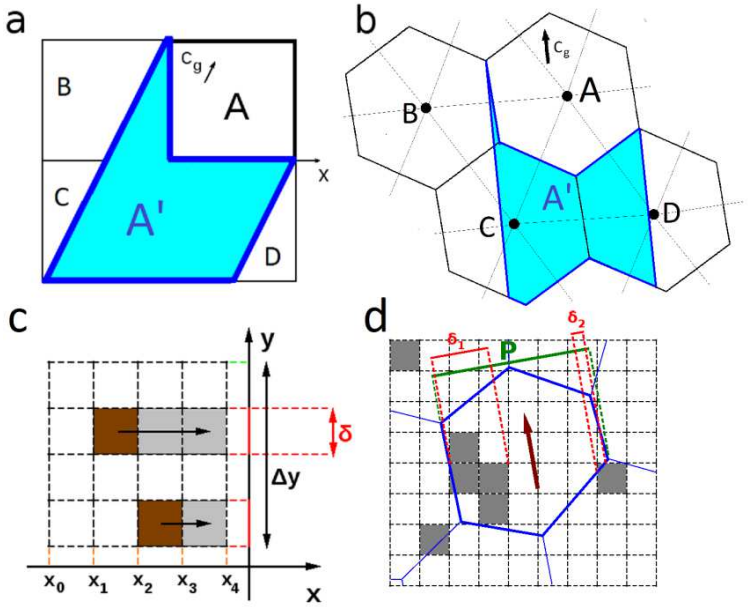

Figure 2.1: a: a square cell (A) and its upstream polygon (A', delimited by blue line, in light blue color) for a spectral component propagating with group velocity c_g. The joint BCD polygon represents the neighborhood polygon. b: same as a, but for a triangular mesh (the hexagons approximate the median dual cells). c: Computation of α and β for a square cell, $N_s = 4$, and a spectral component propagating along the x axis. d: Like c, but for a hexagonal cell and for a tilted spectral component. In panel d the gray squares represent unresolved obstacles.

Namelist parameter	Description	default value
UOSTFILELOCAL	Local α/β input file path	obstructions_local.*gridname*.in
UOSTFILESHADOW	Shadow α/β input file path	obstructions_shadow.*gridname*.in
UOSTFACTORLOCAL	Calibration factor for local transparencies	1
UOSTFACTORSHADOW	Calibration factor for shadow transparencies	1

Table 2.11: UOST parameters, their description and default values.

time step of the cell, i.e., the amount of time needed by the fastest spectral component to entirely cross the cell. Otherwise, part of the energy will leak through the cell without being blocked (Mentaschi et al., 2018a).

In unstructured grids with cells of very different sizes, the application of UOST to all the cells, including the smallest ones, may come with an excedingly small global time step that would affect the economy of the model. To avoid this problem the user can set alphaBetaLab in order to neglect cells smaller than a user-defined threshold, and then set in WAVEWATCH III a global time step equal to the critical CFL time step related with that threshold (Mentaschi et al., 2018c).

2.3.21 S_{xx}: User defined

Switch:	`XXn`
Origination:	—
Provided by:	user

This slot is intended for a source term that is not yet classified in Eq. (2.16). Almost by definition, it cannot be provided here.

2.4 Source terms for wave-ice interactions

Wave-ice interaction processes have been the topic of many investigations. In general, wave-ice interactions require a description of the ice properties that usually include at least the ice concentration (fraction of ocean surface covered by ice), mean ice thickness, and maximum floe diameter. Indeed, the ice is often broken into pieces (the floes) that can have a wide variety of sizes, and these sizes strongly modify the dispersion and wave-ice interaction processes.

In the present version of WAVEWATCH III ®, the different options for treating the ice represent ongoing research. There are now five different versions of dissipation processes activated with the switches `IC1`, `IC2`, `IC3`, `IC4` and `IC5` that can be combined with two different versions of scattering effects `IS1` and `IS2`. The second scattering routine, because it was the only routine to use a maximum floe diameter, also contains an estimation of ice break-up and resulting maximum floe diameter and some dissipation due to hysteresis.

Some features of these switch selections require care in application. For example, other than ice concetration and thickness, the forcing fields are context-based: they take different meanings for different source terms.

At present it is not possible to combine dissipation parametrizations designed for frazil or pancake ice (`IC3` or `IC4`) with a parametrization designed for the ice pack, such as `IC2`. Further, all parameterizations are not yet fully integrated: for example the floe size is not yet taken into account in some modified dispersion relations that take into account the ice. We expect to have a more streamlined way of combining various processes in future versions of WAVEWATCH III, possibly using a maximum floe diameter to call one or the other routines.

In most cases, WAVEWATCH III now permits spatial and temporal variability of ice-related inputs, but in practice this information is usually available only for ice concentration (from satellite or model) and thickness (most often from a model). The variability of other parameters representing the nature of the sea ice is rarely available to the user.

Observational study of effects of sea ice on waves are challenging, and the modeling of these effects in and near the ice edge is particularly difficult, where the accuracy of the non-stationary and non-uniform input ice fields can be a primary limition on accuracy (Rogers et al., 2018a).

A number of important modeling studies use models other than WAVEWATCH III, e.g. Doble and Bidlot (2013). So far, the various options available in WAVEWATCH III and described in the following sections have been tested in real conditions in a handful of studies, including Li et al. (2015), Ardhuin et al. (2016), Wang et al. (2016), Rogers et al. (2016), Rogers et al. (2018a), Cheng et al. (2017), and Liu et al. (submitted).

Experimental routines for representation of the effect of ice on waves have been implemented using the switches `IC1`, `IC2`, `IC3`, etc. The first two implemented in WAVEWATCH III (by Rogers and Orzech (2013)), were `IC1` and the initial version of `IC2` which was based on the work by Liu and Mollo-Christensen (1988) and Liu et al. (1991). These effects can be presented in terms of a complex wavenumber

$$k = k_r + ik_i, \tag{2.182}$$

with the real part k_r representing impact of the sea ice on the physical wavelength and propagation speeds, producing effects analogous to shoaling and refraction by bathymetry, whereas the imaginary part of the complex wavenumber, k_i, is an exponential decay coefficient $k_i(x, y, t, \sigma)$ (depending on location, time and frequency, respectively), producing wave attenuation. The k_i is introduced as $S_{ice}/E = -2c_g k_i$, where S_{ice} is a source term (see also Komen et al. (1994), pg. 170). With the methods that provide k_r, e.g. `IC2`, `IC3`, and `IC5`, the sea ice effects require solution of a new dispersion relation.

The effect of sea ice on k_i is used for all source functions in WAVEWATCH III version 6: `IC1`, ..., `IC5`. The effect of sea ice on k_r has been implemented for `IC2` and `IC3`, but is exported for use in the rest of the code only for `IC3`, and this remains an experimental feature.

The ice source functions are scaled by ice concentration.

In the case of ice, up to five input parameters are allowed as non-stationary and non-uniform fields. These can be referred to generically as $C_{ice,1}$, $C_{ice,2}$, ..., $C_{ice,5}$. The meaning of the ice parameters will vary depending on which S_{ice} routine is selected. For example, in case of `IC1`, only one parameter is specified, $C_{ice,1}$. In some cases, e.g. `IC3` and `IC4`, there are options for taking the simpler approach of inputting the same variables as stationary and uniform, defined using namelist parameters.

The reader is referred to the regression tests ww3_tic1.1-3 and ww3_tic2.1 for examples of how to use the new ice source functions.

Use within ww3_shel*:* Non-stationary and non-uniform input ice parame-

ters are permitted. In the case where any of the ice and mud source functions are activated with the switches `IC1`, `IC2`, `IC3`, `BT8`, or `BT9`, ww3_shel will anticipate intructions for 8 fields (5 for ice, then 3 for mud). These are given prior to the "water levels" information. The new fields can optionally be specified as a homogeneous field using lines found near the end of ww3_shel.inp.

Use within ww3_multi *(New in WAVEWATCH III version 6):* Using the namelist method of providing instructions to ww3_multi, it is now possible to prescribe mud and ice coefficients as non-stationary and non-uniform fields in ww3_multi. Prior to version 6, this was only possible with ww3_shel. However, the new method of reading these instructions via ww3_multi.nml has not yet been tested by this author at time of writing.

Separation of the source terms: The source terms `IC1`,...,`IC5` predict dissipation of wave energy. The reflection and scattering of waves from sea ice (e.g. Wadhams (1975)) are not dissipation: they are conservative processes. They are treated separately in `IS1` and `IS2`.

2.4.1 S_{ice}: Damping by sea ice (simple)

Switch:	`IC1`
Origination:	WAVEWATCH III/NRL
Provided by:	E. Rogers and S. Zieger

The first implemented method (`IC1`) is for the user to specify $k_i(x,t)$, which is uniform in frequency space, $C_{ice,1} = k_i$. The parameters $C_{ice,2}$,...,$C_{ice,5}$ are not used. An example setting is $C_{ice,1} = 2 \times 10^{-5}$. Descriptions specific to `IC2` and `IC3` are given in following sections.

A full description of `IC1` can be found in Rogers and Orzech (2013) and a concise summary is given in Rogers et al. (2018a). This simple method was applied in Li et al. (2015) and is used as the modeling component of a model-data inversion procedure by Rogers et al. (2016), Rogers et al. (2018a), and Rogers et al. (2018b).

2.4.2 S_{ice}: Damping by sea ice (Liu et al.)

Switch:	`IC2`
Origination:	WAVEWATCH III/NRL
Provided by:	E. Rogers, S. Zieger, F. Ardhuin

This method for representing the dissipation of wave energy by wave-ice interaction is based on the papers by Liu and Mollo-Christensen (1988), Liu et al. (1991) and Ardhuin et al. (2015). The main input ice parameters is the ice thickness (in meters) that can vary spatially and temporally and is the forcing field $C_{ice,1}$.

This is a model for attenuation by a sea ice cover, derived on the assumption that dissipation is caused by friction in the boundary layer below the ice, with the ice modeled as a continuous thin elastic plate. The original form by Liu and Mollo-Christensen (1988) is activated by setting the `IC2` namelist SIC2 parameter `IC2DISPER = .TRUE.`. That form assumes that the boundary layer is always laminar but it uses an eddy viscosity ν that can vary spatially and is the forcing field $C_{ice,2}$.

`IC2` and `IS2` share the optional use of the Liu and Mollo-Christensen (1988) dispersion relation for unbroken ice,

$$\sigma^2 = \left(gk_{ice} + Bk_{ice}^5\right) / \left(1/\tanh(k_{ice}H) + \frac{\rho_{ice} h k_{ice}}{\rho_w}\right), \tag{2.183}$$

$$c_g = (g + (5 + 4k_{ice}M)Bk_{ice}^5)/(2\sigma(1 + k_{ice}M)^2). \tag{2.184}$$

B and M quantify the effects of, respectively, ice bending due to waves and ice inertia. The group velocity under the ice, derived from the same relation, is used in the module `W3SIS2MD` and computed in `W3DISPMD`. See Liu and Mollo-Christensen (1988) for details.

For `IC2`, the dispersion relation is defined by

$$\sigma^2 = (gk_r + Bk_r^5)/(\coth(k_r h_w) + k_r M), \tag{2.185}$$

$$c_g = (g + (5 + 4k_r M)Bk_r^5)/(2\sigma(1 + k_r M)^2), \tag{2.186}$$

$$\alpha = (\sqrt{\nu\sigma}k_r)/(c_g\sqrt{2}(1 + k_r M)). \tag{2.187}$$

In our notation, h_w is water depth and h_i is ice thickness. The variables B and M quantify the effects of the bending of the ice and inertia of the ice, respectively. Both of these variables depend on h_i (see Liu and Mollo-Christensen, 1988; Liu et al., 1991).

This equation is only solved when ICEDISP=TRUE in the MISC namelist. Otherwise, $k_{ice} = k$, just like in open water. Note that the effect of k_{ice} is not passed back to the main program.

Stopa et al. (2016) improved `IC2` by adding a better alternative to the eddy viscosity representation of dissipation. They distinguish between laminar and turbulent regimes, allowing this is activated by setting `IC2DISPER = .FALSE.`. In that case the dissipation goes from a laminar form using the molecular viscosity multiplied by an empirical adjustment factor `IC2VISC` to a turbulent form, amplified by a factor `IC2TURB`, for Reynolds numbers above a user-defined threshold `IC2REYNOLDS`. This transition is smoothed over a range `IC2SMOOTH` to take into account the random nature of the wave field. In the turbulent regime, the friction factor is estimated from a user-specified under-ice roughness length `IC2ROUGH`, expected to be of the order of 10^{-4} m. Possibility has also been added to perform a heuristic reduction of the dissipation rate when the floe diameters are much shorter than the ocean wave wavelength (see Ardhuin et al., for details). It relies on the hypothesis that, in these conditions, ice floes follow at least partly the horizontal motion and it is thus logical to reduce the relative velocity between the ice and the water from which the dissipation rate is estimated. This reduction is controlled by the factor called r_D in Ardhuin et al. which value can be set through the namelist parameter `IC2DMAX`. Reduction of the dissipation rate occures for waves longer than $D_{\max}/r_D$ ($D_{\max}$ being the maximum floe size), and there is no reduction for maximum floe diameters of the order of the wavelength or more.

The parameter `IC2TURBS` is an ad hoc enhancement of turbulent dissipation in the Southern hemisphere that was introduced for test purposes to investigate sources of bias. This will be deprecated in future versions. It now appears that combining IC2 with ineleastic dissipation in IS2 can provide good results for dominant waves in both hemispheres (Ardhuin et al.).

A full description of the original form of `IC2` can be found in Rogers and Orzech (2013). It is applied in Li et al. (2015). A concise summary of the present code is given in Rogers et al. (2018a). A description of the improvement of the dissipation mechanism of `IC2` can be found in Appendix B of Stopa et al. (2016), where it is used.

2.4.3 S_{ice}: Damping by sea ice (Shen et al.)

Switch:	`IC3`
Origination:	Clarkson U. Fortran-77 code
Provided by:	E. Rogers, X. Zhao, S. Cheng, S. Zieger

The third method for representing wave-ice interactions is taken from Wang and Shen (2010). This model treats the ice as a visco-elastic layer. $C_{ice,1}$ is used for ice thickness (m); $C_{ice,2}$ is used for the viscosity ($m^2\,s^{-1}$); $C_{ice,3}$ is used for density ($kg\,m^{-3}$); $C_{ice,4}$ is used for effective shear modulus (Pa); $C_{ice,5}$ is not used. An example setting is $C_{ice,1...4} = [0.1, 1.0, 917.0, 0.0]$.

In WAVEWATCH III version 4, this method of S_{ice} (`IC3`) was much more expensive than `IC1` or `IC2`. This issue is largely addressed in model version 5.16.

The namelist SIC3 is introduced in model version 5.16. The namelist parameters are summarized in a list here, and some are discussed in further detail below.

`IC3CHENG`	Solution technique new in version 5.16. Default = `TRUE`.
`IC3HILIM`	Optional limiter on ice thickness. Default=100 (i.e. by default, the option is not used).
`IC3KILIM`	Optional limiter on dissipation rate k_i. Default=100 (i.e. by default, the option is not used).
`USECGICE`	When set to `TRUE`, the model will include the effect of ice on the group velocity. Default = `FALSE`.
`IC3VISC`	If user wishes to use an effective viscosity that is constant and uniform, this can now be done via namelist. Default=N/A.
`IC3ELAS`	As with `IC3VISC`, but for effective elasticity. Default=N/A.
`IC3DENS`	As with `IC3VISC`, but for ice density. Default=N/A.
`IC3HICE`	As with `IC3VISC`, but for ice thickness. Default=N/A.

IC3MAXCNC	Parameter which can be used to optionally switch to another dissipation for some ice conditions (see below). Default=100 (i.e. option is not used). Normal range is 0 to 1.
IC3MAXTHK	Idem. Default=100 (i.e. option is not used). Normal range is 0 to 10 meters.
IC2REYNOLDS	Parameter associated with IC2 non-dispersive turbulent boundary layer scheme. Default=1.5e+5.
IC2ROUGH	Idem. Default=0.02.
IC2SMOOTH	Idem. Default=7.0e+4.
IC2VISC	Idem. Default=2.0.
IC2TURB	Idem. Default=2.0.
IC2TURBS	Idem. Default=0.0.

The IC3CHENG option is new in model version 5. When set to TRUE, the model will use an alternative solution technique provided by S. Cheng. This has two important features. First, stability is improved, such that there is no need to use the ice limiter, i.e. the IC3HILIM parameter. Second, this method requires that three of four ice rheology parameters be stationary and uniform, input via namelist parameters (see below).

If IC3CHENG is set to FALSE, the user is advised to use the ice thickness limiter IC3HILIM to ensure stability (value of 25 to 100 cm is suggested). The parameter IC3KILIM was required for stable and fast computations in some prior development versions of WAVEWATCH III, but is now unnecessary and may be ignored by the user.

In model version 4.18, four ice rheology parameters (ice thickness, effective viscosity, effective elasticity, and ice density) were allowed to be non-stationary and non-uniform. This could be provided using ww3_prep. Or in cases where ww3_shel is used and non-uniform inputs are unnecessary, the "homogeneous" option of ww3_shel was available for rheology input. In model version 5.16, an option is added to specify the four ice rheology parameters via the namelist SIC3. Two restrictions apply: 1) If IC3CHENG is set to FALSE and USECGICE is set to TRUE, the namelist method cannot be used, and 2) If IC3CHENG is set to TRUE, the namelist method must be used for three of the rheology parameters (effective viscosity, effective elasticity, and ice density). If IC3CHENG is set to TRUE or USECGICE is set to FALSE, the fourth ice rheology parameter (ice thickness) can be input by either method (namelist or

non-namelist). The model performs error checking to ensure that the user has specified input for each parameter by a single method (neither method of input is assumed to supercede the other).

The k_r modified by ice is incorporated into the governing equation (2.8) via the c_g (group velocity) and c (phase velocity) calculations on the left-hand side; e.g. Rogers and Holland (2009, and subsequent unpublished work). The modification of wavenumber and group velocity can be optionally passed back to the main program to produce effects analogous to refraction and shoaling by bathymetry. However, this feature has been used so far only in academic studies, rather than in routine application to realistic scenarios.

To activate the shoaling effect, the model should be operated with namelist variable `USECGICE = TRUE`. To activate the refraction effect, the model should be compiled with switch `REFRX`. With this switch, the model computes refraction based on spatial gradients in phase velocity that include ice effects, rather than the simpler wave dispersion relation without ice. These effects are demonstrated in the regression test ww3_tic1.3 which is provided with the code.

The group velocity using `IC3CHENG` solver with zero ice thickness does not collapse exactly to that from the open water dispersion relation. This is caused by numerical error in the calculation $c_g = \frac{\partial\sigma}{\partial k} = \frac{\Delta\sigma}{\Delta k}$. These small differences in group speed will result in slight shoaling and refraction errors if these effects are turned on. Error for ice thickness equal to zero was found less than 10% and frequency dependent. This has been avoided by skipping the solver if ice thickness is exactly zero. If ice thickness is close to but not exactly zero, then this discrepancy may be noticeable. The solutions from CHENG for other parameters (effective viscosity, effective shear modulus) as they approach zero were not tested. Small but material difference have also noted between the solutions from `IC3CHENG` set to `FALSE` vs. `TRUE` for the same ice inputs.

As noted above, `USECGICE = TRUE` is required for the shoaling effect. However, since some ice rheology will lead to an increase in group velocity, the user is advised to use care with this option. The group velocity affects the CFL criterion, which may require that the user reduce the time step size. `USECGICE = FALSE` is recommended for users that do not wish to worry about this issue.

In model version 5.16, a non-default option is added which causes the dissipation parameterization to change for some ice conditions. If ice con-

centration exceeds `IC3MAXCNC` and ice thickness exceeds `IC3MAXTHK`, the `IC2` dissipation (more specifically, the non-default, non-dispersive boundary layer scheme sub-option of `IC2`) is used in place of the dispersion-based dissipation estimate of Wang and Shen (2010). See description of `IC2` for more information. Since it is non-dispersive, this feature should not be used with `USECGICE = TRUE`.

`IC3` first appeared in WAVEWATCH III ® version 4 and the first simple testing was reported in Rogers and Zieger (2014). It has since been used in Li et al. (2015), Wang et al. (2016), Rogers et al. (2016), and Cheng et al. (2017).

2.4.4 S_{ice}: Empirical/parametric damping by sea ice

Switch:	`IC4`
Origination:	WAVEWATCH III/NRL
Provided by:	C. Collins and E. Rogers

The fourth option (`IC4`) for damping of waves by sea ice was introduced by Collins and Rogers (2017). It gives methods to implement one of several simple, empirical/parametric forms for the dissipation of wave energy by sea ice. The motivation for `IC4` is to provide a simple, flexible, and efficient source term which reproduces, albeit in a highly parameterized way, some basic physics of wave-ice interaction. The method is set by the integer value (presently 1 to 7) for `IC4METHOD` namelist parameter: 1) an exponential fit to the field data of Wadhams et al. (1988), 2) the polynomial fit in Meylan et al. (2014), 3) a quadratic fit to the calculations of Kohout and Meylan (2008) given in Horvat and Tziperman (2015), 4) Eq. 1 of Kohout et al. (2014), 5) a simple step function with up to 4 steps (may be nonstationary and non-uniform), and 6) a simple step function with up to 10 steps (must be stationary and uniform), and 7) a formula from Doble et al. (2015) which uses ice thickness. All but the fourth method of `IC4` feature frequency-dependent attenuation. With the fourth method, attenuation varies with waveheight but is uniform in frequency space.

In the following discussion we use `IC4M1` to denote `IC4` method 1, and so forth. `IC4` appears in the switch and namelist `IC4METHOD=1` (for example) appears in the file ww3_grid.inp. Whereas in `IC1`, $C_{ice,1}$ is the user-determined

attenuation, for `IC4M1`, `IC4M2`, and `IC4M4` $C_{ice,n}$ are constants of the equations. For `IC4M3`, $C_{ice,1}$ is ice thickness. For `IC4M5`, $C_{ice,n}$ controls the step function. Note that $C_{ice,n}$ may be provided by the user as non-stationary and non-uniform using methods analogous to methods used to input water levels.

`IC4M1`: an exponential equation was chosen to fit the data contained in table 2 of Wadhams et al. (1988) which results in preferential attenuation of high frequency waves. This parameterizes the well-known low-pass filtering effect of ice. The equation has the following form:

$$\alpha = \exp\left[\frac{-2\pi C_{ice,1}}{\sigma} - C_{ice,2}\right] \tag{2.188}$$

Here, α is the exponential decay rate for energy, which is twice that for amplitude: $\alpha = 2k_i$. The values determined from the data are $C_{ice,1\ldots2} = [0.18, 7.3]$, but these may be modified by the user. This method is described and applied in Collins and Rogers (2017).

`IC4M2`: In this method, the dissipation is represented using a user-specified polynomial. It is a powerful method, since many shapes can be represented, e.g. by fitting to observation-based dissipation rates. The method is described and applied in Collins and Rogers (2017). The equation is the following:

$$\alpha = C_{ice,1} + C_{ice,2}\left[\frac{\sigma}{2\pi}\right] + C_{ice,3}\left[\frac{\sigma}{2\pi}\right]^2 + C_{ice,4}\left[\frac{\sigma}{2\pi}\right]^3 + C_{ice,5}\left[\frac{\sigma}{2\pi}\right]^4 \tag{2.189}$$

If a user wishes to follow Meylan et al. (2014), the suggested values for the coefficients are $C_{ice,1\ldots5} = [0, 0, 2.12\times10^{-3}, 0, 4.59\times10^{-2}]$. Additional suggested polynomials can be found in Rogers et al. (2018b).

With appropriate coefficients, this polynomial method can be used to reproduce the so-called roll-over effect where the attenuation is non-monotonic in frequency space. However, some recent studies do not indicate this effect, e.g. Rogers et al. (2016) and Li et al. (2017), and it may just be a spurious artifact in prior observational studies.

`IC4M3`: Horvat and Tziperman (2015) fit a quadratic equation to the attenuation coefficient calculated by Kohout and Meylan (2008) as a function of frequency, T, and ice thickness, h. Attenuation increases for thicker ice and higher frequencies (lower periods). The number of coefficients of the quadratic equation were prohibitively large to be user-determined, so the

equation is hardwired in and the tunable parameter, $C_{ice,1}$, is ice thickness h. This method is described and applied in Collins and Rogers (2017). For reference, the equation is the following:

$$\ln \alpha(T,h) = -0.3203 + 2.058h - 0.9375T - 0.4269h^2 + 0.1566hT + 0.0006T^2 \tag{2.190}$$

There are two warnings to make about `IC4M3`. First, the equation itself was an extrapolation of the original range of h used to calculate the attenuation coefficients in Kohout and Meylan (2008) which was between 0.5 and 3 m, see Horvat and Tziperman (2015). Second, in Kohout and Meylan (2008), wave attenuation predicted is based on scattering (a conservative process), whereas in `IC4M3`, the wave attenuation is treated as dissipation (non-conservative). This is ad hoc and not recommended for general use. Most especially, users should think twice before using `IC4M3` in combination with scattering routines `IS1` or `IS2`, since this is essentially double-counting scattering.

`IC4M4`: Kohout et al. (2014) found that attenuation was a function of significant wave height. Attenuation increased linearly with H_s until $H_s = 3$ m at which point attenuation is capped, thus:

$$\begin{cases} \frac{\partial H_s}{\partial dx} = C_{ice,1} \times H_s & \text{for } H_s \leq 3 \text{ m} \\ \frac{\partial H_s}{\partial dx} = C_{ice,2} & \text{for } H_s > 3 \text{ m} \end{cases} \tag{2.191}$$

where $k_i = \frac{\partial H_s}{\partial dx} / H_s$.

The values given in Kohout et al. (2014) are $C_{ice,1\ldots2} = [5.35 \times 10^{-6}, 16.05 \times 10^{-6}]$. See regression test ww3_tic1.1/input_IC4/M4 for examples. This method is described and applied in Collins and Rogers (2017).

`IC4M5`: This is a simple step function with up to 4 steps. It is controlled by the optionally nonstationary and non-uniform parameters $C_{ice,1\ldots7}$. Parameters $C_{ice,1\ldots4}$ control the step levels, which are in terms of dissipation rate, k_i. Parameters $C_{ice,5\ldots7}$ control the step boundaries (given in Hz). See regression test ww3_tic1.1/input_IC4/M5 for examples. This method is described in Collins and Rogers (2017).

`IC4M6`: This is a simple step function with up to 10 steps. It is controlled by the stationary and uniform namelist parameters `IC4KI` and `IC4FC`. Array `IC4KI` controls the step levels, which are in terms of dissipation rate, k_i, in radians per meter. Array `IC4FC` controls the step boundaries (given in Hz). See regression test ww3_tic1.1/input_IC4/M6 for examples.

`IC4M7`: This is a formula for dissipation from Doble et al. (2015), developed for a mixture of pancake and frazil ice, using data collected in the

Weddell Sea (Antarctica). The formula depends on wave frequency and ice thickness:

$$\alpha = 0.2T^{-2.13}h \quad . \tag{2.192}$$

This method is described in Rogers et al. (2018a).

2.4.5 S_{ice}: Damping by sea ice (Mosig et al.)

Switch:	`IC5`
Origination:	U. of Otago MATLAB code
Provided by:	Q. Liu, E. Rogers, A. Babanin

The fifth method for representing ice-induced wave decay is based on another viscoelastic-type model, i.e., the EFS ice layer model described in Mosig et al. (2015). The authors introduced viscosity into the thin elastic plate model of Fox and Squire (1994) and restricted it to one horizontal dimension (replacing a plate by a beam). The dispersion relation given by the EFS model can be written in the form

$$Qgk\tanh(kd) - \sigma^2 = 0, \tag{2.193}$$

$$Q = \frac{G_\eta h_i^3}{6\rho_w g}(1+\nu)k^4 - \frac{\rho_i h_i \sigma^2}{\rho_w g} + 1. \tag{2.194}$$

In Eq. (2.193)−(2.194), $G_\eta = G - i\sigma\rho_i\eta$ is the complex shear modulus, where G is the *effective* elastic shear modulus and η is the *effective* viscosity; ρ_w (ρ_i) is the density of water (ice), d is water depth, h_i is the ice cover thickness, σ is the radian frequency, $k = k_r + ik_i$ is the complex wavenumber, g is the gravitational acceleration and $\nu = 0.3$ refers to the Poisson ratio of sea ice.

Same as `IC3`, `IC5` requires four ice parameters as input: $C_{ice,1}$ for ice thickness h_i (m), $C_{ice,2}$ for the *effective* viscosity η (m^2 s^{-1}), $C_{ice,3}$ for ice density ρ_i (kg m^{-3}) and $C_{ice,4}$ for the *effective* shear modulus G (Pa). For example, as shown in Mosig et al. (2015, see their Fig. 8), a setting of $C_{ice,1,\ldots,4}$ = [1.0 m, 5.0×10^7 m^2 s^{-1}, 917.0 kg m^{-3}, 4.9×10^{12} Pa] (with a water depth d of 4300 m) can be used to fit the observed wave attenuation rates reported in Meylan et al. (2014). The application of the EFS model to two realistic case studies is presented in Liu et al. (submitted).

The dispersion relation shown above is solved iteratively using the Newton-Raphson method. The numerical solver, however, may fail for small

wave periods in some rare cases (particularly for shallow water depth d and low G). In such cases, the estimated wavelength k_r is unreasonably low. Several namelist variables (limiters) are introduced to improve the code stability:

`IC5MINIG`	the minimum allowed shear modulus G; Default= 1 Pa (i.e., zero G is not allowed).
`IC5MINWT`	the minimum allowed wave periods T; Default=0 s (i.e., by default, this option is not used).
`IC5MAXKRATIO`	the maximum allowed k_{ow}/k_r, where k_{ow} is the open-water wavenumber; Default=1E9 (i.e., by default, this option is not used).
`IC5MAXKI`	the maximum allowed k_i; Default=100 m^{-1} (i.e., by default, this option is not used).
`IC5MINHW`	the minimum allowed water depth d; Default=300 m (this basically limits `IC5` to the deep-water case).
`IC5MAXITER`	the maximum allowed # of iteration; Default=100.

Note that the EFS model used here regards the ice cover as a continuous homogeneous medium and characterizes various ice types with two *totally empirical* rheological parameters, namely the elastic shear modulus G and the viscosity η. As argued in Mosig et al. (2015), these two parameters *"cannot be measured directly as they do not represent observable physical processes"*. Therefore, "no restrictions on the acceptable values of the rheological parameters, *except positiveness*, can be imposed." So strictly speaking, the EFS model is better termed as an *effective medium* model rather than a *viscoelastic* model.

2.4.6 S_{is}: Diffusive scattering by sea ice (simple)

Switch:	`IS1`
Origination:	WAVEWATCH III/NRL
Provided by:	S. Zieger

The non-conservative effect of ice on waves has been implemented in switches `IC1` through `IC3` (see Section 2.4.1–2.4.3). The conservative effect of sea

ice has been implemented in switch `IS1` and represents a simple form of scattering. It is assumed that the floe size is smaller than the grid size and that a fraction α_{ice} of the incoming wave energy is scattered isotropically. The fraction is determined from sea ice concentration ICE using a simple linear transfer function

$$\alpha_{ice} = \max\{0, C_1 \text{ ICE} + C_2\} \quad . \tag{2.195}$$

The coefficients C_1 and C_2 are customizable through namelist SIS1 with namelist parameters `ISC1` and `ISC2`. At each discrete frequency and direction the wave energy is reduced by the amount of α_{ice} and redistributed to all direction in the same discrete frequency to conserve energy.

2.4.7 S_{is}: Floe-size dependent scattering and dissipation

Switch:	`IS2`
Origination:	WAVEWATCH III
Provided by:	F. Ardhuin, C. Sevigny, G. Boutin, D. Dumont, T. Williams

The implementation of this scattering term generally follows the approach of Meylan and Masson (2006), to which has been added an estimation of the breakup of the ice by waves to be able to update a maximum floe size diameter. Finally a creep-based dissipation was also combined with the scattering.

The scattering source term is defined by the scattering coefficient $\beta_{\text{is,MIZ}}(\theta - \theta')$ which by default is isotropic, but can be given any directional dependency by setting `IS2ISOSCAT=FALSE` in namelist SIS2. The source term is thus,

$$\frac{S_{\text{is}}(k,\theta)}{\sigma} = \int_0^{2\pi} \beta_{\text{is,MIZ}}(\theta - \theta')[s_{\text{scat}} N(k,\theta') - N(k,\theta)]d\theta'. \tag{2.196}$$

where s_{scat} is set to 1.0 by default but can be modified by `IS2BACKSCAT` in namelist SIS2.

The determination of scattering coefficients $\beta_{\text{is,MIZ}}$ is based on the theoretical reflection coefficient $\alpha_n(\sigma, h)$ for waves with a normal incidence going from a half-plane of open water to a half-plane of ice-covered water with a constant ice thickness h. Values of $\alpha_n(\sigma, h)$ as computed by Kohout and Meylan

(2008) (if `IS2WIM1=0.`) or by Bennetts and Squire (2012) (if `IS2WIM1=1.`) are tabulated in the `W3SIS2MD` module. Following Dumont et al. (2011), the broken ice is treated as a series of such ice-water interfaces. Neglecting multiple reflections, the scattering parameterization defines the attenuation per unit time as if the ice-covered part of a grid cell was a succession of floes of mean diameter D_m with a partial reflection $\alpha_n(\sigma, h)$ for each floe, giving,

$$\beta_{\mathrm{is,MIZ}} = c_i \; c_g \alpha_n(\sigma, h)/D_m, \tag{2.197}$$

where c_i is the ice concentration.

The estimation of the mean floe diameter D_m is based on an assumed power law for the number of floes of diameter D, taken proportional to $D^{-\gamma}$. This power law is further assumed to apply for D ranging from the minimum $D_{\min}$ and a maximum $D_{\max}$. The average is thus given by

$$D_m = \frac{\gamma}{\gamma - 1} \times \frac{D_{\max}^{-\gamma+1} - D_{\min}^{-\gamma+1}}{D_{\max}^{-\gamma} - D_{\min}^{-\gamma}}. \tag{2.198}$$

At present $D_{\min}$ is a user-supplied value. $D_{\max}$ can either be provided as a forcing field, e.g. from an ice model or some observations, or, if the namelist parameter `IS2BREAK` is set to `TRUE`, estimated from the breaking of the ice by the local wave field. If the namelist parameter `IS2DUPATE` is set to `TRUE`, small values of $D_{\max}$ will persist even if the waves become too small to be able to break the ice to that size. This is probably the proper model use when external forcing/coupling is available (e.g. advection of ice properties in an ice model). On the contrary, if `IS2DUPDATE` is set to `FALSE`, the value of $D_{\max}$ will be always adjusted to the local sea state, even if that means increasing $D_{\max}$.

Ice breaking by waves of wavelength λ is assumed to produce floes of diameter $\lambda/2$. In the parametrization, ice breaking occurs if the three following criteria are fulfilled (Williams et al., 2013):

1. $\lambda/2 \geq D_{\min}$ and $\lambda/2 \leq D_{\max}$
2. $D_{\max} > D_c$, as it exists a critical diameter, which depends on ice properties, below which no flexural failure is possible
3. $\varepsilon > \varepsilon_c$, the strain due to the incoming wave has to be greater than a defined critical strain

The first criterion is simply checking that the new value of $D_{\max}$ will be larger than $D_{\min}$ and smaller than the previous value of $D_{\max}$.

The second criterion relies on Mellor (1986) with the correction specified in Boutin et al. (2018), where D_c is defined as

$$D_c = \frac{1}{2}\left(\frac{\pi^4 Y^* h^3}{48\rho g(1-\nu^2)}\right)^{1/4}. \tag{2.199}$$

The third criterion corresponds to the flexural strain threshold. The horizontal strain caused by waves is related to the curvature of the ice layer, which, in one dimension is $\varepsilon = 0.5h\partial^2\eta_{\text{ice}}/\partial x^2$. The strain variance is given

$$\langle\varepsilon^2\rangle = \left(\frac{h}{2}\right)^2 \int_{k_1}^{k_2} k_{ice}^4 F(k)dk, \tag{2.200}$$

where h is the ice thickness and k_{ice} is the wavenumber $2\pi/\lambda_{ice}$. Borrowing from wave breaking ideas (Banner et al., 2000), the integration of the curvature variance is limited around the local wavenumber k_{ice}. We also note that we have defined an effective minimum ice thickness $h_{\min}$ so that, if $h < h_{\min}$, the strain variance is computed with $h = h_{\min}$ to avoid unbreakable elastic thin ice in the model that does not correspond to usual observations. We thus take $D_{\max}$ to be half the wavelength of the shortest waves for which the following criterion is met

$$F_{break}\sqrt{\varepsilon^2} > \frac{\sigma_c}{Y^*}, \tag{2.201}$$

where σ_c is the ice flexural strength. F_{break} is a factor representing random waves and adjustable with the SIS2 namelist parameter `IS2BREAKF`. It should in theory depend on the duration for which the ice is forced by the waves, and, based on the typical maximum value over 500 Rayleigh-distributes waves, was taken to be $F_{break} = 3.6$. $F_{break}\,E_s$ is thus the maximum strain for random waves.

Inelastic dissipation was added in this routine, following Wadhams (1973), because it critically depends on the floe size. It assumes that the floes deformation is not fully elastic, and that the secondary creep under the wave-induced cyclic causes the dissipation of wave energy into heat. We use the ice floe law

$$\left(\frac{d\varepsilon}{dt}\right)_{ij} = \frac{\tau^2}{B^3}\sigma'_{i,j}, \tag{2.202}$$

B is the floe law constant and is a function of ice temperature. Using the normalized parameter estimated by Cole et al. (1998) from laboratory experiments, $A = 10^{11}$, and a uniform ice temperature of 270 K gives a value of $B = 10^7$ s$^{1/3}$. The volumic dissipation rate is

$$\frac{de}{dt} = |\sigma_{xx}^4/(2B)^3|. \tag{2.203}$$

Also, the cyclic deformation of the ice can require a much larger elastic energy than the gravity potential energy, but this is only true if the ice is not broken. As a result, working with a wave elevation spectrum $E(k)$ could introduce large changes in $E(k)$ when the ice is broken or reformed. Instead we prefer to work with an energy spectrum $RC_gE(k)/C_{g,\mathrm{ice}}$, using the coefficient R introduced by Wadhams (1973), which is the ratio of elastic to gravity potential energies. For unbroken ice R is

$$R = 1 + C_R \frac{4Y^* h^3 \pi^4}{3\rho g \lambda^4 (1-\nu^2)}, \tag{2.204}$$

where we have been careful that Wadhams (1973) used $2h$ for the ice thickness, and C_R is by default set to 1.0 using the namelist parameter `IS2BREAKE`, but it can be set to zero to work with the true elevation spectrum instead. This factor R is also applied in the calculation of ice breakup by the waves.

The inelastic dissipation is linearized as $S_{\mathrm{ine}} = -\alpha_{\mathrm{ine}} E_{ice}$. The coefficient α_{creep} was adapted from the Wadhams (1973) monochromatic formula and is equal to

$$\alpha_{\mathrm{ine}} = 0.05 B h^5 \left(\frac{Y^*}{2B(1-\nu^2)} \right)^4 I_3 k^4 \frac{C_g^2}{\rho g C_{g_{ice}} R^2} F_{\mathrm{broken}} \int_{k_1}^{k_2} k_{ice}^4 E(k) dk, \tag{2.205}$$

where $I_3 = \frac{1}{\pi} \int_0^\pi \sin^4 \beta d\beta$. Details of the computation are given in Boutin et al. (2018). F_{broken} is a heuristic smooth transition from unbroken to broken ice, so that the dissipation gradually goes to zero for waves much longer than the floe sizes, because in that case the ice does not deform and produces no dissipation of wave energy,

$$F_{\mathrm{broken}} = \tanh \left(\frac{D_{\max} - C_\lambda \lambda_{ice}}{D_{\max} C_{smooth}} \right). \tag{2.206}$$

Inelastic dissipation is computed after updating $D_{\max}$. The two parameters in this smooth transition C_λ and C_{smooth} are set to 0.4 and 0.2 by the adjustable namelist parameters `IS2CREEPD` and `IS2CREEPC`.

Possibility of subsituting this inelastic dissipation by an anelastic dissipation term has also been added. It is activated by setting the namelist parameter `IS2ANDISB` to `TRUE`. Anelastic dissipation corresponds to the energy dissipated into heat during the oscillatory motion of the dislocations induced by the cyclic stress associated to waves. This behaviour results in a hysteresis that can be seen in stress-strain diagrams when sea ice is submitted to a sinusoidal stress as presented in Fig. 4 of Cole et al. (1998) study. This latter article also suggests a model for the anelastic behaviour of sea ice, with a stress-strain relationship that enables to compute the area within the ellipse which results from the hysteresis. Similarly to what has been done for inelastic dissipation, anelastic dissipation is linearized as $S_{\text{ane}} = -\alpha_{\text{ane}} E_{ice}$. Boutin et al. (2018) have derived α_{ane} from Cole et al. (1998) model for a monochromatic stress. They obtained the following coefficient:

$$\alpha_{\text{ane}} = \frac{A}{6}\left(k_i^2 \frac{Y^*}{(1-\nu^2)\rho g G}\right)^2 h^3 \frac{C_g}{G C_{g,i}} F_{\text{broken}}, \tag{2.207}$$

where F_{broken} is the same as for inelastic dissipation and A is equal to:

$$A = \frac{4}{3}\sigma\alpha_d \; \delta D^d \; \frac{1}{\exp(\alpha_d s) + \exp(-\alpha_d s)}, \tag{2.208}$$

in which terms are detailed in the table at the end of this section.

Finally we recall the various model parameters used in IS2 in the following table. Some are defined as constants in the `W3IS2MD` module, others can be adjusted with the SIS2 namelist.

Parameters	Symbol	namelist parameter	default values
Minimum floe size	$D_{\min}$	N. A.	20 m
Initial floe size	D_{init}	N. A.	1000 m
Ice fragility	ξ	N. A.	0.9
Ice density	ρ_{ice}	N. A.	922.5 kg m^{-3}
Effective Young Modulus	Y^*	N. A.	5.49 GPa
Poisson Coefficient	ν	N. A.	0.3
Flexural strength	σ_c	N. A.	0.27 MPa
Flow law parameter	n	`IS2CREEPN`	3
Flow law parameter	B	`IS2CREEPB`	10^7 s$^{1/3}$
Elastic energy correction	C_R	`IS2BREAKE`	1.0
Relax. of disloc. compliance	δD^d	`IS2ANDISD`	$\Delta_d \Omega b^2/K$ (Pa^{-1})
Dislocation density	Δ_d	N. A.	1.8×10^9 (m^{-2})
Restoring stress term	K	N. A.	0.07 (Pa)
Orientation factor	Ω	N. A.	π^{-1}
Burgers vector	b	N. A.	4.52×10^{-10} (m)
Drag term	B	N. A.	$B_0 \exp(Q_v/(k_b T_K)$ (Pa.s)
-	B_0	N. A.	1.205×10^{-9} (Pa.s)
Boltzmann constant	k_B	N. A.	8.617×10^{-5} (eV.K^{-1})
Activation energy	Q_v	`IS2ANDISE`	0.55 (eV)
Peak broadening term	α_d	N. A.	0.54
Reduced variable	s	N. A.	$\log(\tau\omega)$
Relax. time of disloc.	τ	N. A.	B/K (s)
Temperature of sea ice	T_K	N. A.	268.15 (K)

2.4.8 S_{ref}: Energy reflection at shorelines and icebergs

Switch:	`REF1`
Origination:	WAVEWATCH III
Provided by:	F. Ardhuin

Reflections by shorelines and icebergs is activated by using the `REF1` switch and setting namelists parameters `REFCOAST`, `REFSUBGRID` or `REFBERG` (in namelist REF1) to non-zero values that are the target reflection coefficients R_0^2 for the wave energy. If the `IG1` switch is also used, then the energy source at the shoreline also includes free infragravity waves in both ingoing and

outgoing directions. That particular source is described in section 2.4.9.

From these values R_0^2 may be varied with wave height and period following a Miche-type parameter: this is activated by setting `REFFREQ` to a non-zero value, and is based on the field measurements of Elgar et al. (1994). These coefficients can also be made to vary spatially, by setting `REFMAP` to a non-zero value. In that case ww3_grid will expect to find a extra line after the reading of the water depths and obstructions in ww3_grid.inp, which will define the map of shoreline slopes. The values in this map are multiplied by the value `REFMAP`.

Wave reflection at the shoreline varies from a fraction of a percent to about 50% of the incoming wave energy, and may have important consequences for the directional wave spectrum, and the wave climate in otherwise sheltered locations (O'Reilly et al., 1999). Wave reflection is also extremely important for the generation of seismic noise by ocean waves.

Because reflection involve wave trains with different directions, in a model like WAVEWATCH III, their interaction can only be represented through a source term in the right hand side. Nevertheless, this is physically linked to propagation.

In practice, for the regular and curvilinear grids, the reflection source term puts into the reflected wave directions the proper amount of energy that will be taken away by propagation at the next time step. When neglecting the cross-shore current, this is

$$\mathcal{S}_{ref}(k,\theta) = \int R^2(k,\theta,\theta') \frac{C_g(k)}{\Delta A} \left[\cos(\theta-\theta_q)\Delta q + \sin(\theta-\theta_p)\Delta p\right] N(k,\theta') \mathrm{d}\theta' \;, \tag{2.209}$$

where R^2 is an energy reflection coefficient, and Δp and Δq are the grid spacing along the two axes of the grid, and ΔA is the cell area. The definition of the shoreline direction from the land/sea mask is explained in Ardhuin et al. (2011b). This has not been tested for the SMC grids, and it is not expected to work for that type of grid.

In the case of unstructured grids, the spectral density of outgoing directions on the boundary is directly set to the expected reflected value and the boundary condition is handled specifically by the the numerical schemes.

The reflection coefficient R^2 is taken to be non-zero only for the directions for which $\cos(\theta-\theta') < 0$, and its magnitude is the product of a reflection coefficient $R_0^2(k)$, integrated over the scattered directions θ, and a directional distribution $R_2(\theta,\theta')$ around the specular direction θ_s,

$$R^2(k,\theta,\theta') = R_0^2(k) R_2(\theta,\theta') \quad . \tag{2.210}$$

This directional distribution takes three forms:

- isotropic in all directions opposite to the incoming direction: this is for sub-grid islands and icebergs or sharp shoreline angles,
- proportional to $\cos(\theta - \theta_s)^2$ for moderate shoreline angles,
- proportional to $\cos(\theta - \theta_s)^n$ for small shoreline angles (nearly straight shoreline). Where $n = 4$ by default and can be changed to any value using the `REFCOSP_STRAIGHT` namelist parameter in the REF1 namelist.

That parameterization is described in detail by Ardhuin and Roland (2012).

In the case of icebergs and sub-grid islands, the reflected energy is redistributed evenly in all directions within 90° of the direction opposite to the incoming waves. For resolved lands, a mean direction perpendicular to shore θ_n was defined from the land or sea status of the 8 grid points surrounding the local point (Fig. 2.2).

For each model grid point adjacent to land, the analysis of the land-sea geometry gives one value of θ_n among 16 possible directions. Together with any incoming wave direction θ_i this defines a specular reflection direction $\theta_r = 2\theta_n - \theta_i + \pi$. For each spectral component of direction θ_i going towards the coast (i.e. such that $\cos(\theta_i - \theta_n) > 0$), the total reflection is R^2 times the incoming energy. This reflected energy $R^2 E(f) M(f,\theta_i)$ is redistributed over directions around the specular reflection direction θ_r, with a broad distribution taken proportional to $\cos^n(\theta - \theta_r)$, where the power n is a function of the local shoreline geometry.

For this purpose we distinguish three different shoreline geometries relative to the local point as illustrated by Fig. 2.2: we set $n = 2$ for a straight coast (three connected land points among the neighbors), $n = 1$ for a mild corner (two land points among the neighbors), and $n = 0$ at a sharp corner (only one land point, among the 4 closest neighbors) which corresponds to the same treatment done for sub-grid islands and icebergs. Changing these values of n in the range 0 to 2 has little effect on our results. $n = 1$ corresponds to a Lambertian surface approximation, which is used for electromagnetic wave scattering from rough surfaces. A pure specular reflection would be obtained with n infinite. A more rigorous treatment should use the distribution of the

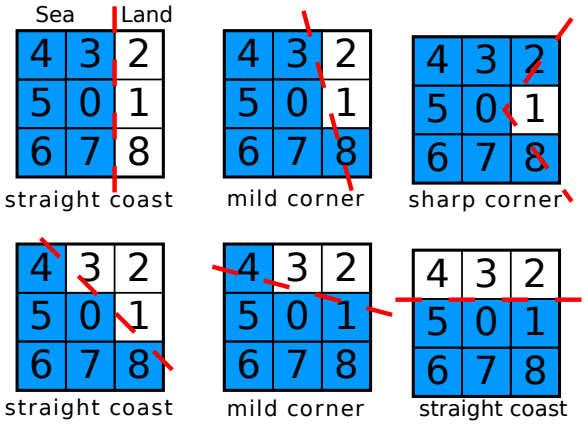

Figure 2.2: Examples of determination of the shoreline orientation and geometry using the land/sea mask. For any sea point (number 0) which is the ocean (in blue) and has at least one neighbor in land (in white) the eight neighbors, numbered from 1 to 8 are used to define the shoreline geometry. For 'mild' corners and straight coasts, the estimated shoreline orientation (dashed line) is used to compute the directional distribution of the reflected wave energy.

shoreline orientation at at the scale of the ocean wavelength, namely of the order of 100 m.

2.4.9 Second-order spectrum and free infragravity waves

Switch:	`IG1`
Origination:	WAVEWATCH III
Provided by:	F. Ardhuin

WARNING: *A bug has been identified with IG wave sources in unstructured grids. A model patch using an older version of the code will be provided shortly.*

The linear dispersion relation used in section 2.1 is a good approximation for most of the wave energy but a significant part of the spectrum at high frequencies, with typical frequencies above three times the windsea wave peak (e.g. Leckler, 2013). In shallow water, another strongly nonlinear part of the spectrum is found at very low frequencies, which are called infragravity waves.

In the case of horizontally homogeneous conditions over a flat bottom, both low and high frequency non-linear components can be estimated from the linear wave spectrum, using perturbation theory (e.g. Hasselmann, 1962). Also, the non-linear evolution of a homogeneous wave field is better described in terms of this 'linearized spectrum'. It is thus practical to work with this 'linearized spectrum' and convert to the observable spectrum that contains non-linear components when post-processing the model results. One method to perform this transformation is a canonical transformation proposed by Krasitskii (1994). The properties of this transformation were further explored by Janssen (2009) and implemented for post-processing in the ECMWF version of the WAM model.

The code for the canonical transform written by P. Janssen was interfaced with WAVEWATCH III. Using the `IG1` switch and setting the parameter `IGADDOUTP = 2` in the SIG1 namelist, this canonical transformed, which conserves energy, will be used for the output point spectra. If `IGADDOUTP = 1`, then the second-order spectrum is added on top of the model spectrum using the theory (e.g. Hasselmann, 1962). That option does not conserve energy and is not consistent at high frequency because the quasi-linear term

in the second-order spectrum are ignored (Janssen, 2009).

However, when comparing to measurements, one should be aware that different measuring devices have different responses to the nonlinear part of the spectrum. In particular, surface-following buoys also linearize the spectrum, and the second-order pressure field is not related to the second-order elevation via the relations used for linear waves. The canonical transform is thus only applicable for wave gauges that measure elevation at a fixed location.

When the wave field is not homogeneous, the nonlinear properties of the waves lead to an exchange of energy between different modes. In shallow water this usually results in the transfer of energy to infragravity waves, that are released along shorelines and travel as free waves. The `IG1` switch allows the parameterization of that effect with several methods. These are very crude parameterizations compared to the full hydrodynamic solution that would require solving the bispectral evolution across the surf zone at a very high spatial resolution (e.g. Herbers and Burton, 1997). The default namelist settings correspond to the parameterization presented by Ardhuin et al. (2014), with minor adjustments in version 6.06. The power in the definition of $\widehat{E}_{IG}(f)$ was changed from $f^{1.5}$ to $f^{1.0}$, with an associated adjuspent of the constant fro 0.015 to 0.013.

In practice the free infragravity wave energy is added via the S_{ref} source term, by setting the `SIG1` namelist `IGSOURCE` to 1 or 2.

In the first method, activated with `IGSOURCE =1`, the second-order spectrum is computed using either the Hasselmann perturbation (`IGMETHOD = 1`) or the canonical transform (any other value of `IGMETHOD`) as described in Janssen (2009). This approach may lead to better directional distribution of IG wave energy but it is still being tested. The second method, activated with `IGSOURCE = 2`, and the free IG spectrum is given by the following expressions,

$$A_{IG} = H_s T_{m0,-2}^2, \tag{2.211}$$

$$\widehat{E}_{IG}(f) = 1.2\alpha_1^2 \frac{kg^2}{c_g 2\pi f} \frac{(A_{IG}/4)^2}{\Delta_f} \left[\min(1., 0.013\mathrm{Hz}/f)\right]^{1.0}, \tag{2.212}$$

$$\widehat{E}_{IG}(f,\theta) = \widehat{E}_{IG}(f)/(2\pi), \tag{2.213}$$

where the mean period is defined as $T_{m0,-2} = \sqrt{m_{-2}/m_0}$ with the moments

$$m_n = \int_{f_{min}\ \mathrm{Hz}}^{0.5\ \mathrm{Hz}} E(f) f^n \mathrm{d}f, \tag{2.214}$$

and the empirical coefficient α_1 is of the order of 10^{-3} s^{-1}, and is set by the `SIG1` namelist parameter `IGEMPIRICAL`. The minimum frequency f_{min} used to define $T_{m0,-2}$ is set by the namelist parameter `IGMAXFREQ` and it is also the maximum frequency of the IG band over which this source of energy is applied. Also, in this band the IG energy at the coast can be added on top of pre-existing energy, or the pre-existing energy can be reset to zero. That latter behavior is the default and controlled by `IGBCOVERWRITE = 1`. For other choices, (`IGBCOVERWRITE = 0`), the results are very sensitive to the maximum shoreline reflection coefficient allowed (`REFRMAX` parameter in namelist REF1).

Finally, IG energy can also be added for frequencies beyond f_{min}, this is the default behavior and it is activated by setting `IGSWELLMAX = TRUE`. For that part of the IG wave field, the IG wave source is now reduced by a factor 4 which is now hard-coded in w3ref1md.ftn. This should be adjusted together with the maximum reflection which is defined by the REF1 namelist parameter `REFRMAX`. In the present version, the option `IGSWELLMAX = TRUE` does not work well with unstructured grids. We thus advise to use `IGSWELLMAX = FALSE` for these grids, this will unfortunately lead to a spectral gap between the IG band and the swell-windsea band.

2.5 Air-sea processes

2.5.1 General concepts

Additional subroutines are provided within WAVEWATCH III for use as part of coupled ocean-wave or ocean-atmosphere systems. These subroutines are designed to compute additional quantities related to the surface wave field which are intended to be passed to external models (e.g. ocean models). The motivation for these subroutines is to allow the external model to include the impact of waves on quantities such as the wind stress and the upper ocean turbulence.

Sea-state dependent air-sea fluxes The air-sea momentum flux, or the total wind stress, is the sum of the momentum flux into both surface waves and subsurface currents. Coupled atmosphere-ocean models that do not consider the impact of the surface gravity wave field typically compute the total wind stress based on an empirical relationship between the wind speed and the wind stress (via a drag coefficient, C_d). The provided *FLD* subroutines allows the computation of the total wind stress based on the WAVEWATCH III wavenumber-direction spectrum for use in coupled numerical models.

To the leading order, the total wind stress is equal to the sum of the momentum flux into surface waves (form drag of surface waves) and the momentum flux directly into the subsurface currents (through viscous stress). The momentum flux into the waves may be expressed as an integral of the wave variance spectrum multiplied by the wave growth rate (momentum-uptake rate). A few assumptions are needed to calculate the wave form drag. First, the wave form drag is sensitive at the leading order to the level of the high frequency waves (or the spectral tail). This part of the wave spectrum contains a great deal of uncertainty within the wave model, and therefore may need to be separately parameterized for computing the wind stress. An assumption must therefore be made to parameterize the high frequency, which is not constrained by observational data and wind speeds above 15 m/s. Second, assumptions of the wave growth-rate function are needed since it has historically been parameterized from either the wind speed or the wind stress. In either case, empirical coefficients are needed within the growth-rate function based on wavelength and wave direction relative to the wind and/or stress. Third, there is feedback due to the wave form drag on the turbulence profile and the wind profile within the wave boundary layer (roughly the upper 10 meters above the air-sea interface). How important this feedback is on determining the wind stress and the mean wind profile is not entirely understood. Finally, the growth rate is known to be different over breaking and non-breaking waves. However, there are no simple methods for explicitly including the breaking wave impact within wind-stress calculation models. Therefore, no separation is made in either of the present *FLD* subroutines between breaking and non-breaking wave growth-rates.

The total air-sea momentum flux can be expressed (to the leading order) as:

$$\vec{\tau} = \vec{\tau}_\nu + \vec{\tau}_f, \tag{2.215}$$

where $\vec{\tau}_\nu$ is the viscous stress vector and $\vec{\tau}_f$ is the wave form drag. At the air-sea interface, the wave form drag can be computed as the contribution of the momentum flux into all waves:

$$\vec{\tau}_f = \rho_w \int_{k_{min}}^{k_{max}} \int_{-\pi}^{\pi} \beta_g(k,\theta)\sigma F(k,\theta) d\theta \vec{k} dk, \tag{2.216}$$

where ρ_w is the water density, k is the wavenumber, θ is the wave direction, σ is the angular frequency, $\beta_g(k,\theta)$ is the growth rate, $F(k,\theta)$ is the wave variance spectrum, and k_{min} and k_{max} are the minimum and maximum wavenumbers of contributing waves. The expression for the growth rate varies based on the theory applied in the model, and will be described separately for each theory in their following descriptions.

The spectral tail at wind speeds above 15 m/s is not well constrained observationally or theoretically. Therefore, the spectral tail level in the *FLD* subroutines has been empirically parameterized such that the mean drag coefficient corresponds to the standard bulk drag coefficient used within the modeling system. In this way, the mean value of the wind stress will not be modified by using any explicit sea state dependent wind stress formulation, but the stress will deviate from the mean based on the sea-state. It is assumed that the tail level is a function of a wind speed only and is independent of sea states.

2.5.2 Sea-state dependent τ: Reichl et al. 2014

Switch:	`FLD1`
Origination:	WAVEWATCH III
Provided by:	B. Reichl

Wind stress according to Reichl et al., 2014 In Reichl et al. (2014) the total stress is constant in height, but is decomposed into two components as a function of height as:

$$\vec{\tau} = \vec{\tau}_t(z) + \vec{\tau}_f(z), \tag{2.217}$$

where τ_t is the turbulent stress and is equal to the viscous stress very near the surface. The wave form stress can be expressed as:

$$\vec{\tau}_f(z) = \rho_w \int_{k_{min}}^{k=\delta/z} \int_{-\pi}^{\pi} \beta_g(k,\theta)\sigma F(k,\theta) d\theta \vec{k} dk, \tag{2.218}$$

that is, the wave form stress at height z is equal to the integration of the wave form stress at the surface for wavenumbers below $k = \delta/z$, where δ/k is the inner layer height (Hara and Belcher, 2004) for waves at a wavenumber k. This expression is derived by assuming that the wave-induced stress is significant from the surface up to the inner layer height, but is negligible further above. Since at the surface

$$\vec{\tau} = \vec{\tau}_\nu + \vec{\tau}_f(z=0) = \vec{\tau}_\nu + \rho_w \int_{k_{min}}^{k_{max}} \int_{-\pi}^{\pi} \beta_g(k,\theta)\sigma F(k,\theta) d\theta \vec{k} dk, \tag{2.219}$$

the turbulent stress at a height z can be expressed as:

$$\vec{\tau}_t(z) = \vec{\tau}_\nu + \rho_w \int_{k=\delta/z}^{k_{max}} \int_{-\pi}^{\pi} \beta_g(k,\theta)\sigma F(k,\theta) d\theta \vec{k} dk. \tag{2.220}$$

In this model it is assumed that the turbulent stress at the inner layer height $z = \delta/k$ determines the growth rate of waves at wavenumber k:

$$\beta_g(k,\theta) = c_\beta \sigma \frac{|\tau_t(z=\delta/k)|}{\rho_w c^2} \cos^2(\theta - \theta_\tau), \tag{2.221}$$

where θ_τ is the direction of the turbulent stress at the inner layer height. The turbulent stress at the inner layer height is used in place of the total wind stress because longer waves reduce the effective wind forcing on shorter waves (wave sheltering).

The growth rate coefficient c_β varies depending on the ratio of the wave phase speed to the local turbulent friction velocity (friction velocity at the inner layer height), $u_\star^l = \sqrt{\tau_t(z=\delta/k)/\rho_a)}$.

$$c_\beta = \begin{cases} 25 & : \cos(\theta - \theta_w) > 0 \quad : c/u_\star^l < 10 \\ 10 + 15\cos[\pi(c/u_\star - 10)/15] & : \quad : 10 \le c/u_\star^l < 25 \\ -5 & : \quad : 25 \le c/u_\star^l \\ -25 & : \cos(\theta - \theta_w) < 0 \end{cases} \tag{2.222}$$

The wind profile is explicitly calculated using the energy conservation constraint in the wave boundary layer. From the top of the viscous sublayer to the inner layer height of the shortest waves the wind shear is expressed as:

$$\frac{d\vec{u}}{\partial z} = \frac{\rho_a}{\kappa z}\left|\frac{\vec{\tau}_\nu}{\rho_a}\right|^{3/2} \frac{\vec{\tau}_\nu}{\vec{\tau}_\nu \cdot \vec{\tau}_{tot}} \quad \text{for} \quad z_\nu < z < \delta/k_l. \tag{2.223}$$

Between the inner layer height of the shortest waves and that of the longest waves the wind shear is expressed as:

$$\frac{d\vec{u}}{\partial z} = \left[\frac{\delta}{z^2}\tilde{F}_w\left(k=\frac{\delta}{z}\right) + \frac{\rho_a}{\kappa z}\left|\frac{\vec{\tau}_t(z)}{\rho_a}\right|^{3/2}\right] \times \frac{\vec{\tau}_t(z)}{\vec{\tau}_t(z)\cdot\vec{\tau}_{tot}} \quad \text{for} \quad \delta/k_l \le z, \tag{2.224}$$

where $\tilde{F}_w(k=\delta/z)$ is the energy uptake by surface waves:

$$\tilde{F}_w(k=\delta/z) = \rho_w \int_{-\pi}^{\pi} \beta_g(k,\theta) g F(k,\theta) k d\theta. \tag{2.225}$$

Finally, above the inner layer height of the longest waves the wave effect is negligible and the wind shear is aligned in the direction of the wind stress:

$$\frac{d\vec{u}}{dz} = \frac{u_\star}{\kappa z}\frac{\vec{\tau}_{tot}}{|\vec{\tau}_{tot}|}. \tag{2.226}$$

Note that when using the FLD1 switch, internal variables and output values of the viscous stress, friction velocity, surface roughness length and Charnock parameter are recalculated and overwritten.

2.5.3 Sea-state dependent τ: Donelan et al. 2012

Switch:	`FLD2`
Origination:	UMWM
Provided by:	B. Reichl

Wind stress according to Donelan et al., 2012 In Donelan et al. (2012) the growth rate parameter in Eq. (2.216) is expressed as:

$$\beta_g(k,\theta) = A_1\sigma\frac{\left[u_{\lambda/2}\cos(\theta-\theta_w)-c\right]\left|u_{\lambda/2}cos(\theta-\theta_w)-c\right|}{c^2}\frac{\rho_a}{\rho_w}, \tag{2.227}$$

$$A_1 = \begin{cases} 0.11, & : u_{\lambda/2}\cos\theta > c, \quad \text{for wind forced sea} \\ 0.01 & : 0 < u_{\lambda/2}\cos\theta < c, \quad \text{for swell faster than the wind} \\ 0.1 & : \cos\theta < 0, \quad \text{for swell opposing the wind} \end{cases} \tag{2.228}$$

where A_1 is the proportionality coefficient determined empirically (so that modeled wave spectra agree with field observations), $u_{\lambda/2}$ is the wind speed at the height of half the wavelength (up to 20 m), θ_w is the wind direction, and c is the wave phase speed. The wind speed is calculated using the law of the wall for rough surfaces:

$$u(z) = \frac{u_\star}{\kappa} \ln\left(\frac{z}{z_0}\right), \tag{2.229}$$

where κ is the von Kármán coefficient (default 0.4). The viscous stress is calculated from the law of the wall for smooth surfaces. The viscous drag coefficient, Cd_ν is adjusted to account for sheltering:

$$Cd'_\nu = \frac{Cd_\nu}{3}\left(1 + \frac{2Cd_\nu}{Cd_\nu + Cd_f}\right), \tag{2.230}$$

where Cd_f is the wave form drag coefficient. The viscous stress can then be solved for as:

$$\vec{\tau}_\nu = \rho_a Cd'_\nu \left|\mathbf{u_z}\right| \mathbf{u}_z. \tag{2.231}$$

Note that when using the FLD2 switch, internal variables and output values of the viscous stress, friction velocity, surface roughness length and Charnock parameter are recalculated and overwritten.

2.6 Output parameters

The wave model can output parameters on the geographical grid of the model that can be integrated parameters or parameters as a function of frequency, with one grid for each frequency. All parameters that are a function of frequency (e.g. **EF** or **USF**) require the setting of specific namelist parameters in the OUTS namelist defined in ww3_grid.inp or ww3_grid.nml . This is to reduce the memory use if these parameters are not needed.

Below, a brief definition of output field parameters is provided. A table with definitions may be found in the sample `ww3_shel.inp` file, in Section 4.4.10. That input file also provides a list of flags indicating if output parameters are available in different field output file types (ASCII, grib, igrads, NetCDF). For any details on how these parameters are computed, the user may read the code of the `w3iogo` routine, in the `w3iogomd.ftn` module.

Selection of field outputs in `ww3_shel.inp` is most easily performed by providing a list of the requested parameters, for example, **H**S DIR SPR will request the calculation of significant wave height, mean direction and directional spread. These will thus be stored in the `out_grd.XX` file and can be post-processed, for example in NetCDF using `ww3_ouf`. Examples are given in Section 4.4.12 and Section 4.4.15. The names for these namelists are the bold names below, for example **HS**.

All parameters listed below are available in ASCII and NetCDF output files. If selected output file types are grads or grib, some parameters may not be available. Availability (or not) is identified in the first two columns in the field output parameter table within the example input file in Section 4.4.10. That table also identifies, for all parameters, the internal WAVEWATCH III code tags, the output tags (names used is ASCII file extensions, NetCDF variable names and namelist-based selection (see also Section 4.4.15), and the long parameter name/definition.

Finally we note that in all definitions the frequency is the *relative* frequency. Thus, in the presence of currents, these can only be compared to drifting measurement data or data obtained in wavenumber and converted to frequency. Comparison to fixed instrument data requires the use of the full spectrum and proper conversion to the fixed reference frame.

I) Forcing fields

1) **DPT** The mean water depth (m). This includes varying water levels.
2) **CUR** The mean current velocity (vector, m/s).
3) **WND** The mean wind speed (vector, m/s). This wind speed is always the speed as input to the model, i.e., is not corrected for the current speed.
4) **AST** The air-sea temperature difference (°C).
5) **WLV** Water level.
6) **ICE** Ice concentration.
7) **IBG** Wave attenuation due to icebergs: this parameter is the inverse

of the e-folding scale associated to the loss of wave energy in a field of small icebergs (Ardhuin et al., 2011b).

8) **D50** Sediment median grain size (D_{50}).
9) **IC1** Ice thickness.
10) **IC5** Maximum ice flow diameter, $D_{\max}$.

II) Standard mean wave parameters

1) **HS** Significant wave height (m) [see Eq. (2.23)]

$$H_s = 4\sqrt{E} \; . \tag{2.232}$$

2) **LM** Mean wave length (m) [see Eq. (2.22)]

$$L_m = 2\pi\overline{k^{-1}} \; . \tag{2.233}$$

3) **T02** Mean wave period (s) [see Eq. (2.22)]

$$T_{m02} = 2\pi/\sqrt{\overline{\sigma^2}} \; . \tag{2.234}$$

4) **T0M1** Mean wave period (s) [see Eq. (2.22)]

$$T_{m0,-1} = 2\pi\overline{\sigma^{-1}} \; . \tag{2.235}$$

5) **T01** Mean wave period (s) [see Eq. (2.22)]

$$T_{m0,1} = 2\pi/\overline{\sigma} \; . \tag{2.236}$$

6) **FP** Peak frequency (Hz), calculated from the one-dimensional frequency spectrum using a parabolic fit around the discrete peak.
7) **DIR** Mean wave direction (degr., meteorological convention)

$$\theta_m = \mathrm{atan}\left(\frac{b}{a}\right) \; , \tag{2.237}$$

$$a = \int_0^{2\pi}\int_0^{\infty} \cos(\theta) F(\sigma,\theta) \, d\sigma \, d\theta \; , \tag{2.238}$$

$$b = \int_0^{2\pi}\int_0^{\infty} \sin(\theta) F(\sigma,\theta) \, d\sigma \, d\theta \; . \tag{2.239}$$

8) **SPR** Mean directional spread (degr.; Kuik et al., 1988)

$$\sigma_\theta = \left[2 \left\{ 1 - \left(\frac{a^2 + b^2}{E^2} \right)^{1/2} \right\} \right]^{1/2} , \tag{2.240}$$

9) **DP** Peak direction (degr.), defined like the mean direction, using the frequency/wavenumber bin containing of the spectrum $F(k)$ that contains the peak frequency only.
10) **HIG** Infragravity height.
11) **MXE** Max surface elev (Space-time extreme, STE)
12) **MXES** St Dev of max surface elev (STE)
13) **MXH** Max wave height (STE)
14) **MXHC** Max wave height from crest (STE)
15) **SDMH** St Dev of MXC (STE)
16) **SDMHC** St Dev of MXHC (STE)
17) **WBT** Dominant wave breaking probability b_T (2.147)

III) Spectral parameters (first 5 moments and wavenumbers).
All these parameters are a function of frequency and thus these are three-dimensional arrays. Because of the large memory use, the computation of these parameters requires the activation of switches in the OUTS namelist.

1) **EF** Wave frequency spectrum (m^2/Hz)

$$E(f) = 2\pi \int F(\sigma,\theta) \mathrm{d}\theta \, . \tag{2.241}$$

2) **TH1M** Mean direction for each frequency (degr.; Kuik et al., 1988)

$$\theta_1(f) = \mathrm{atan} \left(\frac{b_1(f)}{a_1(f)} \right) , \tag{2.242}$$

$$a_1(f) = 2\pi \int_0^{2\pi} \int_0^{\infty} \cos(\theta) F(\sigma,\theta) \, \mathrm{d}\theta \, , \tag{2.243}$$

$$b_1(f) = 2\pi \int_0^{2\pi} \int_0^{\infty} \sin(\theta) F(\sigma,\theta) \, \mathrm{d}\theta \, . \tag{2.244}$$

3) **STH1M** First directional spread for each frequency (degr.;)

$$\sigma_1(f) = \left[2 \left\{ 1 - \left(\frac{a_1(f)^2 + b_1(f)^2}{E(f)^2} \right)^{1/2} \right\} \right]^{1/2} , \tag{2.245}$$

4) **TH2M** Mean direction from a_2 and b_2 (degr.)

$$\theta_2(f) = \mathrm{atan} \left(\frac{b_2(f)}{a_2(f)} \right) , \tag{2.246}$$

$$a_2(f) = 2\pi \int_0^{2\pi} \int_0^{\infty} \cos(2\theta) F(\sigma,\theta) \, \mathrm{d}\theta , \tag{2.247}$$

$$b_2(f) = 2\pi \int_0^{2\pi} \int_0^{\infty} \sin(2\theta) F(\sigma,\theta) \, \mathrm{d}\theta . \tag{2.248}$$

5) **STH2M** Directional spreading from a_2 and b_2 (degr.)

$$\sigma_2(f) = \left[0.5 \left\{ 1 - \left(\frac{a_2(f)^2 + b_2(f)^2}{E(f)^2} \right)^{1/2} \right\} \right]^{1/2} , \tag{2.249}$$

6) **WN** Wavenumbers $k(\sigma)$ (rad/m)

$$\sigma^2 = gk \tanh(kD) , \tag{2.250}$$

IV) Spectral partition parameters

These output parameters are based on partitioning of the spectrum into individual wave fields.

1) **PHS** Wave heights H_s of partitions of the spectrum (see below).
2) **PTP** Peak (relative) periods of partitions of the spectrum (parabolic fit).
3) **PLP** Peak wave lengths of partitions of the spectrum (from peak period).
4) **PSP** Mean direction of partitions of the spectrum.
5) Directional spread of partition of the spectrum Cf. Eq. (2.240).
6) **PWS** Wind sea fraction of partition of the spectrum. The method of Hanson and Phillips (2001) is used, implemented as described

in Tracy et al. (2007). With this, a 'wind sea fraction' W is introduced

$$W = E^{-1}\, E|_{U_p>c} \quad , \tag{2.251}$$

where E is the total spectral energy, and $E|_{U_p>c}$ is the energy in the spectrum for which the projected wind speed U_p is larger than the local wave phase velocity $c = \sigma/k$. The latter defines an area in the spectrum under the direct influence of the wind. To allow for nonlinear interactions to shift this boundary to lower frequencies, and subsequently to have fully grown wind seas inside this are, U_p includes a multiplier C_{mult}

$$U_p = C_{mult} U_{10} \cos(\theta - \theta_w) \quad . \tag{2.252}$$

The multiplier can be set by the user. The default value is $C_{mult} = 1.7$.

7) **PDP** Peak direction of partitions of the spectrum.
8) **PQP** Goda peakedness of partitions of the spectrum.
9) **PPE** JONSWAP peak enhancement factor of partitions of the spectrum. It is a measure of amplification of the spectral peak to the peak of the corresponding Pierson and Moskowitz spectrum.
10) **PGW** Gaussian frequency width (Hz) of partitions of the spectrum. Least-square fit of a Gaussian spectrum to the one-dimensional spectral partition.
11) **PSW** Spectral width of partitions of the spectrum (Longuet-Higgins, 1984)
12) **PTE** Wave energy period of partitions of the spectrum [see Eq. (2.235)]
13) **PT1** Mean wave period of partitions of the spectrum [see Eq. (2.236)]
14) **PT2** Wave zero-upcrossing period of partitions of the spectrum [see Eq. (2.234)]
15) **PEP** Wave peak spectral density of partitions
16) **TWS** Wind sea fraction of the entire spectrum.
17) **PNR** Number of partitions found in the spectrum.

V) Atmosphere-waves layer

1) **UST** The friction velocity u_* (scalar). Definition depends on selected source term parameterization (m/s). An alternative vector

version of the stresses is available for research (requires user intervention in the code).

2) **CHA** Charnock parameter for air-sea friction (without dimensions)
3) **CGE** Energy flux (W/m)

$$C_g E = \rho_w g \overline{C_g} E \; . \tag{2.253}$$

4) **FAW** Wind to wave energy flux
5) **TAW** Net wave-supported stress (wind to wave momentum flux)
6) **TWA** Negative part of the wave-supported stress
7) **WCC** Wave to wind momentum flux
8) **WCF** Whitecap coverage (without dimensions)
9) **WCH** Whitecap mean thickness (m)
10) **WCM** Mean breaking wave height (m) (NOT AVAILABLE YET)

VI) Wave-ocean layer

1) **SXY** Radiation stresses

$$S_{xx} = \rho_w g \iint \left(n - 0.5 + n \cos^2\theta \right) \; F(k,\theta) \; dk d\theta \; , \tag{2.254}$$

$$S_{xy} = \rho_w g \iint n \sin\theta \cos\theta \; F(k,\theta) \; dk d\theta \; , \tag{2.255}$$

$$S_{yy} = \rho_w g \iint \left(n - 0.5 + n \sin^2\theta \right) \; F(k,\theta) \; dk d\theta \; , \tag{2.256}$$

where

$$n = \frac{1}{2} + \frac{kd}{\sinh 2kd} \; . \tag{2.257}$$

2) **TWO** Wave to ocean momentum flux
3) **BHD** Bernoulli head (m^2/s^2)

$$J = g \iint \frac{k}{\sinh 2kd} \; F(k,\theta) \; dk d\theta \; , \tag{2.258}$$

4) **FOC** Wave to ocean energy flux (W/m^2)
5) **TUS** Stokes volume transport (m^2/s)

$$(M_x^w, M_y^w) = g \iint \frac{(k\cos(\theta), k\sin(\theta))}{\sigma} \; F(k,\theta) \; dk d\theta \; , \tag{2.259}$$

6) **USS** Stokes drift at the sea surface (m/s)

$$(U_{ssx}, U_{ssy}) = \iint \sigma \cosh 2kd \frac{(k\cos(\theta), k\sin(\theta))}{\sinh^2 kd} F(k,\theta)\, dkd\theta \,, \tag{2.260}$$

7) **P2S** Second order pressure variance (m^2) and peak period of this pressure (s) which contributes to acoustic and seismic noise,

$$F_{p2D}(k=0) = \int_0^\infty \frac{4\sigma}{C_g} \int_0^\pi F(k,\theta)\, F(k,\theta+\pi)\, d\theta dk \,, \tag{2.261}$$

8) **USF** Frequency spectrum of Stokes drift at the sea surface (m/s/Hz)

$$(U_{ssx}(f), U_{ssy}(f)) = \int \sigma \cosh 2kd \frac{(k\cos(\theta), k\sin(\theta))}{\sinh^2 kd} F(k,\theta) \frac{2\pi}{C_g} d\theta \,, \tag{2.262}$$

Note that US3D=1 must be set in the ww3_grid.inp OUTS namelist in order to activate the 3D arrays needed for USF output.

9) **P2L** Frequency spectrum of the second order pressure (m^2s) which contributes to acoustic and seismic noise,

$$F_{p2D}(k=0,f) = \frac{2\sigma}{\pi} \int_0^\pi \frac{4\pi^2}{C_g^2} F(k,\theta)\, F(k,\theta+\pi)\, d\theta \,. \tag{2.263}$$

10) **TWI** Wave to sea ice stress
11) **FIC** Wave to sea ice energy flux
12) **USP** Surface Stokes drift partitioned into run-time defined frequency components (m/s),

$$(U_{ssp}, V_{ssp})(n) = \iint_{k_n} \sigma \cosh 2kd \frac{(k\ \cos(\theta), k\sin(\theta))}{\sinh^2 kd} F(k,\theta)\, dkd\theta \,. \tag{2.264}$$

The number n and the central wavenumbers for the partitions k_n are 'run-time' in the sense that they are defined in the model definition file generated from ww3_grid.inp (see the sample input file for details on setting these quantities). The wavenumber integral bounds for each n partition divide the model wave spectra to accomodate the prescribed values of k_n.

VII) Wave-bottom layer

1) **ABR** Near-bottom rms excursion amplitude

$$a_{b,rms} = \left[2 \iint \frac{1}{\sinh^2 kd} \, F(k,\theta) \, dkd\theta \right]^{1/2} . \tag{2.265}$$

2) **UBR** Near-bottom rms orbital velocity

$$u_{b,rms} = \left[2 \iint \frac{\sigma^2}{\sinh^2 kd} \, F(k,\theta) \, dkd\theta \right]^{1/2} . \tag{2.266}$$

3) **BED** Bedform parameters: ripple height and directions (NOT TESTED YET)
4) **FBB** Energy dissipation in WBBL
5) **TBB** Momentum loss in WBBL

VIII) Spectrum parameters

1) **MSS** Mean square slopes in u and c directions (down-wave and cross-wave components of slopes variances).
2) **MSC** Spectral tail level (without dimensions)
3) **MSD** Direction of the maximum slope variance mss_u
4) **MCD** Spectral tail direction
5) **QP** Peakedness parameter (Goda, 1970)

$$Q_p = \frac{2}{E^2} \int_0^{2\pi} \int_0^{\infty} \sigma \, F(\sigma,\theta)^2 \, d\sigma \, d\theta \tag{2.267}$$

IX) Numerical diagnostics

1) **DTD** Average time step in the source term integration (s).
2) **FC** Cut-off frequency f_c (Hz, depends on parameterization of input and dissipation).
3) **CFX** Maximum CFL number for spatial advection
4) **CFD** Maximum CFL number for angular advection
5) **CFK** Maximum CFL number for wavenumber advection

X) User defined

1) **U1** Slot for user defined parameter (requires modification of code).
2) **U2** Idem.

2.7 Derived parameters

2.7.1 Directional slopes and near-nadir backscatter

Under the linear wave assumption, the surface slopes are Gaussian and fully prescribed by the mean square slope tensor mss_x, mss_y, mss_{xy}. In WAVEWATCH III the computed mss parameters are the down-wave mss_u, which is in the direction given by mss_d, and a cross-wave mss_c which is in the perpendicular dimension.

As a a result, the mean square slope tensor for the (long) wave resolved by the wave model, after converting mss_d to radians, are given by

$$\mathrm{mss}_{x,\mathrm{long}} = \mathrm{mss}_u \cos^2(\mathrm{mss}_d) + \mathrm{mss}_c \sin^2(\mathrm{mss}_d) \quad (2.268)$$

$$\mathrm{mss}_{y,\mathrm{long}} = \mathrm{mss}_u \sin^2(\mathrm{mss}_d) + \mathrm{mss}_c \cos^2(\mathrm{mss}_d) \quad (2.269)$$

$$\mathrm{mss}_{xy,\mathrm{long}} = 0.5(\mathrm{mss}_u - \mathrm{mss}_c) \sin(2\mathrm{mss}_d) \quad (2.270)$$

The contribution of short waves (above the maximum frequency of the model) should be added to these for a comparison with observations such as the backscatter power (σ^0) of near-nadir optical or radar data (altimeters, GPM or CFOSAT/SWIM data).

2.7.2 Stokes drift profile

The spectrum of the surface Stokes drift has the two components that are computed as

```
usf(IK)= SUM( E(IK,ITH)*ECOS(ITH)*DTH(ITH)
)*DK(IK)*2*WN(IK,ISEA) * FACT(KD)/DF(IK)
   vsf(IK)= SUM( E(IK,ITH)*ESIN(ITH)*DTH(ITH)
)*DK(IK)*2*WN(IK,ISEA) * FACT(KD)/DF(IK)
```

Where WN is the wavenumber and FACT(KD) is a function of the non-dimensional water depth.

From this spectrum, the Stokes drift at any depth z, counted positive upwards from a reference $z = 0$, are given by

Us(z)= SUMIK=I1,I2 (usf(IK) * FACT2(z,KD)*DF(IK)) Vs(z)= SUMIK=I1,I2 (vsf(IK) * FACT2(z,KD)*DF(IK))

This requires a knowledge of the local water depth D and mean sea level LEV and the wavenumbers WN(IK). KD=D*WN(IK) is thus a local function if the frequency/wavenumber index IK.

The coefficent FACT2(z,KD) is equal to EXP(2*WN(IK)*(z-LEV)) if KD $>$ 6, and otherwise,
FACT2(z,KD)=COSH(2*WN(IK)*(z+H))/COSH(2*WN(IK)*D).

This page is intentionally left blank.

3 Numerical approaches

The Wave Action Equation in Cartesian (2.8) or spherical (2.12) coordinates is the basic equations of the wave model. However, modified versions of these equations are used in the model, where (a) they are solved on a variable wavenumber grid (see below), where (b) modified versions of these equations are used to properly describe dispersion for discretized equations in selected numerical schemes (see Section 3.4), and where (c) sub-grid obstacles such as islands are considered (see Section 3.4).

3.1 Spectral discretization

If Eq. (2.8) or 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 grid in relative frequency (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} and allow a good separation of swell frequencies, a frequency discretization with exponentially increasing increments is adopted, so that the varying frequency resolution is proportional to the local frequency,

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

where m is a discrete grid counter in k-space. X_σ is defined by the user in the input files of the program. Traditionally, in most applications of third-generation models $X_\sigma \simeq 1.1$ is used.

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

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

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

3.2 Splitting of the wave action equation

In WAVEWATCH III Eq. (3.2) is solved using a fractional step method. The first step treats the 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 off 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. Starting with version 5.10, the source term S is further split into non-ice $S_{no\ ice}$ and ice S_{ice} source term. For a single model grid, the following sequence of integration is performed by the `W3WAVE` routine:

1. Update of water level
2. Intra-spectral part 1: integration over $\Delta t_g/2$ of $\frac{\partial}{\partial t}\frac{N}{c_g} + \frac{\partial}{\partial \kappa}\frac{\dot{\kappa}N}{c_g} + \frac{\partial}{\partial \theta}\frac{\dot{\theta}N}{c_g} = 0$
3. Spatial propagation: integration over Δt_g of $\frac{\partial}{\partial t}\frac{N}{c_g} + \frac{\partial}{\partial x}\frac{\dot{x}N}{c_g} + \frac{\partial}{\partial y}\frac{\dot{y}N}{c_g} = 0$
4. Intra-spectral part 2: integration over $\Delta t_g/2$ of $\frac{\partial}{\partial t}\frac{N}{c_g} + \frac{\partial}{\partial \kappa}\frac{\dot{\kappa}N}{c_g} + \frac{\partial}{\partial \theta}\frac{\dot{\theta}N}{c_g} = 0$
5. Source term integration: integration over Δt_g of $\frac{\partial}{\partial t}\frac{N}{c_g} = \frac{S_{no\ ice}}{\sigma c_g}$
6. Ice source term integration: integration over Δt_g of $\frac{\partial}{\partial t}\frac{N}{c_g} = \frac{S_{ice}}{\sigma c_g}$

The succession of these 6 steps is, in the limit $\Delta t_g \to 0$, equivalent to the integration of Eq. (3.2) over a global time step Δt_g.

This splitting in multiple steps allows an efficient vectorization and parallelization 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. WAVEWATCH III makes a distinction between 4 different time steps.

1) The 'global' time step Δt_g, is the common step of all the splitted sub-integrations. In that sense, it is the smallest time step for which a physically meaningful solution can be obtained, because all terms in the equation have been integrated. As a result, this is a possible time step for evaluating model output or coupling with other models, and, in the case of a multi-grid system, it is the time step at which communication between grids is performed. In the case of a forced – not coupled – model, input winds and currents are interpolated at this global step. This time step is provided by the user in the input file of `ww3_grid`, 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. This is not used for triangular-based grids, for which the advection step is – in the case of explicit schemes – adjusted internally for each spectral component. For other grid types, the user supplies a reference maximum propagation time step for the lowest model frequency $\Delta t_{p,r}$, assuming no currents, and no grid motion. For the frequency with counter m, the maximum time step $\Delta t_{p,m}$ is calculated within the model as

$$\Delta t_{p,m} = \frac{\dot{x}_{p,r}}{\dot{x}_{p,m}} \Delta t_{p,r} \; , \tag{3.4}$$

   where $\dot{x}_{p,r}$ is the maximum advection speed for the longest waves without currents or grid motion, and $\dot{x}_{p,m}$ is the actual maximum advection speed (including current) for frequency m. If the propagation time step is smaller than the global time step, the propagation effects are calculated with a number of successive smaller time steps. This generally implies that several partial time steps are used for the lowest frequency, but that the highest frequencies are propagated over the interval Δt_g with a single calculation. The latter results in a significantly more efficient model, particularly if higher-order accurate propagation schemes are used. Note that $\Delta t_{p,m}$ may be defined bigger than Δt_g, and that this has potential impact in model economy for cases with (strong) currents.
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. If strong refraction of long period swells occur, this may result in a notable undulation of mean wave parameters. This can be avoided by setting this time step to an even integer fraction of Δt_g.

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 3.6). This results in more accurate calculations for rapidly changing wind and wave conditions, and a more economical integration for slowly varying conditions. In order to limit the calculation time, a minimum time step is defined by the user.

The following sections deal with the separate steps in the fractional step method, and various subjects associated with this. The main issue are covered in Section 3.3, which addresses treatment of temporal variations of the water depth, Section 3.4 which addresses spatial propagation, Section 3.5 which addresses intra-spectral propagation, and Sections 3.6 and 3.7 which address the numerical integration of non-icea and ice source terms. The other sections deal with additional numerical approaches and techniques, covering the treatment of winds and currents (Section 3.9), including tides (Section 3.10), calculating space-time extremes (Section 3.11), treatment of ice (Section 3.8), spectral partitioning and the corresponding tracking of wave systems in space and time (Sections 3.12, 3.13), and nesting (Section 3.14).

3.3 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 Eq. (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 $0.05\Delta k$.

3.4 Spatial propagation

3.4.1 General concepts

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

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

where the propagated quantity $\mathcal{N}$ is defined as $\mathcal{N} \equiv N\, c_g^{-1} \cos\phi$. For the Cartesian grid, a similar equation is found for $\mathcal{N} \equiv N\, c_g^{-1}$. In this section

equations for the more complicated spherical grid are presented only. Conversion to a Cartesian grid is generally a simplification and is trivial.

Equation (3.5) in form is identical to the conventional deep-water propagation equation, but includes effects of both limited depths and currents. At the land-sea boundaries, wave action propagating toward 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 toward such points is absorbed, whereas action at the boundary points is used to estimate action fluxes for components traveling into the model.

The spatial grids can use two different coordinate systems, either a 'flat' Cartesian coordinate system typically used for small scale and idealized test applications, and a spherical (latitude-longitude) system used for most real-world applications. In model version 3.14, the coordinate system was selected at compile time with the `XYG` or `LLG` switches. In more recent model versions, the grid type is now a variable defined in ww3_grid and stored in the mod_def.ww3 file.

There is an option for spherical grids to have simple closure, to be periodic in the longitude direction, e.g. so that energy can propagate east from the maximum longitude in the grid to the minimum longitude in the grid. This closure is "simple" insofar as the index for latitude does not change across this "seam". A "not simple" type of closure is also permitted: this is associated with tripole grids. The tripole grid is a type of irregular grid and so this closure is discussed further in (3.4.3).

Up to model version 3.14, WAVEWATCH III considered only regular discrete grids, where the two main grid axes (x, y) are discretized using constant increments Δx and Δy. In model version 6.07 additional options have been included, including curvilinear grids and unstructured grids. In the following sections these grid approaches will be discussed, before additional propagation issues are addressed, covering the Garden Sprinkler Effect (3.4.6), continuously moving grids (3.4.8) unresolved islands (3.4.7), and rotated grids (3.4.9).

3.4.2 Traditional regular grids

Propagation schemes for traditional regular grids are selected at compile time using switches. Several schemes are available in WAVEWATCH III. These schemes are described in order of complexity below.

First-order scheme

Switch:	`PR1`
Origination:	WAVEWATCH III
Provided by:	H. L. Tolman

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

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

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

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

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

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

where Δt is the propagation time step, and $\Delta\phi$ and $\Delta\lambda$ are the latitude and longitude increments, respectively. Equations (3.6) through (3.8) with $\mathcal{N} = 0$

on land and applying Eq. (3.9) on sea points only automatically invokes the required boundary conditions.

Note that Eq. (3.9) represents a two-dimensional implementation of the scheme, for which the norm of the actual advection vectors needs to be used in Eq. (3.4). Note furthermore, that this implies a CFL criterion for the full equation, which is generally more stringent than that for a scheme where λ and ϕ propagation are treated separately as in the third order schemes discussed below. For a grid with equal increments in both directions, this results in a maximum time step that is a factor $1/\sqrt{2}$ smaller for the first order scheme than for the third order schemes.

Second-order scheme (UNO)

Switch:	`UNO`
Origination:	MetOffice
Provided by:	J.-G. Li

The upstream non-oscillatory 2nd order (UNO) advection scheme (Li, 2008) is an extension of the MINMOD scheme (Roe, 1986). In the UNO scheme, the interpolated wave action value at the mid-flux point for the cell face between cell i-1 and cell i is given by

$$N^*_{i-} = N_c + sign\,(N_d - N_c)\,\frac{(1-C)}{2}\min\left(|N_u - N_c|, |N_c - N_d|\right) \quad , \qquad (3.10)$$

where i- is the cell face index; $C = \left|\dot{\phi}_b\right| \Delta t/\Delta\phi$ is the absolute CFL number; and the subscripts u, c and d indicate the *upstream, central* and *downstream* cells, respectively, relative to the given i- cell face velocity $\dot{\phi}_b$. If $\dot{\phi}_b > 0$, $u = i$-2, c=i-1, d=i for the cell face between cell i-1 and cell i. If $\dot{\phi}_b \leq 0$ then u=i+1, c=i, d=i-1. Details of the UNO scheme are given in Li (2008) alongside standard numerical tests which demonstrate that the UNO scheme on Cartesian multiple-cell grids is non-oscillatory, conservative, shape-preserving, and faster than its classical counterpart as long as the CFL number is less than 1.0.

The flux and cell value update follow the same formulations as the first order upstream scheme, that is,

$$\mathcal{F}_{i-} = \phi_b \dot{N}^*_{i-}; \quad N_i^{n+1} = N_i^n + \frac{\Delta t}{\Delta\phi}\left(\mathcal{F}_{i-} - \mathcal{F}_{i+}\right) \quad , \qquad (3.11)$$

where $\mathcal{F}_{i+}$is the flux for the cell face between cell i and cell i+1. It can be estimated with a mid-flux value similar to (3.10) but with i replaced with i+1. An advective-conservative hybrid operator (Leonard et al., 1996) that reduces the time-splitting error is used to extend the UNO schemes to multi-dimensions.

Third-order scheme (UQ)

Switch:	`UQ`
Origination:	WAVEWATCH III
Provided by:	H. L. Tolman

The third-order accurate scheme available in WAVEWATCH III is the QUICKEST scheme (Leonard, 1979; Davis and More, 1982) combined with the ULTIMATE TVD (total variance diminishing) limiter (Leonard, 1991). This is the default propagation scheme for WAVEWATCH III. 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 (see Cahyono, 1994; Falconer and Cayhono, 1993; Tolman, 1995b). 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[7]

$$\mathcal{F}_{i,-} = \left[\dot{\phi}_b \, \mathcal{N}_b \right]^n_{j,l,m} \, , \tag{3.12}$$

$$\dot{\phi}_b = 0.5 \, \left(\dot{\phi}_{i-1} + \dot{\phi}_i \right) \, , \tag{3.13}$$

$$\mathcal{N}_b = \frac{1}{2} \left[\, (1+C)\mathcal{N}_{i-1} + (1-C)\mathcal{N}_i \, \right] - \left(\frac{1-C^2}{6} \right) \mathcal{CU} \, \Delta\phi^2, \tag{3.14}$$

$$\mathcal{CU} = \begin{cases} (\, \mathcal{N}_{i-2} - 2\mathcal{N}_{i-1} + \mathcal{N}_i \,) \, \Delta\phi^{-2} & \text{for} \quad \dot{\phi}_b \geq 0 \\ (\, \mathcal{N}_{i-1} - 2\mathcal{N}_i + \mathcal{N}_{i+1} \,) \, \Delta\phi^{-2} & \text{for} \quad \dot{\phi}_b < 0 \end{cases} \, , \tag{3.15}$$

$$C = \frac{\dot{\phi}_b \, \Delta t}{\Delta\phi} \, , \tag{3.16}$$

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

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

$$\begin{array}{lllll} \mathcal{N}_c = \mathcal{N}_{i-1}\,, & \mathcal{N}_u = \mathcal{N}_{i-2}\,, & \mathcal{N}_d = \mathcal{N}_i & \text{for} & \dot{\phi}_b \geq 0 \\ \mathcal{N}_c = \mathcal{N}_i\,, & \mathcal{N}_u = \mathcal{N}_{i+1}\,, & \mathcal{N}_d = \mathcal{N}_{i-1} & \text{for} & \dot{\phi}_b < 0 \end{array}\,. \tag{3.17}$$

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

$$\tilde{\mathcal{N}} = \frac{\mathcal{N} - \mathcal{N}_u}{\mathcal{N}_d - \mathcal{N}_u}\,. \tag{3.18}$$

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

$$\tilde{\mathcal{N}}_c \leq \tilde{\mathcal{N}}_b \leq 1, \quad \tilde{\mathcal{N}}_b \leq \tilde{\mathcal{N}}_c\, C^{-1}. \tag{3.19}$$

otherwise

$$\tilde{\mathcal{N}}_b = \tilde{\mathcal{N}}_c\,. \tag{3.20}$$

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 (3.14) are replaced by

$$\dot{\phi}_b = \dot{\phi}_s\,, \tag{3.21}$$

$$\mathcal{N}_b = \mathcal{N}_u\,, \tag{3.22}$$

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 (3.17) through (3.20).

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

$$\mathcal{N}^{n+1}_{i,j,l,m} = \mathcal{N}^{n}_{i,j,l,m} + \frac{\Delta t}{\Delta \phi}\left[\mathcal{F}_{i,-} - \mathcal{F}_{i,+}\right]\,. \tag{3.23}$$

The scheme for propagation in λ-space is simply obtained by rotating indices and increments in the above equations.[8]

Note that the ULTIMATE QUICKEST scheme is implemented as alternate one-dimensional schemes, for which the maxima of component advection speeds need to be used in Eq. (3.4). For consistency, the same time steps are always used for λ and ϕ propagation for a given component.

3.4.3 Curvilinear grids

Origination:	WAVEWATCH III(NRL Stennis)
Provided by:	W. E. Rogers, T. J. Campbell

As an extension to traditional "regular" grids, computations may be made on "irregular" grids within WAVEWATCH III . This makes it possible to run the model on alternate grid projections (e.g. Lambert conformal conic), rotated grids, or shoreline-following grids with higher resolution near shore, though the restrictions on time step from the conditionally stable schemes still apply. The same propagation schemes are utilized for irregular grids as for regular grids (Section 3.4.2).

The implementation is described in detail in Rogers and Campbell (2009), and summarized here: a Jacobian is used to convert the entire domain between the normal, curving space, and a straightened space. This conversion is performed only within the propagation routine, rather than integrating the entire model in straightened space. A simple, three step process is used every time the propagation subroutine is called (i.e. every time step and every spectral component): first, the dependent variable (wave action density) is converted to straightened space using a Jacobian; second, the wave action density is propagated via subroutine calls for each (of two) grid axes; third, the wave action density is converted back to normal, curved space. The actual flux computation is not significantly modified from its original, regular grid form. The same process occurs, regardless of grid type (regular or irregular); for regular grids, the Jacobian is unity.

Regarding the user interface: in ww3_grid.inp, a string is used to indicate

[8] The 'soft' boundary treatment as described on page 31 of Tolman (2002e) is no longer available, because it is incompatible with the advanced nesting techniques introduced in model version 3.14.

the grid type. In cases where this grid string is 'RECT', the model processes input for a regular grid. In case where this grid string is 'CURV' , the model processes input for an irregular grid. [Note that with WAVEWATCH III version 4.00, the coordinate system (i.e. degrees vs. meters) and the closure type (e.g. global/wrapping grid) are also specified in ww3_grid.inp ; the switches LLG and XYG are deprecated.]

With WAVEWATCH III version 5, capability is added to run on a special type of curvilinear grid, the "tripole grid" using the first-order propagation scheme. In the northern hemisphere, this grid type uses two poles instead of one, and both are over land to prevent singularities in grid spacing. This type of grid is sometimes used in ocean models, e.g. (Murray, 1996) and (Metzger et al., 2014). No special switch is required, and the grid is read in as any other irregular grid would be, but the user must specify a closure type (CSTRG) of TRPL in ww3_grid.inp. Specific details can be found in the documentation for ww3_grid.inp in Section 4.4.3. Propagation and gradient calculations are modified to deal with the new closure method. The TRPL closure type is compatible only with the first-order PR1 propagation scheme. An attractive feature of the tripole grid is that it allows the user to run a single grid which extends all the way to the North Pole. However, though the three poles are over land, there is still a convergence of meridians at the sea points nearest to them, meaning that the grid spacing in terms of real distances (which determines the maximum propagation time step) is still highly variable. More efficient grid spacing (meaning: with less variation of grid spacing in terms of real distances) can be achieved through the use of the multi-grid capability. Though this scheme addresses singularities in grid spacing at the pole, it does not address the singularity associated with definition of wave direction.

3.4.4 Triangular unstructured grids

Origination:	WWM-II
Provided by:	A. Roland, F. Ardhuin, M. Dutour-Sikirić, A. Abdolali

Triangle-based unstructured grids can be chosen in WAVEWATCH III by using numerical schemes based on contour residual distribution (CRD) (Ricchiuto et al., 2005) for the discretization of the space derivative. These

schemes have been adopted to the wave action balance equation in (Roland, 2009) and have been implemented in the Wind Wave Model-II (WWM-II) and in WW-III. With respect to the time integration methods, explicit and implicit schemes are available for unstructured grids. If the explicit solver is chosen the original fractional step method of WW3 is used, where the solution of the two dimensional hyperbolic part in geographical space is solved on unstructured grids based on the above mentioned methods. In this case the spectral part and the source terms are integrated in a similar way as in the structured version of WW3. The definition of the time steps is also done in the same way as for the structured part except that the number of sub-iterations for the unstructured part are estimated automatically based on the maximal CFL number in the domain. If implicit methods are chosen, a linear equation system is assembled based on the CRD-N schemes following (Roland, 2009). The spectral propagation part is solved with simple implicit 1^{st} order upwind schemes and the source terms are written in the matrix in the same way as in the dynamic scheme of WW3 using $\epsilon = 1$ in eq. 3.61 and assembling the equation system following simple Patankar rules. The full description of new implementations is discussed in Roland et al. (2019) It can be shown analytically and by numerical experiment that for CFL <1 the explicit and the implicit methods give similar results. This was also demonstrated in practical applications e.g. (Abdolali et al., 2018; Smith et al., 2018; Abdolali et al., 2019; Aboulali et al., 2019). However, for CFL >1 the transient solutions of the implicit schemes have a time step depedency since the source terms are linearized in time. The time step in the implicit schemes is equal the global timestep and the other time steps for spectral space or source terms have no effect. The fractional step methods has a time step dependency as well due to the splitting errors, which become especially predominant in shallow water (Roland, 2009). For both numerical methods proper convergence analysis with respect to the global time step and the spatial resolution need to be carried out unless the level of accuracy is satisfactory for the given applications. The numerical implementations have subsequently been evaluated in WWIII (e.g. Ardhuin et al., 2009b; Babanin et al., 2011; Bertin et al., 2014, 2015; Dodet et al., 2013; Ferrarin et al., 2008, 2013; Janekovic et al.; Kerr et al., 2013; Liau et al., 2011; Magne et al., 2010; Perrie et al., 2018, 2013b; Roland et al., 2006a,b, 2009, 2012; Roland and Ardhuin, 2014b; Sikirić et al., 2012, 2013, 2018; Zanke et al., 2006).

This option is activated by setting the grid string to '`UNST`' in ww3_grid.inp.

Four schemes have been implemented, and the choice of one or the other is done with the `UNST` namelist. These are the CRD-N-scheme (1st order), the CRD-PSI-scheme (better than 1st order, 2nd order on triangular structured grids), the CRD-FCT-scheme (2nd order space-time), and the implicit N-scheme. The default is the most efficient but diffusive explicit N-scheme. An implicit variant of the RD-Schemes using the method of lines and the N-Scheme for the space discretization was implemented in the SWAN model by Zijlema (2010) a similar discretization is chosen in the actual version of ECWAM see Roland (2012).

In this method the evolution of the spectrum at the nodes, where it is evaluated, is based on the redistribution over the nodes of the flux convergence into the median dual cells associated with the nodes (see Figure 3.1). For any spectral component, the advection equation, Eq. (3.5), is solved on the median dual cells: the incoming flux into a cell gives the rate of change of the wave action at the corresponding node. The various schemes implemented have different discretization for the estimation of this flux. The schemes have been presented in (see Roland, 2009, for a review).

We note that these advection schemes do not include corrections for the garden sprinkler effect (GSE). In the usual sense of GSE corrections it was found that the N-Scheme has sufficient numerical cross-diffusion to well compensate for this effect. However, higher order schemes such as CRD-PSI or CRD-FCT may create significant GSE effecting depending on the spatial resolution.

The parallelization of the unstructured grid schemes is done either using the Card Deck approach, as done for the structured grids, or using domain decomposition methods. The domain decompositions methods are based on ParMetis (Karypis, 2011), which is a C implementation of a parallel graph partitioning algorithm. ParMetis is being interfaced using PDLIB (Parallel Decomposition Library) in Fortran, which beside interfacing ParMetis provides the needed exchange routines for unstructured grids using the Fortran2002 standard. The exchange routines do all the needed communication for the decomposed grids between the various threads based on the decomposition provided by ParMetis. In this way, all the domain decomposition parts are incorporated in the main source with just one line of code. For the compilation of the PDLIB option in the switch files, which activates the domain decomposition parallelization, ParMetis needs to be installed and the code needs to be compiled with using MPI. PDLIB works with all usual MPI implementations, such as OpenMPI or MPICH2 and others. An schematic

view of implemented parallelization algorthims in WW3 (card deck and domain decomposition) are shown in figure 3.2. For real case applications, the convergence of these two methods is demonstrated in Aboulali et al. (2019) for Hurricane Ike, 2008.

In practice, the unstructured grid can be easily generated, using the PolyMesh software (developed by Aron Roland), from a shoreline polygons database and a list of arbitrary depth soundings. PolyMesh interfaces Triangle and adopts a-priori error estimates based on wave propagation characteristics and depth error based on the barycentric bathymetric error (e.g. Wessel and Smith, 1996). Other grid generation tools such as SMS and OceanMesh2D can also be used. There are no constrains with respect to the triangle shape.

The equivalent of the CFL condition for explicit finite difference schemes on regular grids is the ratio of the dual cell area divided by the product of the time step and the sum of the positive fluctuations into the dual cell. Because the spectral levels are imposed on the boundary for the positive fluxes, the boundary nodes are excluded from this CFL calculation and the incoming energy is set to zero, whereas the outgoing energy is fully absorbed. Only for the reflection part energy is allowed to penetrate from the boundary into the domain.

The boundary condition at the shoreline depends on the wave direction relative to the shoreline orientation. This particular treatment is enforced using the '`IOBPD`' array which is updated whenever the grid points status map '`MAPSTA`' changes. The grid geometry is also used to define local gradients of the water depth and currents. All other operations, such as interpolation of the forcing on the grid and interpolation from the grid onto output locations, is performed using linear interpolation in triangles.

All the triangle geometry operations assume a locally flat Earth. Depth and current gradients on the grid are estimated at the nodes by interpolating within on triangle using linear shape functions.

3.4.5 Spherical Multiple-Cell (SMC) grid

Switch:	`SMC`
Origination:	WAVEWATCH III(MetOffice)
Provided by:	J.-G. Li

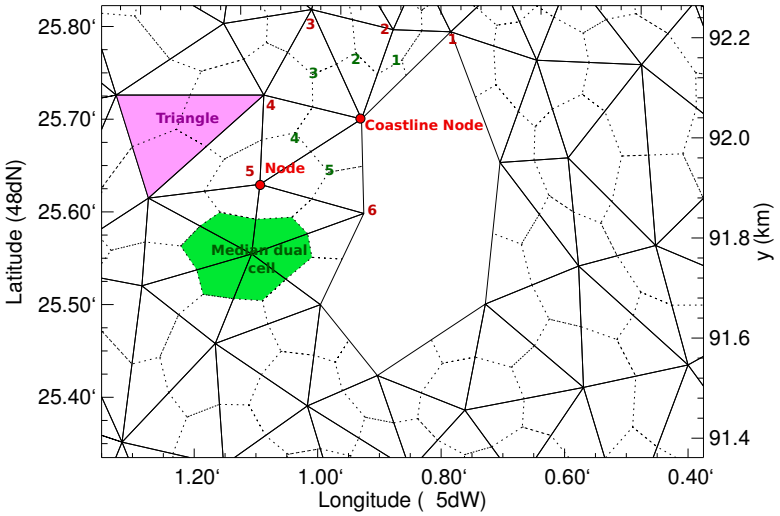

Figure 3.1: Example of a region of a triangle-based mesh, with in this case the small Island of Bannec, France. If the depth is greater than the minimum depth, the nodes of the shoreline are active. These are characterized by a larger number of neighbor nodes (6 in the example chosen) than neighbor triangles (5 in the same example).

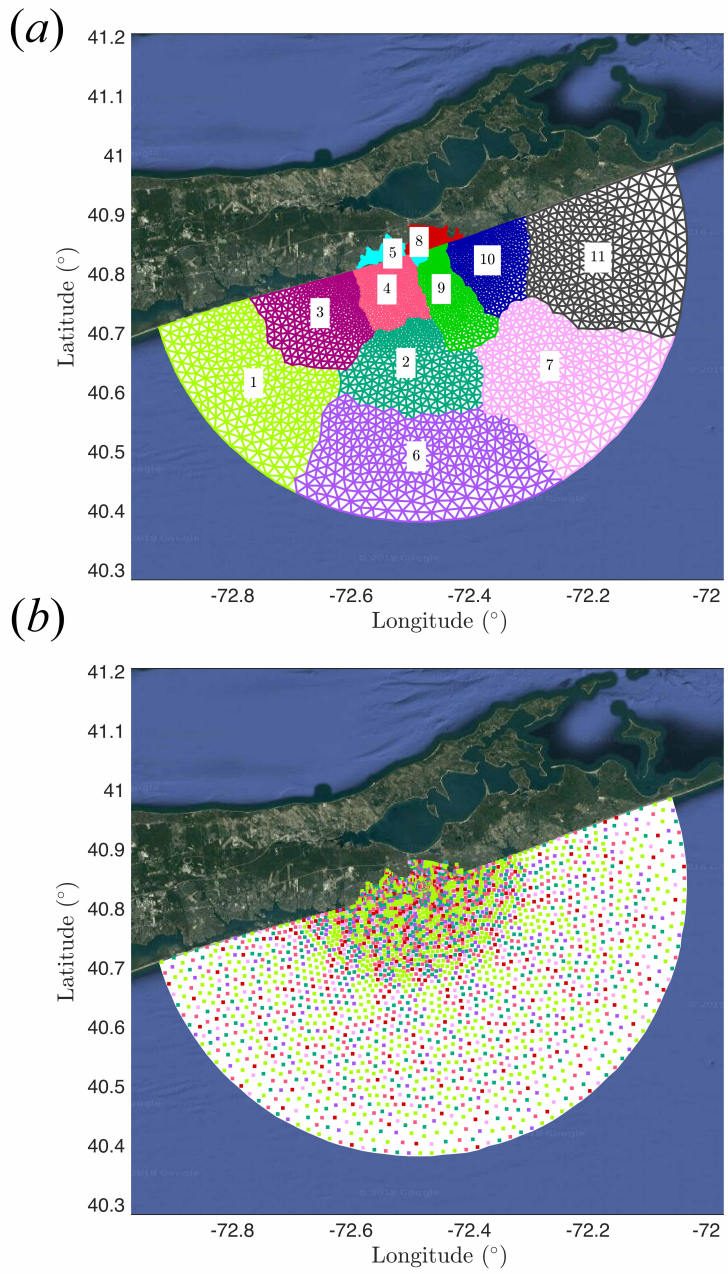

Figure 3.2: The schematic difference between Domain Decomposition (a) and Card Deck approach (b) on 11 computational cores represented by colors. The grid has ∼3k nodes and is build for Shinnecock Inlet, NY taken from an ADCIRC example problem.

The Spherical Multiple-Cell (SMC) grid[9] (Li, 2011) is an extension of the Cartesian multiple-cell grid (Li, 2003) onto the spherical coordinate system. It is an unstructured grid but retains the conventional lat-lon grid cells so that all propagation formulations on the spherical coordinates are still applicable and hence all the finite difference schemes. The SMC grid relaxes the CFL restriction at high latitudes in a similar fashion as the reduced grid (Rasch, 1994). Polar cells are introduced to remove the polar singularity of the differential transport equation by switching to an integral equation. The upstream non-oscillatory 2nd order (UNO) advection schemes (Li, 2008) is implemented on the SMC grid for both spatial and inter-spectral propagation. This 2nd order scheme can be replaced with a 3rd order scheme using the PSMC namelist logical variable `UNO3`. The UNO3 scheme is similar to the UQ scheme but replacing the flux limiters with the UNO 2nd order scheme. A simple rotation scheme is used for wave refraction-induced rotation and the great circle turning (Li, 2012). The refraction scheme is unconditionally stable for any time step but the maximum refraction induced rotation angle is limited by the maximum possible refraction angle towards the local gradient direction. Diffusion term similar to the Booij and Holthuijsen (1987) for alleviation of the garden sprinkler effect is used but the diffusion coefficient is simplified to a single homogeneous parameter (D_{nn} as in Eq. (3.32)). An additional 1-2-1 weighted averaging scheme is also available by the PSMC namelist logic variable `AVERG`. Reduction of computing time with this SMC grid is significant in comparison with the conventional grid, thanks to the relaxed time step restriction at high latitudes and removal of land points from the model. A remedy for the invalided scalar assumption at high latitude is provided to extend the global wave model into the entire Arctic Ocean (Li, 2016). This Arctic part can be activated by adding the ARC switch along side the SMC switch.

The SMC grid can be used for replacing the regular lat-lon grid so that the model domain can be extended to high latitudes or even the North Pole without reducing the time step. This application requires few changes to the regular grid model except for preparing a few extra input files, including the cell array and face array files. The cell array can be generated with the existing regular grid bathymetry by using the sea points only and merging cells in the longitudinal directions at a few latitude steps (Li, 2011).

[9] Presently this grid is activated by a compile switch and can only be used as a stand-alone grid. This will become a run time option in upcoming model versions.

Another important use of the SMC grid is for multi-resolution grids. The base level SMC grid cell can be refined into 4 quarterly cells by halving both the longitude and latitude grid lengths. Any cell on this refined level can be further divided into another 4 quarterly cells. This refinement can go on as required, resulting in multi-resolution grids in a few refined levels. For consistency, the single resolution SMC grid is considered to have only one level. The normal regular grid input files, such as the water depth, land-sea masks, and sub-grid obstruction, are no longer required, replaced with sea-point only cell and face arrays and a sub-grid obstruction file. The water depth is stored in the cell array in the last (5-th) column as an integer in meter. The masks will be defined inside ww3_grid with the sea-point cell array.

Wind and ocean current (if any) forcing can be applied using a regular grid at the base resolution as default input forcing for any SMC grid (single or multi-level). It will be interpolated on to the refined levels (if any) inside the model. One option of sea-point only wind and current input for SMC grid can be switched on by the `SEAWND` logical variable in the PSMC namelist. This sea-point only option not only removes the need to interpolate the wind (and current, if any) inside the model but also allows different resolution wind (and current) forcings to be mixed up and interpolated to multi-resolution cells directly, making it a truly multi-resolution model.

One important feature of the SMC grid is that it is an unstructured grid, that is, the cells are not required to be listed side by side as in their physical position. For the convenience of multi-resolution SMC grid, the cells are sorted by their sizes so that cells on one given level are grouped together in one sub-loop for a shared sub-time-step. The base level time step is halved as the grid length for the refined level sub-step. This effectively avoids the model to be slowed down by the refined cells due to their CFL restrictions. Neighboring cells information for propagation schemes are provided with cell face arrays, which are pre-calculated for the given cell array list. So there is no need to expand the sea point only SMC grid cells onto a full grid for propagation. Fig. 3.3 illustrates how SMC cell arrays are defined and Fig. 3.4 shows the Arctic region in a 6-12-25 km three level SMC grid. The golden and red circles mark the global and Arctic parts in the SMC6-25 grid. The Arctic part within the golden circle requires a fixed reference direction to define its wave directional bins. The global part (up to the golden circle) can be run independently without the Arctic part. The 4 rows from the red to the golden circles are duplicated in the Arctic part as boundary cells if the

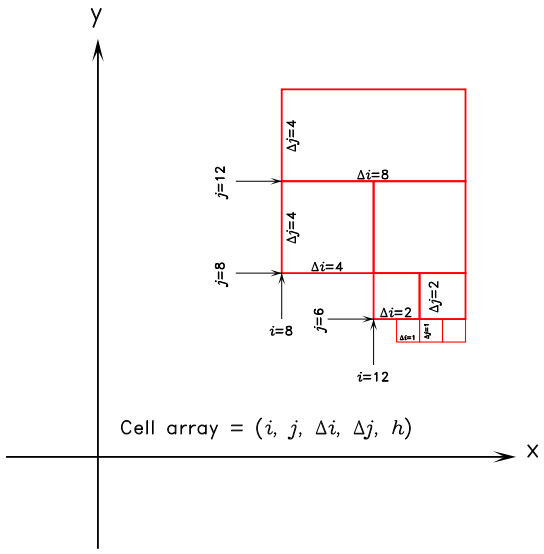

Figure 3.3: Illustration of cell arrays used in the SMC grid.

Arctic part is activated with the ARC option. Separate cell and face arrays are used for the Arctic part and they are merged into the global ones within the wave model for propagation.

Some IDL and F90 programs have been developed for generation of SMC grid cell and face arrays and visualization of the grid mesh and wave fields but they have not been formally included in the WW3 package yet. An IDL program (Glob50SMCels.pro) is provided in smc_docs/SMCG_TKs/ to generate a global 50km SMC grid using a 50km regular grid bathymetry ASCII input file (G50kmBathy.dat). Face array generation is done with two F90 programs, one for the global part (G50SGlSide.f90) and one for the Arctic part (G50SAcSide.f90). Due to the special treatment of the polar cell (Li, 2012), face arrays for the Arctic polar cell requires a different approach than other cells. The cell array file has to be sorted with a simple Linux script (countcells) before it is fed into the face array generation program. The face arrays also need to be sorted with a Linux script (countijsd) to determine the multi-level sub-loop counts. An independent spectral propagation test (G50SMCSRGD.f90) can be run to test the cell and face arrays and its output can be visualized with an IDL script, g50smstrspb.pro, which uses the saved projection files from the SMC grid visualization program, g50smcgrids.pro.

By modifying the projection parameters in g50smcgrids.pro, users can choose a projection view point (in lat-lon degree) and save the projection for model output visualization. The sub-grid obstruction file can be generated with the idl script Glob50SMCObstr.pro.

Compilation of the SMC grid option is similar to that for the regular lat-lon grid except for that the SMC switch is substituted for the PR2 UNO combination switches. Note that the SMC grid is built inside the regular lat-lon grid type so regular lat-lon grid parameters, such as NX, NY, SX, SY, X1, and Y1, are still required for SMC grid in ww3_grid.inp file at the base resolution level. The regular lat-lon grid water depth, land-sea masks, and sub-grid obstruction input files are no longer required and they are replaced with SMC grid sea point only files (depth is stored in the cell array and subgrid obstruction in G50GObstr.dat). The depth and land-sea mask input lines in ww3_grid.in are, however, kept for passing parameters, such as the minimum depth. Due to the merges at high latitudes and refined resolutions if any, regular grid mapping arrays are modified slightly for consistency with the SMC grid cells. Refer to the regression test *regtests/ww3_tp2.10* for an example of a 3-level SMC grid model for the Lake Erie.

Output post processing for SMC grid models has been developed for ww3_ounf and ww3_outf. For ww3_ounf the user can choose between retrieving multi-resolution information at native model grid sea-points, or generating regular grids at any of the resolution levels set in the model grid definition. The ww3_outf program processes SMC grid data as either the fully expanded regular lat-lon grid output at the base resolution level or as ASCII output at all SMC grid sea-points (type-4). For regular grid outputs, parameter values are determined using an area-weighted average of all SMC grid cells overlapping the chosen output cell boundaries. Sea-point data latitudes and longtitudes correspond to the native model grid cell centres. For ww3_ounf sea-point outputs, variables describing cell sizes are also provided in the netCDF file.

Visualization tools for sea-point outputs are not provided with this code release; however some examples are available from the Met Office on request. For netCDF sea-point outputs a python based GUI tool has been developed. Visualization of the all cell ASCII output can be done with the aid of the input cell array file because the output cell sequence is the same as the input cell array. The IDL script g50smcswhglb.pro is an example program to plot the global 50km SMC grid SWH output. It uses the projection files produced by g50smcgrids.pro. Users are encouraged to develop their own

grid-generating and post-processing programs in other languages.

It is recommended to read the smc_docs/SMC_Grid_Guide.pdf or the conference paper (Li and Saulter, 2017) at conference web page: http://www.waveworkshop.org/15thWaves/ for more information or to contact Jian-Guo.Li@metoffice.gov.uk for any help about the SMC grid.

3.4.6 The Garden Sprinkler Effect

The higher-order accurate propagation schemes are sufficiently free of numerical diffusion for the so-called 'Garden Sprinkler Effect' (GSE) to occur, i.e., a continuous swell field disintegrates into a set of discrete swell fields due to the discrete description of the spectrum (Booij and Holthuijsen, 1987, Fig. 3c). Several GSE alleviation methods are available in WAVEWATCH III, as described in the following sections.

No GSE alleviation

Switch:	`PR0 / PR1`
Origination:	WAVEWATCH III
Provided by:	H. L. Tolman

In case of no propagation (switch `PR0`) or for the first-order propagation scheme in a traditional or curvilinear grid no GSE alleviation is available or needed.

Booij and Holthuijsen 1987

Switch:	`PR2`
Origination:	WAVEWATCH III
Provided by:	H. L. Tolman

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

$$\frac{\partial \mathcal{N}}{\partial t} + \frac{\partial}{\partial x}\left[\dot{x}\mathcal{N} - D_{xx}\frac{\partial \mathcal{N}}{\partial x}\right] + \frac{\partial}{\partial y}\left[\dot{y}\mathcal{N} - D_{yy}\frac{\partial \mathcal{N}}{\partial y}\right] - 2D_{xy}\frac{\partial^2 \mathcal{N}}{\partial x \partial y} = 0\,, \quad (3.24)$$

$$D_{xx} = D_{ss}\,\cos^2\theta + D_{nn}\,\sin^2\theta\,, \quad (3.25)$$

$$D_{yy} = D_{ss}\,\sin^2\theta + D_{nn}\,\cos^2\theta\,, \quad (3.26)$$

$$D_{xy} = (D_{ss} - D_{nn})\,\cos\theta\,\sin\theta\,, \quad (3.27)$$

$$D_{ss} = (\Delta c_g)^2\,T_s/12\,, \quad (3.28)$$

$$D_{nn} = (c_g \Delta\theta)^2\,T_s/12\,, \quad (3.29)$$

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

$$\frac{\partial \mathcal{N}}{\partial t} = \frac{\partial}{\partial x}\left[D_{xx}\frac{\partial \mathcal{N}}{\partial x}\right] + \frac{\partial}{\partial y}\left[D_{yy}\frac{\partial \mathcal{N}}{\partial y}\right] + 2D_{xy}\frac{\partial^2 \mathcal{N}}{\partial x \partial y}\,. \quad (3.30)$$

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

$$D_{ss} = \left((X_\sigma - 1)\,\frac{\sigma_m}{2k_m}\right)^2 \frac{T_s}{12}\,, \quad (3.31)$$

$$D_{nn} = \left(\frac{\sigma_m}{2k_m} \Delta\theta \right)^2 \frac{T_s}{12} \,, \tag{3.32}$$

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 (3.30) 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. (3.30) (denoted as $\mathcal{D}_{i,-}$) is estimated as

$$D_{xx} \frac{\partial \mathcal{N}}{\partial x} \approx \mathcal{D}_{i,-} = D_{xx} \left(\frac{\mathcal{N}_i - \mathcal{N}_{i-1}}{\Delta x} \right) \Bigg|_{j,l,m} \,. \tag{3.33}$$

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 is located on land, Eq. (3.33) is set to zero. The mixed derivative at the right side of Eq. (3.30) (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 as

$$\mathcal{D}_{ij,--} = D_{xy} \left(\frac{-\mathcal{N}_{i,j} + \mathcal{N}_{i-1,j} + \mathcal{N}_{i,j-1} - \mathcal{N}_{i-1,j-1}}{0.5(\Delta x_j + \Delta x_{j-1})\, \Delta y} \right) \Bigg|_{l,m} \,. \tag{3.34}$$

Note that the increment Δx is a function of y due to the use of the spherical grid. This term is evaluated only if all four grid points considered are sea points, otherwise it is set to zero. Using a forward in time discretization of the first term in Eq. (3.30), and central in space discretizations for the remainder of the first and second term on the right side, the final algorithm becomes

$$\begin{aligned} \mathcal{N}^{n+1}_{i,j,l,m} = \mathcal{N}^{n}_{i,j,l,m} &+ \frac{\Delta t}{\Delta x} \left(\mathcal{D}_{i,+} - \mathcal{D}_{i,-} \right) + \frac{\Delta t}{\Delta y} \left(\mathcal{D}_{j,+} - \mathcal{D}_{j,-} \right) \\ &+ \frac{\Delta t}{4} \left(\mathcal{D}_{ij,--} + \mathcal{D}_{ij,-+} + \mathcal{D}_{ij,+-} + \mathcal{D}_{ij,++} \right) \,. \end{aligned} \tag{3.35}$$

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

$$\frac{D_{\max}\, \Delta t}{\min(\Delta x, \Delta y)^2} \leq 0.5 \,, \tag{3.36}$$

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 this putting undue constraints on the time step of a model, a corrected swell age $T_{s,c}$ is used

$$T_{s,c} = T_s \min \left\{ 1 \, , \, \left(\frac{\cos(\phi)}{\cos(\phi_c)} \right)^2 \right\} \, , \tag{3.37}$$

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

The above diffusion is needed for swell propagation, but is not realistic for growing wind seas. For wind seas, the ULTIMATE QUICKEST scheme without the dispersion correction is sufficiently smooth to render stable fetch-limited growth curves (Tolman, 1995b). 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

$$D_g = \frac{c_g \, \min(\Delta x, \Delta y)}{\mathcal{R}} \, . \tag{3.38}$$

The diffusion 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

$$X_g = \min \left\{ 1 \, , \, \max \left[0 \, , \, 3.3 \left(\frac{k \, u_{10}}{\sigma} \right) - 2.3 \right] \right\} \, , \tag{3.39}$$

$$D_{ss} = X_g D_g + (1 - X_g) D_{ss,p} \, , \tag{3.40}$$

$$D_{nn} = X_g D_g + (1 - X_g) D_{nn,p} \, , \tag{3.41}$$

where the suffix p denotes propagation diffusion as defined in Eqs. (3.31) and (3.32). The constants in Eqs (3.38) and (3.39) are preset in the model.

Spatial averaging

Switch:	PR3
Origination:	WAVEWATCH III
Provided by:	H. L. Tolman

The major drawback of the above GSE alleviation method is its potential impact on model economy as discussed in relation to Eq. (3.36) and in Tolman (2001, 2002a). For this reason, an alternative additional GSE alleviation method has been developed for WAVEWATCH III.

This method which represents the default for WAVEWATCH III, replaces the additional diffusion step (3.30) with a separate fractional step in which direct averaging of the field of energy densities for a given spectral component is considered. The area around each grid point over which the averaging is performed extends in the propagation (**s**) and normal (**n**) directions as

$$\pm\, \gamma_{a,s}\, \Delta c_g\, \Delta t\, \mathbf{s} \quad , \pm \gamma_{a,n}\, c_g \Delta\theta\, \Delta t\, \mathbf{n} \quad , \tag{3.42}$$

where $\gamma_{a,s}$ and $\gamma_{a,n}$ are tunable constants, the default value of which is set to 1.5. This averaging is illustrated in Fig. 3.5. Note that these values may require some retuning for practical applications, as discussed in Tolman (2002a). Appendix A of the latter paper presents details of the averaging scheme, including conservation considerations. Consistency with the Booij and Holthuijsen (1987) approach furthermore implies that $\gamma_{a,s}$ and $\gamma_{a,n}$ should vary with the spatial grid resolution (see Chawla and Tolman, 2008, Appendix).

Note that this kind of averaging with dominant directions **s** and **n** is similar to the Booij and Holthuijsen (1987) diffusion method, that uses the same main directions. The averaging method, however, never influences the time step, because it is completely separated from the actual propagation. Moreover, if explicit schemes are used with typically $c_g \Delta t / \Delta x < 1$, it is obvious that the averaging over the area as defined in (3.42) will generally require information at directly neighboring spatial grid points only, as in Fig. 3.5. Furthermore, this method does not require high-latitude filtering.

As is illustrated in Tolman (2002a,d), this method gives virtually identical results as the previous method, but does so at slightly lower costs. For high-resolution applications, the averaging method may become dramatically more economical.

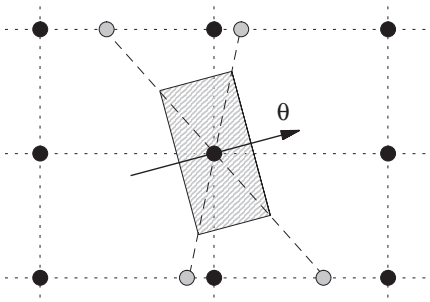

Figure 3.5: Schematic of spatial averaging GSE alleviation technique. Solid circles and dotted lines represent the spatial grid. Hatched area represent averaging area to be considered. Corner point values are obtained from the central grid point and the gray points. The latter values are obtained by interpolation from adjacent grid points (from Tolman, 2002a).

Finally, the GSE can be alleviated somewhat by assuring that the discrete spectral directions do not coincide with spatial grid lines. This can be achieved by defining the first discrete direction θ_1 as

$$\theta_1 = \alpha_\theta \, \Delta\theta \quad , \tag{3.43}$$

where $-0.5 \leq \alpha_\theta \leq 0.5$ can be defined by the user. Note that setting $\alpha \neq 0$ is beneficial to the first-order scheme, but has negligible impact on the third-order scheme.

3.4.7 Unresolved obstacles

Origination:	WAVEWATCH III
Provided by:	H. L. Tolman, L. Mentaschi

Even at the time of the original tuning of WAVEWATCH III version 1.15 (Tolman, 2002f), it was clear that unresolved islands groups are a major source of local wave model errors. This was illustrated in some more detail in Tolman (2001, Fig. 3), and Tolman et al. (2002, Fig. 8). In WAVEWATCH III, two approaches are available for the parameterization of unresolved obstacles. The first, originally from SWAN (Booij et al., 1999; Holthuijsen et al., 2001), applies the effects of unresolved obstacles at the cell boundaries of the spatial grid within the numerical scheme, and is established for regular grids. The second approach, proposed by (Mentaschi et al., 2015b), is based on source terms, and can be applied to different types of mesh. In this section the first approach is presented, while for a description of the second approach the reader is referred to section 2.3.20.

In the propagation-based approach, the numerical fluxes between cells through their common boundary are suppressed according to the degree of obstruction provided by the unresolved obstacle. In this approach, the numerical propagation scheme of the ULTIMATE QUICKEST scheme of Eq. (3.23) is modified as

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

where $\alpha_{i,-}$ and $\alpha_{i,+}$ are 'transmissions' of the corresponding cell boundaries, ranging from 0 (closed boundary) to 1 (no obstructions). For outflow boundaries, transparencies by definition are 1, otherwise energy will artificially accumulate in cells. For inflow boundaries, transparencies less than 1 result in elimination of obstructed energy at the cell boundary. This approach is illustrated in Fig. 3.6. Note that a similar approach is easily adopted in the first- and second-order schemes. Note, furthermore, that an alternate obstruction approach with obstructions as a function of the spectral direction θ has been used by Hardy and Young (1996) and Hardy et al. (2000).

Two methods for defining the obstructions are available in the model. The first defines the obstructions directly at the grid boundary. This requires the generation of staggered depth-transparency grids. The second allows the user to define depths and transparencies at the same grid. In this case, the

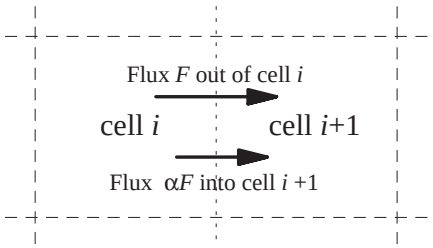

Figure 3.6: Treatment of unresolved obstacles. Common cell boundary (dotted line) has transparency α. Dashed lines represent other cell boundaries. Numerical flux from left to right.

---

transparency at the inflow boundary becomes $0.5(1 + \alpha_i)$, and the outflow transparency by definition is 1. To complete the total transparency α_i, the next cell in the flow direction will have an inflow transparency $2\alpha_i/(1 + \alpha_i)$. If consecutive cells are partially obstructed, the product of individual transparencies is applied.

This approach can also be used to continuously model the effects of ice coverage on wave propagation. This is discussed in Section 3.8. Details of the sub-grid treatment of islands and ice can be found in Tolman (2003b). A study of impacts of this approach in large-scale wave models is presented in Tolman (2002d, 2003b).

The default setting of WAVEWATCH III is to not include sub-grid modeling of obstacles. Generating obstruction grids can be labor intensive. For this reason, an automated approach for generating bottom and obstruction grids was developed by Chawla and Tolman (2007, 2008). Note that this option does not involve compile-level choices, but is entirely controlled from the grid preprocessor (see Chapter 4).

3.4.8 Continuously moving grids

Switch:	`MGx`
Origination:	WAVEWATCH III
Provided by:	H. L. Tolman

In order to address wave growth issues in rapidly changing, small scale

conditions such as hurricanes, an option to add a given continuous advection speed to the grid has been added to the model in model version 3.02. This model version is described in detail in Tolman and Alves (2005). Here, only a cursory description is given.

WARNING

The continuously moving grid version of WAVEWATCH III is only intended for testing wave model properties in highly-idealized conditions. This model version should only be used for deep water without mean currents and land masses. Furthermore, to avoid complications with great circle propagation, only a Cartesian grid should be used. The option is furthermore implemented only for propagation options PR1 and PR3. Note that this is not checked in the scripts or programs at either the compile or run time level. This option is not considered to be for general application.

WARNING

For the above described application Eq. (2.8) can be written as

$$\frac{\partial N}{\partial t} + (\dot{\mathbf{x}} - \mathbf{v}_g) \cdot \nabla_x N = \frac{S}{\sigma} \,, \tag{3.45}$$

where $\mathbf{v}_g$ represents the advection velocity of the grid. This option is selected when compiling the model. A second compile level option allows for adding the grid advection velocity $\mathbf{v}_g$ to the wind field. This allows for a simple method to assure mass conservation of a wind field independent of the actual and instantaneous grid advection velocity. The advection velocity $\mathbf{v}_g$ can vary in time and is provided by the user at the run time of the model (see below).

For the simplified conditions for which Eq. (3.45) is valid, the implementation of the moving grids is trivial if it is considered that this equation is equivalent to

$$\frac{\partial N}{\partial t} + \nabla_x \cdot (\dot{\mathbf{x}} - \mathbf{v}_g)\, N = \frac{S}{\sigma} \,, \tag{3.46}$$

which in turn implies that the advection velocity $\mathbf{v}_g$ can be added directly to $\dot{\mathbf{x}}$ for arbitrary numerical schemes solving Eq. (2.8). Because this influences the net advection velocity, it also influences stability characteristics. This impact has been accounted for automatically by including the moving grid

velocity in the calculation of the actual propagation time step in Eq (3.4). Hence, the user need to provide a proper maximum propagation time step representative for $\mathbf{v_g} = \mathbf{0}$ only.

The motion of the grid has an apparent influence on the Garden Sprinkler Effect (GSE), due to the different retention time in the grid of spectral components with identical frequency but different propagation direction. Current GSE alleviation methods tend to be more efficient for younger swells than for older swells. Hence, swells with longer retention time in the moving grid tend to show a more pronounced GSE (see Tolman and Alves, 2005). To mitigate this apparent imbalance in GSE alleviation, Eq. (3.42) is replaced with

$$\pm\, \gamma_a \gamma_{a,s}\, \Delta c_g\, \Delta t\, \mathbf{s} \quad , \pm \gamma_a \gamma_{a,n}\, c_g \Delta\theta\, \Delta t\, \mathbf{n} \quad , \tag{3.47}$$

$$\gamma_a = \left(\frac{|\dot{\mathbf{x}}|}{|\dot{\mathbf{x}} - \mathbf{v_g}|} \right)^p \tag{3.48}$$

where γ_a is a correction factor accounting for the grid movement, and where the power p is a parameter allows for some tuning. With this modification, the effects of the GSE can be distributed more evenly over the grid by rescaling the amount of smoothing applied with the expected residence time of corresponding spectral component in the moving grid (see Tolman and Alves, 2005).

To switch on the moving grid option or the corrections of the wind field or GSE, three optional switches are added to the WAVEWATCH III source code (also see, Section 5.9:

MGP	Apply advection correction for continuous moving grid.
MGW	Apply wind correction for continuous moving grid.
MGG	Apply GSE alleviation for continuous moving grid.

The advection velocity and direction is input to the shell similar to the input of homogeneous currents (see bottom of file ww3_shel.inp in Section 4.4.10), exchanging the keyword '`CUR`' with '`MOV`'. The advection velocity can be changed in time like all homogeneous input fields. An example of running with a moving grid model is given in test case ww3_ts3. A similar capability exist in ww3_multi.inp in Section 4.4.12, and is tested in test case mww3_test_05.

3.4.9 Rotated grids

Switch:	`RTD`
Origination:	WAVEWATCH III (MetOffice)
Provided by:	J.-G. Li

The rotated grid is a latitude-longitude (lat-lon) grid and is obtained by rotating the North Pole to a new position at latitude ϕ_p and longitude λ_p in the standard latitude-longitude system. The new pole position is chosen so that the model domain of interest may be placed around the rotated equatorial area for a evenly-spaced lat-lon mesh. For this reason the rotated grid is also known as *Equatorial grid*. For instance, the North Atlantic and European wave (NAEW) model used in the UK Met Office uses a rotated pole at 37.5N, 177.5E so that London, UK (˜51.5N 0.0E) is almost on the rotated equator. This rotated grid allows a much more evenly spaced lat-lon mesh in the NAEW domain than the standard lat-lon grid in the same area.

In WAVEWATCH III the rotated grid is implemented with minimum changes to the original lat-lon grid. In fact, the rotated grid is treated just like the standard lat-lon grid inside the model. To set up and run a rotated grid model configuration, users should choose the regular lat-lon grid along with the `RTD` switch. The rotated pole position is set using the `PLAT` and `PLON` variables in the **ww3_grid.inp** namelist `ROTD`. Model input files, like wind, current and ice files should be mapped on to the rotated grid. For convenience of nesting in standard lat-lon grid frameworks, boundary conditions provided to and output from the rotated grid use spectra referenced to a standard grid north and standard lat-lon grid points values, which are converted into rotated grid lat-lon inside WAVEWATCH IIIṪhe list of 2D spectral output locations in **ww3_shel.inp** are also specified in standard lat-lon.

Model directional and x-y vector outputs can be converted to a standard grid north reference by setting the UNROT variable in the **ww3_grid.inp** namelist ROTD to True. With this set, for point outputs lat-lon locations all directional values such as wind direction, current direction and 2D spectra are converted into standard lat-lon orientation. Functions to de-rotate gridded fields are applied in **ww3_ounf**, **ww3_outf** and **ww3_grib**. When running **ww3_ounf** and **ww3_ounp**, the resulting netCDF files will include a variable attribute direction_reference, which describes whether a standard (True North) or rotated grid directional reference frame has been used. Gridded netCDF files generated by **ww3_ounf** also include *standard_latitude* and

standard_longitude two-dimensional arrays that describe location of the rotated model cell centres in the standard lat-lon reference frame.

Six subroutines are provided in module **w3servmd.ftn** for rotated grid conversion:

W3SPECTN	Turns wave spectrum anti-clockwise by AnglD
W3ACTURN	Turns wave action(k,nth) anti-clockwise by AnglD
W3THRTN	Turns direction parameters anti-clockwise by AnglD
W3XYRTN	Turns x-y vector parameters anti-clockwise by AnglD
W3LLTOEQ	Convert standard into rotated lat/lon plus AnglD
W3EQTOLL	Reverse of w3lltoeq, but AnglD unchanged

These subroutines are self-contained and can be extracted outside the model for pre- or post-processing of rotated grid files. Some conversion tools have been developed based on these subroutines but have not been included in WAVEWATCH III yet. Refer to the regression test *regtests/ww3_tp2.11* for an example of a rotated grid model (NAEW). Users may find more information in *smc_docs/Rotated_Grid.pdf* or contact Jian-Guo Li for help (Jian-Guo.Li@metoffice.gov.uk).

3.5 Intra-spectral propagation

3.5.1 General concepts

The third step of the numerical fractional step algorithm considers refraction and residual (current-induced) wavenumber shifts. Irrespective of the spatial grid discretization and coordinate system, the equation to be solved in this step becomes

$$\frac{\partial N}{\partial t} + \frac{\partial}{\partial k}\dot{k}_g N + \frac{\partial}{\partial \theta}\dot{\theta}_g N = 0\,, \tag{3.49}$$

$$\dot{k}_g = \frac{\partial \sigma}{\partial d}\frac{\mathbf{U}\cdot\nabla_x d}{c_g} - \mathbf{k}\cdot\frac{\partial \mathbf{U}}{\partial s}\,, \tag{3.50}$$

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 or due to strong current shears. Similarly, the propagation in k-space suffers from similar problems in very shallow water. To avoid the need of extremely small time steps due to refraction, the propagation velocities in θ-space and k-space (2.11) are filtered,

$$\dot{\theta} = X_{rd}(\lambda, \phi, k) \left(\dot{\theta}_d + \dot{\theta}_c + \dot{\theta}_g \right) \ , \tag{3.51}$$

where the indices d, c and g refer to the depth, current and great-circle related fraction of the refraction velocity in (2.11). The filter factor X_{rd} is calculated for every wavenumber and location separately, and is determined so that the CFL number for propagation in θ-space due to the *depth* refraction term cannot exceed a pre-set (user defined) value (default 0.7). This corresponds to a reduction of the bottom slope for some low frequency wave components. For mid-latitudes, the affected components are expected to carry little energy because they are in extremely shallow water. Long wave components carrying significant energy are usually traveling toward the coast, where their energy is dissipated anyway. This filtering is also important for short waves, and close to the pole. The effect of this filter can be tested by reducing the time steps for intraspectral refraction and by looking at the maximum CFL numbers in the output of the model. These are computed just before the filter is applied.

The spectral space is always discretized with constants directional increments and a logarithmic frequency grid (3.1) to accommodate computations of the nonlinear interaction S_{nl}. First, second and third orders schemes are available, and are presented in the following sections.

3.5.2 First-order scheme

Switch:	`PR1`
Origination:	WAVEWATCH III
Provided by:	H. L. Tolman

In the first order scheme the fluxes in θ- and k-space are calculated using Eqs. (3.6) through (3.8) (replacing $\mathcal{N}$ with N and rotating the appropriate counters). The complete first order scheme becomes

$$N^{n+1}_{i,j,l,m} = N^{n}_{i,j,l,m} + \frac{\Delta t}{\Delta\theta}\left[\mathcal{F}_{l,-} - \mathcal{F}_{l,+}\right] + \frac{\Delta t}{\Delta k_m}\left[\mathcal{F}_{m,-} - \mathcal{F}_{m,+}\right] \ , \tag{3.52}$$

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.

3.5.3 Second-order scheme (UNO)

Switch:	`UNO`
Origination:	Met Office
Provided by:	J.-G. Li

The UNO scheme for the directional θ-space is identical to the regular grid one assuming that the directional bins are regularly spaced. For the k-space, however, the UNO scheme uses its irregular version, which uses local gradients instead of differences to estimate wave action value at the mid-flux point for the cell face between spectral bin i-1 and i, that is:

$$N^{*}_{i-} = N_c + sign\left(N_d - N_c\right)\frac{\left(\Delta k_c - |\dot{k}_{i-}|\Delta t\right)}{2}\min\left(|\frac{N_u - N_c}{k_u - k_c}|, |\frac{N_c - N_d}{k_c - k_d}|\right) \ , \tag{3.53}$$

where i- is the wave number k bin index; the subscripts u, c and d indicate the *upstream, central* and *downstream* cells, respectively, relative to the given

i- face velocity $\dot{k}_{i-}$; k_c is the central bin wave number and Δk_c is the central bin widith. Details of the irregular grid UNO scheme are given in Li (2008).

Boundary conditions for the θ-space is the natural periodic condition. For the k-space, two more zero spectral bins are added to each end of the wave spectral domain as the UNO scheme is 2nd order in accuracy.

3.5.4 Third-order scheme (UQ)

Switch:	`UQ`
Origination:	WAVEWATCH III
Provided by:	H. L. Tolman

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 (3.12) through (3.16) then become

$$\mathcal{F}_{m,-} = \left[\dot{k}_{g,b}\, N_b \right]_{i,j,l}^{n} \,, \tag{3.54}$$

$$\dot{k}_{g,b} = 0.5 \left(\dot{k}_{g,m-1} + \dot{k}_{g,m} \right) \,, \tag{3.55}$$

$$N_b = \frac{1}{2} \left[(1+C)N_{i-1} + (1-C)N_i \right] - \frac{1-C^2}{6}\, \mathcal{CU}\, \Delta k_{m-1/2}^2, \tag{3.56}$$

$$\mathcal{CU} = \begin{cases} \frac{1}{\Delta k_{m-1}} \left[\frac{N_m - N_{m-1}}{\Delta k_{m-1/2}} - \frac{N_{m-1} - N_{m-2}}{\Delta k_{m,-3/2}} \right] & \text{for} \quad \dot{k}_b \geq 0 \\ \frac{1}{\Delta k_m} \left[\frac{N_{m+1} - N_m}{\Delta k_{m+1/2}} - \frac{N_m - N_{m-1}}{\Delta k_{m-1/2}} \right] & \text{for} \quad \dot{k}_b < 0 \end{cases} \,, \tag{3.57}$$

$$C = \frac{\dot{k}_{g,b}\, \Delta t}{\Delta k_{m-1/2}} \,, \tag{3.58}$$

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. (3.17) through (3.20), if the CFL number of Eq. (3.58) 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

$$N^{n+1}_{i,j,l,m} = N^{n}_{i,j,l,m} + \frac{\Delta t}{\Delta k_m} \left[\mathcal{F}_{m,-} - \mathcal{F}_{m,+} \right] , \tag{3.59}$$

3.6 Non-ice source term integration

The source terms not involving ice are accounted for by solving

$$\frac{\partial N}{\partial t} = \mathcal{S}_{no\ ice} . \tag{3.60}$$

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

$$\Delta N(k,\theta) = \frac{\mathcal{S}(k,\theta)}{1 - \epsilon D(k,\theta)\Delta t} , \tag{3.61}$$

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

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

$$\Delta t^n_d = \min_{f<f_{hf}} \left[\frac{\Delta N_m}{|\mathcal{S}|} \left(1 + \epsilon D \frac{\Delta N_m}{|\mathcal{S}|} \right)^{-1} \right] , \tag{3.62}$$

$$\Delta t^n_d = \max \left[\Delta t^n_d , \Delta t_{d,\min} \right] , \tag{3.63}$$

$$\Delta t_d^n = \min \left[\Delta t_d^n \, , \, \Delta t_g - \sum_{i=1}^{n-1} \Delta t_d^i \right] \, , \tag{3.64}$$

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

$$N^n = \max \left[0 \, , \, N^{n-1} + \left(\frac{\mathcal{S} \Delta t_d}{1 - \epsilon D \Delta t_d} \right) \right] \, . \tag{3.65}$$

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

$$\Delta N_m(k,\theta) = \min \left[\, \Delta N_p(k,\theta) \, , \, \Delta N_r(k,\theta) \, \right] \, , \tag{3.66}$$

$$\Delta N_p(k,\theta) = X_p \, \frac{\alpha}{\pi} \, \frac{(2\pi)^4}{g^2} \, \frac{1}{\sigma k^3} \, , \tag{3.67}$$

$$\Delta N_r(k,\theta) = X_r \, \max \left[\, N(k,\theta) \, , \, N_f \, \right] \, , \tag{3.68}$$

$$N_f = \max \left[\Delta N_p(k_{\max},\theta) \, , \, X_f \, \max_{\forall k,\theta} \{ N(k,\theta) \} \right] \, , \tag{3.69}$$

where X_p, X_r and X_f are user-defined constants (see Table 3.1), α is a PM spectrum energy level (set to $\alpha = 0.62 \times 10^{-4}$) and $k_{\max}$ is the maximum discrete wavenumber. The parametric spectral shape in (3.67) 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 (3.69) 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 (3.67) in conditions where Eq. (3.63) dictates Δt_d^n. In this case Eq. (3.63) becomes a limiter as in the WAM model. Impacts of limiters are discussed in detail in for instance Hersbach and Janssen (1999, 2001), Hargreaves and Annan (2001) and Tolman (2002c).

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

It is possible to compile WAVEWATCH III without using a linear growth term. In such a case, waves can only grow if some energy is present in

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

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

the 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 WAVEWATCH III (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

$$N_{\min}(k_{seed},\theta) = 6.25\times10^{-4}\frac{1}{k_{\rm seed}^3\,\sigma_{\rm seed}}\max\left[\,0.\,,\,\cos^2(\theta-\theta_w)\right] \min\left[\,1\,,\,\max\left(\,0\,,\,\frac{|u_{10}|}{X_{\rm seed}g\sigma_{\rm seed}^{-1}}-1\,\right)\right], \quad (3.70)$$

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 (3.67) for large wind speeds. $X_{\rm seed} \geq 1$ is a user-defined parameter to shift seeding to higher frequencies. Seeding starts if the wind speed reaches $X_{\rm seed}$ times the equilibrium wind speed for the highest discrete frequency, and reaches its full strength for twice as high wind speeds. The default model settings include the seeding algorithm, with $X_{\rm seed} = 1$.

In model version 3.11, surf-zone physics parameterizations have been introduced. Such physics, particularly depth-induced breaking, operate on much smaller time scales than deep water and limited-depth physics outside the surf zone. To assure reasonable behavior for larger time steps, an additional optional limiter has been adopted from the SWAN model, which can be used instead of modeling surf-breaking explicitly. This limiter is similar to the Miche style maximum wave height in the depth-limited wave breaking source term of Eq. (2.166). In this limiter, the maximum wave energy E_m is computed as

$$E_m = \frac{1}{16}[\gamma_{lim} \tanh(\bar{k}d)/\bar{k}]^2 \quad , \tag{3.71}$$

where γ_{lim} is a factor comparable to γ_M in Eq. (2.166), with the caveat that γ_M is representative for an individual wave, whereas γ_{lim} is representative for the significant wave height. For monochromatic waves, the original expression by Miche (1944) would correspond to $\gamma_{lim} = 0.94$ and replacing H_s by the height H of the waves. Here this idea is applied to random waves. In shallow water, this limits H_s to be less than $\gamma_{lim}d$. If the total spectral energy E is larger than the maximum energy E_m, the limiter is applied by simply rescaling the spectrum by the factor E/E_m, loosely following the argumentation from Eldeberky and Battjes (1996) and used in Section 2.3.17.

This limiter can be switched on or off in the compilation of the model, and γ_{lim} can be adjusted by the user. The default is set to $\gamma_{lim} = 1.6$ because H_{rms} values close to d have indeed been recorded and thus taking a ratio H_s/H_{rms} of 1.4, using 1.6 allows this large steepness to be exceeded by some margin. Note that this limiter should be used as a 'safety valve' only, and hence that it should be less strict than the breaking criterion in the surf-breaking or whitecapping source terms, if these source terms are modeled explicitly.

Also, this limiter does not guarantee that all parts of the spectrum are realistic. Indeed, the use of a mean wavenumber, as in the Komen et al. dissipation, makes it possible to have unrealistically steep short waves in the presence of swell. A future extension of this limiter could be to limit the steepness with a partial spectral integration in frequencies, to make sure that waves of all scales are indeed not too steep.

3.7 Ice source terms integration

Because the attenuation and scattering in the ice can be very strong (although they are linear), it is convenient to perform a separate integration of the ice terms $S_{\text{ice}} = S_{\text{id}} + S_{\text{is}}$. This combines a dissipation term

$$S_{\text{id}}/\sigma = \beta_{\text{id}} N, \tag{3.72}$$

and a scattering term which is of the form

$$\frac{S_{\rm is}(k,\theta)}{\sigma} = \int_0^{2\pi} \beta_{\rm is}[N(k,\theta') - N(k,\theta)]d\theta', \tag{3.73}$$

in which the scattering coefficient $\beta_{\rm is}$ is a priori a function of the difference in direction between incident θ' and scattered θ directions, as well as the shape of ice floes. In general the directional spectrum $N(k,.)$ is a vector with NTH (number of directions) components, and the source term is a vector of the same size given by the matrix product $S/\sigma = MN(k,.)$ where M is a positive symmetric square NTH by NTH matrix with components given from the $\beta_{\rm id}$ values. The matrix M is easily diagonalized as

$$M = VDV^T, \tag{3.74}$$

where D is a diagonal matrix containing all eigenvalues and V is the array of eigenvectors, and V^T is its transpose. As a result the split wave action equation for ice source terms

$$\frac{\partial}{\partial t}\frac{N}{c_g} = \frac{S_{\rm id}}{\sigma c_g}, \tag{3.75}$$

can be rewritten for the action N_i of each eigenvector V_i with eigenvalue λ_i as

$$\frac{\partial}{\partial t}\frac{N_i}{c_g} = \frac{\beta_{\rm id}+\lambda_i}{\sigma c_g}N_i, \tag{3.76}$$

which has the following exact solution

$$N_i(t+\Delta t_g) = N_i(t)\exp\left[(\beta_{\rm id}+\lambda_i)+\Delta t_g\right]. \tag{3.77}$$

In all cases the eigenvector corresponding to an isotropic spectrum has an eigenvalue $\lambda = \beta_{\rm id}$. In the case of an isotropic back-scatter, the other eigenvalues are all equal to $(\beta_{\rm id} + \beta_{\rm is})$. This decomposition over the two eigenspaces simplifies the solution to

$$N(t+\Delta t_g) = \exp(\beta_{\rm id}\Delta t_g)\overline{N}(t) + \exp\left[(\beta_{\rm id}+\beta_{\rm is})\,\Delta t_g\right]\left[N(t)-\overline{N}(t)\right], \tag{3.78}$$

where $\overline{N}$ is the average over all directions. As a result, for a spatially homogeneous field, the spectrum exponentially tends to isotropy over a time scale $1/(\beta_{\rm id})$.

3.8 Simple ice blocking (`ICO`)

Switch:	`ICO`
Origination:	WAVEWATCH III
Provided by:	H. L. Tolman

Ice covered sea is considered as 'land' in WAVEWATCH III, 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 low energy spectrum is used to assure that spectra are realistic, even for shallow coastal points.

The above discontinuous ice treatment represents the default model setting in WAVEWATCH III. In the framework of the modeling of unresolved obstacles as discussed in Section 3.4.7, a continuous method is also available, as given by Tolman (2003b). In this method, a user-defined critical ice concentration at which obstruction begins ($\epsilon_{c,0}$) and is complete ($\epsilon_{c,n}$) are given (defaults are $\epsilon_{c,0} = \epsilon_{c,n} = 0.5$, i.e., discontinuous treatment of ice). From these critical concentrations, corresponding decay length scales are calculated as

$$l_0 = \epsilon_{c,0} \min(\Delta x, \Delta y), \tag{3.79}$$

$$l_n = \epsilon_{c,n} \min(\Delta x, \Delta y), \tag{3.80}$$

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

$$\alpha_x = \begin{cases} 1 & \text{for} \quad \epsilon\Delta x < l_0 \\ 0 & \text{for} \quad \epsilon\Delta x > l_n \\ \frac{l_n - \epsilon\Delta x}{l_n - l_0} & \text{otherwise} \end{cases}, \quad \alpha_y = \begin{cases} 1 & \text{for} \quad \epsilon\Delta y < l_0 \\ 0 & \text{for} \quad \epsilon\Delta y > l_n \\ \frac{l_n - \epsilon\Delta y}{l_n - l_0} & \text{otherwise} \end{cases}. \tag{3.81}$$

Details of this model can be found in Tolman (2003b).

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.

The above description pertains to the switch `IC0`. Note that either ice transmissions for propagation (`IC0`), or ice as a source term can be used (`IC1`, `IC2`, `IC3`), but not both approaches at the same time.

3.9 Winds and currents

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

$$X_u = \max \left[1.25 \, , \, \frac{u_{10,rms}}{u_{10,l}} \right] \, , \tag{3.82}$$

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

Note that the auxiliary programs of WAVEWATCH III include a program to pre-process input fields (see Section 4.4.7). 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. (3.82).

3.10 Use of tidal analysis

Origination:	WAVEWATCH III
Provided by:	F. Ardhuin

In order to reduce the volume of input files, the water levels and currents can be defined by their tidal amplitudes and phases. This is made possible by

using the `TIDE` switch which activates the detection of the needed information in current.ww3 and level.ww3 files. The tidal analysis can be performed from NetCDF current or water level files, using the ww3_prnc preprocessing program. In that case the analysis method uses the flexible tide analysis package by Foreman et al. (2009). The precomputed tidal constituents can be used at run time by ww3_shel.

However, that method may not be very efficient due to the large memory required to store a large number of tidal constituents because, like other forcing parameters, they are not decomposed across processors: each processor stores the full spatial grid of forcing parameters. To avoid this, the tidal constituents can be used to generated time series with the tidal prediction program ww3_prtide, which produces the usual current.ww3 or level.ww3 files.

The choice of tidal constituents for the analysis and prediction are specified in the input files for ww3_prnc and ww3_prtide. Two short-cuts are defined. `VFAST` is the following selection of 20 components, Z0 (mean), SSA, MSM, MSF, MF, 2N2, MU2, N2, NU2, M2, S2, K2, MSN2, MN4, M4, MS4, S4, M6, 2MS6, and M8. When using `ww3_shel` to do the tidal prediction, the time step for currents or water is set to 1800 s.

In ww3_prtide, there is also a quality check on the values of the tidal constituents that is performed: unrealistically large values of the amplitudes for some constituents can be defined in ww3_prtide.inp. For model grid points where these are exceeded, all components are set to zero, except for UNST grids, in which the neighbors are searched to provide a reasonable value and avoid strong gradients.

3.11 Wave crest and height space-time extremes

Origination:	WAVEWATCH III(ISMAR, NCEP)
Provided by:	Barbariol, F., Benetazzo, A., Alves, J.H.G.M.

Space-Time (ST) extreme waves are modeled in WAVEWATCH III based on the Euler Characteristics (EC) approach, which states that for a given multi-dimensional (2-D space + time), statistically homogeneous and stationary Gaussian random wave field, the probability of exceedance of the maximal sea surface elevation is approximated by means of the mean value of the EC (Fedele et al., 2012). The ST extreme elevation model used here

was formulated by Fedele (2012) for Gaussian sea waves, and extended to second-order nonlinear spatial wave fields by Fedele et al. (2013) and spatio-temporal fields by Benetazzo et al. (2015). The proposed ST extreme linear model was assessed with numerical simulations (Barbariol et al., 2015, 2016), while the extension to second-order nonlinear waves was verified using stereo imaging (Fedele et al., 2013; Benetazzo et al., 2015). According to those models, the probability of exceedance of the second-order nonlinear ST maximal crest height η_{2ST_m} is approximated (for large threshold z_2 with respect to the standard deviation of the surface elevation σ) as:

$$P(\eta_{2ST_m} > z_2) \approx \left[N_{3D} \left(\frac{z_1}{\sigma} \right)^2 + N_{2D} \left(\frac{z_1}{\sigma} \right) + N_{1D} \right] \exp\left(-\frac{z_1^2}{2\sigma^2} \right), \quad (3.83)$$

where the nonlinear threshold z_2 is related to its linear approximation z_1 via the Tayfun quadratic equation using the steepness parameter μ, strictly valid in deep waters, which accounts for bandwidth effects. Parameters N_{3D}, N_{2D}, and N_{1D} express the average number of 3D, 2D, and 1D waves within the ST region, respectively, and are determined from the moments m_{ijl} of the directional wave spectrum $S(k, \theta)$ defined as follows:

$$m_{ijl} = \int k_x^i k_y^j \omega^l S(k, \theta) dk d\theta. \quad (3.84)$$

The average number of waves in Eq. (3.83) also depends on the size of the spatio-temporal domain, namely the spatial dimension X along the mean direction of wave propagation, the spatial dimension Y orthogonal to the mean direction of wave propagation, and the duration D. The expected value $\bar{\eta}_{2ST_m}$ (output parameter **STMAXE**, in meters) of the random variable η_{2ST_m} is given by

$$\bar{\eta}_{2ST_m} = \mathbb{E}\left\{ \eta_{2ST_m} \right\} = \sigma \left[(h_1 + \frac{\mu}{2} h_1^2) + \gamma \left(h_1 - \frac{2N_{3D}h_1 + N_{2D}}{N_{3D}h_1^2 + N_{2D}h_1 + N_{1D}} \right)^{-1} (1 + \mu h_1) \right], \quad (3.85)$$

where $\gamma \approx 0.5772$ is the Euler-Mascheroni constant, and h_1 is the dimensionless (with respect to the standard deviation σ) most probable (mode) extreme value, which is the largest solution of the implicit equation in h

$$\left[N_{3D}h^2 + N_{2D}h + N_{1D} \right] \exp\left(-\frac{h^2}{2} \right) = 1. \quad (3.86)$$

The standard deviation σ_{2_m} (output parameter **STMAXD**, in meters) of the crest height η_{2ST_m} is given by:

$$\sigma_{2_m} = std(\eta_{2ST_m}) = \sigma \frac{\pi}{\sqrt{6}} \left(h_1 - \frac{2N_{3D}h_1 + N_{2D}}{N_{3D}h_1^2 + N_{2D}h_1 + N_{1D}} \right)^{-1} (1 + \mu h_1). \tag{3.87}$$

The expected value of the ST extreme crest-to-trough wave height is obtained using the Quasi-Determinism (QD) model, which predicts the mean shape of ST wave groups close to the apex of their development. According to the QD model the expected value of the crest-to-trough height $\bar{H}_{1_{cm}}$ (output parameter **HCMAXE**, in meters) of the wave with linear extreme crest height $\bar{\eta}_{1ST_m}$ is expressed as

$$\bar{H}_{1_{cm}} = \mathbb{E}\left\{H_{1_{cm}}\right\} = \bar{\eta}_{1ST_m}(1 - \psi_1^*/\sigma^2), \tag{3.88}$$

where $\psi_1^* < 0$ is the value of the first minimum of the temporal autocovariance function computed from the spectrum as

$$\psi_1(\tau) = \int S(\omega) \cos(\omega\tau) d\omega, \tag{3.89}$$

and $\bar{\eta}_{1ST_m}\psi_1^*/\sigma^2 < 0$ is the expected displacement of the wave trough preceding or following the expected linear extreme crest height $\bar{\eta}_{1ST_m}$, which is computed using Eq. (3.85) after letting the wave steepness $\mu = 0$. For a given linear group, the height $\bar{H}_{1_{cm}}$ is generally smaller than the maximum expected wave height $\bar{H}_{1_m}$ (output parameter **HMAXE**, in meters), which is computed as

$$\bar{H}_{1_m} = \mathbb{E}\left\{H_{1_m}\right\} = \bar{\eta}_{1ST_m}\sqrt{2(1 - \psi_1^*/\sigma^2)}. \tag{3.90}$$

The effect on wave heights of second-order nonlinearities is generally small, particularly in narrow band seas, and it will be neglected in the present implementation to reduce the computational cost. Uncertainty of estimates of $\bar{H}_{1_{cm}}$ (output parameter **HCMAXD**, in meters) and $\bar{H}_{1_m}$ (output parameter **HMAXD**, in meters) are determined using the standard deviation of η_{1ST_m} (σ_{1_m}, which is computed using Eq. (3.87) after letting the wave steepness $\mu = 0$) as follows:

$$\begin{aligned} std(H_{1_m}) &= \sigma_{1_m}\sqrt{2(1 - \psi_1^*/\sigma^2)}, \\ std(H_{1_{cm}}) &= \sigma_{1_m}(1 - \psi_1^*/\sigma^2). \end{aligned} \tag{3.91}$$

To activate the computation of wave crest and height space-time extremes in WAVEWATCH III, the user has to specify values of the MISC namelist parameters `STDX`, `STDY` and `STDT` in ww3_grid.inp different to -1 (the latter is a default value that avoids computer overheads when these parameters are not wanted). `STDX` and `STDY` are spatial dimensions over which extremes are calculated. `STDT` is the time length over which extremes are calculated. If `STDX` and `STDY` are left at default values (-1), but `STDT` has a namelist value different to default (e.g., greater than 0), then extreme values are provided over time, for a point. Conversely, if `STDT` is kept at default (-1) and `STDX` and `STDY` are greater than zero, instantaneous extreme values are computed over space. When all three parameters are greater than zero, space-time probabilities and values are computed.

Wave crest and height space-time extremes outputs follow the standard WAVEWATCH III parameter framework, and have to be specified as namelists or flags in ww3_shel.inp or ww3_multi.inp, in which case they are included in the standard gridded binary output files during a model run. Consequently, they also have to be specified in gridded output post-processors for obtaining a final human-readable form. Space-time extremes output parameters available in WAVEWATCH III are provided in Table 3.2.

Internal Label	User-Interface Label	Description
STMAXE	MXE	Max surface elev (STE)
STMAXD	MXES	STD of max crest (STE)
HMAXE	MXH	Max wave height (STE)
HCMAXE	MXHC	Max wvhgt from crest (STE)
HMAXD	SDMH	STD of MXH (STE)
HCMAXD	SDMHC	STD of MXHC (STE)

Table 3.2: User-defined parameters in the computation of wave crest and height space-time extremes.

3.12 Spectral partitioning

Origination:	APL Wave / XWaves / IMEDS
Provided by:	B. Tracy, C. Bunney

The sea-states predicted by numerical wave models are often complex, due to the presence of multiple wave trains formed as a result of both local wind action and more distant storms. In order to better describe these local wind-sea and remote swell components, without dealing with the full complexity of the model wave spectrum, the spectrum can be partitioned into collections of spectral bins from which more recognisable wave statistics (height, period, direction) can be derived.
Fig. 3.7 shows an example surface plot of an energy density spectrum at one grid point at a specific time. The amount of energy density at each frequency-direction intersection is shown by this surface. The surface is divided into shaded areas or partitions representing energy from sub-peaks within the spectrum. Fig. 3.7 shows four spectral partitions, an area of windsea and three swell trains. The total energy represented by this spectrum can be defined by bulk parameters, such as the significant wave height H_s. The shaded areas, called partitions of the spectrum, show spectral sub-features that give more information about this grid point's energy situation. WAVEWATCH III has point and field output options available to provide quantitative descriptions of these individual spectral partition such as partition wave height, peak period of partition (parabolic fit), peak wavelength of partition, mean direction of partition, wind-sea fraction of partition (W) using Eq. (2.251), and the number of partitions. In the field output, these parameters correspond to spectral partitioned output fields 1 through 17 and can be found in Section 2.7.

Across the wave forecasting community, various methods of spectral partitioning and definition of waves as sea or swell have been adopted. Two varieties of partitioning are available from the W3PARTMD module.

1. Topographic partitioning that groups spectral bins together into (multiple) distinct wave systems.

2. A simple frequency based cutoff that produces two partitions; one above and one below a cutoff frquency.

The topographic partitioning methods in WAVEWATCH III are all based on

a watershedding technique using the original partitioning code implemented by B. Tracey (see below). These are further categorised according to the method by which a partition is determined to be either wind-sea or swell. The cutoff methods are varied dependent upon whether the user chooses to use a set frequency (e.g. to delineate between high frequency sea waves and lower frequency swells) or allows the threshold to vary according to the local wind speed and direction.

3.12.1 Topographic partitioning method

Since the two-dimensional spectrum in Fig. 3.7 looks like a topological surface, it is logical to apply an image processing partitioning algorithm that treats the spectral surface like a topographical surface. The partitioning shown in Fig. 3.7 is based on a digital image processing watershed algorithm (Vincent and Soille, 1991) first prototyped by Hanson and Jensen (2004) for the analysis of ocean wave data. The US continental divide where everything to the east goes into the Atlantic Ocean and everything to the west goes into the Pacific Ocean is a typical example of a watershed line. The oceans represent minima that determine the watershed line. If the spectral surface is inverted, the spectral peaks become catchments and watershed lines or partition boundaries can be determined using the Vincent and Soille (1991) algorithm. Calculation of parameters for each spectral partition can then be accomplished and wave system analysis as described in Hanson and Phillips (2001) can be applied. Hanson and Jensen (2004) and Hanson et al. (2006) used a MATLAB code to apply the Vincent and Soille (1991) algorithm[10]. This code has been transformed to an efficient FORTRAN routine for use in WAVEWATCH IIIsince version 3.11. Coding follows the Vincent and Soille (1991) paper but incorporates an efficient sort routine (O(n)) discussed in Tracy et al. (2006).

3.12.2 Sea/swell assignment and partitioning method

The second stage in the partitioning process is the assignment of individual partitions to (wind-)sea and swell categories in point or field outputs. In the default option for WAVEWATCH III these comprise a single wind-sea

[10] Now available as XWaves from http://www.WaveForceTechnologies.com, replacing the previous APL WAVES package

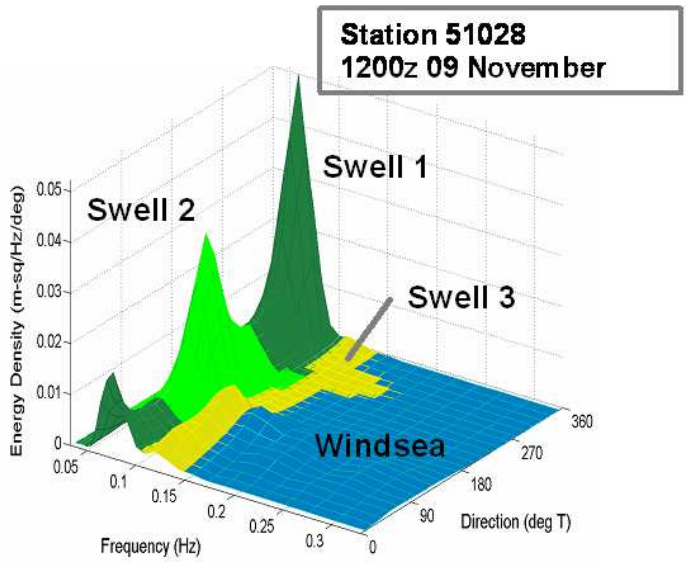

Figure 3.7: Surface plot of an energy density spectrum showing spectral partitions for windsea and three swell trains. This is a snapshot of hindcasted conditions at Christmas Island (NOAA buoy 51028) at 12:00 UTC on November 9, 2000.

(partition 1) and a set of swell components ordered by wave height. Selecting alternative partition methods changes the categorisation as described below.

The partitioning method is set in the MISC namelist in ww3_grid.inp via the PTM parameter:

- PTM=1: Wind sea and swells defined using topographic partitions and classified using a wind sea fraction cutoff (WWIII default scheme). Outputs wind-sea plus swell 1,2,3 etc.
- PTM=2: Wind sea and swells defined using topographic partitions and spectral wave-age cutoff. Outputs wind-sea plus swell 1,2,3 etc.
- PTM=3: Wave components defined using topographic partitions only. Outputs partitions ranked by wave height as partition 1,2,3 etc.
- PTM=4: Wind sea and swell defined using spectral wave-age cutoff. Outputs wind-sea and single swell partition.
- PTM=5: Wave components defined using a user defined frequency cutoff (PTFCUT). Outputs high frequency and low frequency partition.

In PTM1, topographic partitions for which the percentage of wind-sea energy exceeds a defined fraction are aggregated and assigned to the wind-sea component (e.g., the first partition). The remaining partitions are assigned as swell components in order of decreasing wave height.

PTM2 works in a very similar way to PTM1, by first identifying a primary wind-sea component, which is assigned as the first partition, then a number of swell (or secondary wind-sea) partitions are identified, as follows. A set of secondary spectral partitions is established using the topographic method, each partition is checked in turn, with any of their spectral bins influenced by the wind (based on a wave age criterion) being removed and assigned as separate, secondary wind-sea partitions. The latter are by default combined into a single partition, but may remain separate if the namelist parameter FLC is set to ".False.". Swell and secondary wind-sea partitions are then ordered by decreasing wave height. Operational forecasts made at the Met Office suggests that when PTM2 is run with the default single wind-sea partition, this provides a smoother spatial transition between partitions and a more direct link between the primary wind-sea component and wind speed than PTM1. Using the default method, the fraction of wind-sea for all partitions except the primary wind-sea partition should be close to 0.0.

PTM3 does not classify the topographic partitions into wind-sea or swell - it simply orders them by wave height. This approach is useful for producing data for spectral reconstruction applications using a limited number of partitions (e.g. Bunney et al. (2013)), where the classification of the partition as wind-sea or swell is less important than the proportion of overall spectral energy each partition represents.

Methods 4 and 5 do not use the watershedding partitioning method, but adopt a simpler frequency cutoff that produces just two partitions. PTM4 uses the wave age criterion derived from the local wind speed to split the spectrum in to a wind-sea and single swell partition. In this case waves with a celerity greater than the directional component of the local wind speed are considered to be freely propogating swell (i.e. unforced by the wind). This is similar to the method commonly used to generate wind-sea and swell from the WAM model.

PTM5 works in a similar fashion but applies a user defined static frequency cutoff to split the spectrum into a low- and high-band partition. The cutoff frequency is defined as PTFCUT in the MISC namelist and defaults to 0.1Hz. This method is useful if you have a downstream application that defines swell simply as those waves in a spectrum with period greater than a predefined constant value. This is similar to the default partitioning method found in SWAN.

For methods PTM1, PTM2 and PTM4 the wind cutoff can be controlled by modifying the WSM factor in the MISC namelist. This applies a constant multiplier to the wind speed and defaults to 1.0.

At present the outputs using methods PTM1 and PTM2 should be obtained using any of the WAVEWATCH III standard output processing tools. The full range of partitioning methods are available for gridded outputs using the ww3_ounf module, and the netCDF file(s) that are produced will include the partitioning method used within the variable attributes.

Note that for methods PTM4 and PTM5, only two partitions will ever be created, therefore the NOSWLL parameter (defined in the MISC namelist; default=5) will be overriden with a value of 2. Similarly, the wind sea fraction cutoff value WSCUT will be set to 0.0.

Regression test *regtests/ww3_tpt1.1* provides examples of each partitioning method.

3.13 Spatial and temporal tracking of wave systems

Origination:	IFP Swan
Provided by:	Van der Westhuysen, Hanson, Devaliere

The spectral partitioning procedure described above is carried out within the spectral space, independently at each geographical grid point. As a result, there is no coherence between the identified partitions over geographical space and in time. Following Voorrips et al. (1997), Hanson and Phillips (2001) and Devaliere et al. (2009), a spatial correlation step is therefore applied. This is done by means of an outwardly running spiral, originating at an arbitrary point (typically the center) inside the computational domain. Figure 3.8 presents an example of such a tracking spiral on a regular computational grid over a coastal domain featuring landmass. At the spiral origin (location 1), each spectral partition is assigned an initial system index. The spatial correlation is then determined for each subsequent geographical location (2, 3, 4, ...) moving outward along the spiral. At each new geographical location, the peak period T_p, peak direction θ_p and significant wave height H_m0 of each of its spectral partitions are correlated with the spatial means $\tilde{T}^\mathrm{n}_{\mathrm{p},i}$, $\tilde{\theta}^\mathrm{n}_{\mathrm{p},i}$ and $\tilde{H}^\mathrm{n}_{\mathrm{m0},i}$ of the corresponding parameters at its neighboring geographical grid points (indicated by the superscript n) previously assigned a system i. the partition at the present grid point is assigned to the neighboring system i that minimizes the following Goodness-of-Fit (GoF) function:

$$GoF_i = \left(\frac{T_\mathrm{p} - \tilde{T}^\mathrm{n}_{\mathrm{p},i}}{\Delta T_\mathrm{n}} \right)^2 + \left(\frac{\theta_\mathrm{p} - \tilde{\theta}^\mathrm{n}_{\mathrm{p},i}}{\Delta \theta_\mathrm{n}} \right)^2 + \left(\frac{H_\mathrm{m0} - \tilde{H}^\mathrm{n}_{\mathrm{m0},i}}{\Delta H_\mathrm{n}} \right)^2 , \qquad (3.92)$$

where ΔT_n, $\Delta \theta_\mathrm{n}$ and ΔH_n are combining criteria (Van der Westhuysen et al., 2016). If either of the first two terms on the right hand side of (3.92) exceed unity for the closest match, the difference is considered too great and a new wave system is assigned to that partition. Here, the search range for neighboring points is set at 1, so that a maximum of four previously-associated neighbors can be found (e.g. location 15 will have the previously processed neighbors 3, 4, 5 and 14). In some cases, iterative combining is required.

The next step is to correlate these wave systems over time. Each system i at the current time level t is associated with its closest match amongst the systems j at the previous time level $(t-1)$. Three characteristics of

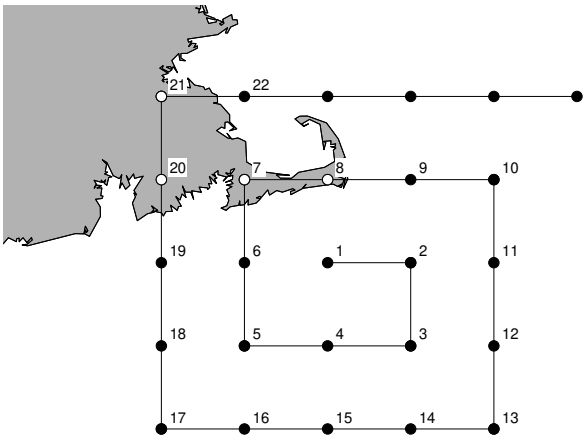

Figure 3.8: Example of a tracking spiral on a regular computational grid over a coastal domain featuring landmass (shaded). Black dots indicate active grid points and white dots indicate inactive (dry) grid points.

---

the wave systems are considered in this process, namely: (i) the spatial mean peak wave period over the system, $\tilde{T}^{\mathrm{s}}_{\mathrm{p},t,i}$, with s denoting the system mean, (ii) the spatial mean peak wave direction, $\tilde{\theta}^{\mathrm{s}}_{\mathrm{p},t,i}$ and (iii) the number of overlapping grid points between the two systems in geographical space $\cap_{i,j}$. These characteristics are combined to form the following GoF function:

$$GoF_{i,j} = \left(\frac{\tilde{T}^{\mathrm{s}}_{\mathrm{p},t,i} - \tilde{T}^{\mathrm{s}}_{\mathrm{p},t-1,j}}{\Delta T_{\mathrm{s}}} \right)^2 + \left(\frac{\tilde{\theta}^{\mathrm{s}}_{\mathrm{p},t,i} - \tilde{\theta}^{\mathrm{s}}_{\mathrm{p},t-1,j}}{\Delta \theta_{\mathrm{s}}} \right)^2 + \left(\frac{N_{t-1,j} - \cap_{i,j}}{0.5 N_{t-1,j}} \right)^2 , \tag{3.93}$$

where ΔT_{s} and $\Delta \theta_{\mathrm{s}}$ are combining criteria, and N is the total number of grid points in a system, see Van der Westhuysen et al. (2016). In order to focus the tracking process on high-energy regions in the wave field, the spatial mean period and peak direction values of each system are weighted with the square of the significant wave height. System i at the current time level t is assigned the system j from the previous time level $(t-1)$ that minimizes (3.93). If any of the three terms on the right hand side of (3.93) exceed unity for the system that minimizes (3.93), a new system number is assigned. For

the last term, this implies a minimum spatial overlap requirement, arbitrarily set at 50%. This term mostly has an impact over basin scale domains, where systems are typically smaller than the computational area. In order to improve robustness, the details of identified systems are stored for five time levels, after which the system association is released.

3.14 Nesting

Origination:	WAVEWATCH III
Provided by:	H. L. Tolman

Traditionally, wave models only consider one-way nesting, with boundary data from low-resolution grids being provided to high-resolution grids. This approach has always been available in WAVEWATCH III, and is discussed in Section 3.14.1. In model version 3.14, a multi-grid wave model driver was introduced, considering full two-way nesting between grids. This approach is discussed in Section 3.14.2. The illustrations below consider regular grids, but the principles discussed are applicable to curvilinear and triangular grids too.

3.14.1 Traditional one-way nesting

The conventional wave model program ww3_shel considers a single wave model grid. This program includes options to transfer boundary conditions from large-scale runs to small-scale runs. Each run can simultaneously 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.

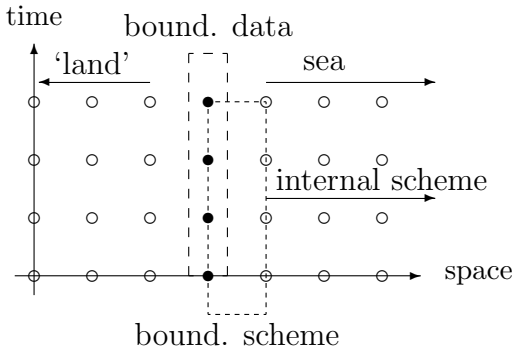

Figure 3.9: Traditional one-way nesting approach as used in ww3_shel. One-dimensional representation in space and time, symbols represent grid points.

The numerical approach for including boundary data in a wave model is illustrated in Fig. 3.9. Active boundary points are assigned in the grid to separate sea points from land points or from otherwise deactivated grid points. Between the active boundary points and sea points, a local boundary scheme is applied (typically first order). In the internal sea points of the model, the selected propagation scheme is used.

Practical aspects of the conventional one-way nesting approach are discussed in more detail in Appendix B.

3.14.2 Two-way nesting

Model version 3.14 includes an option to use the multi-grid or mosaic approach to wave modeling with the program ww3_multi (Tolman, 2006, 2007, 2008b). In this program, an arbitrary number of grids with arbitrary resolutions is considered, with data exchange between grids at each relevant model time step. The grids are given a rank number, where lower rank corresponds to lower resolution, and equal rank corresponds to similar resolution (but not necessarily equal resolution). Three types of data transfer between grids are considered:

- Transfer of data from lower to higher rank grids.

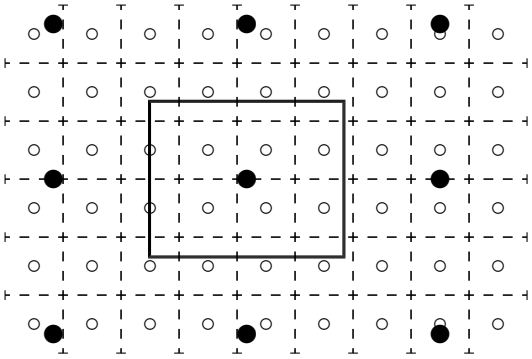

Figure 3.10: Concept for reconciling lower ranked grid with higher ranked grid in two-way nesting approach. ∘ and hashed lines represent the higher ranked grid points and grid boxes, respectively, • and solid lines represent lower ranked grid and central grid box.

- Transfer of data from higher to lower rank grids.
- Transfer of data between grids with equal rank.

Data transfer from lower to higher ranked grids is accomplished by providing boundary data to the higher ranked grid, as in the traditional one-way nesting approach described in the previous section and in Fig. 3.9.

When this approach is combined with data transfer from higher to lower rank, a full two-way nesting approach is established. In ww3_multi the data at the lower ranked grids is reconciled with the data at the higher ranked grids after the higher ranked grids have 'caught up' in time with the lower ranked grids. Considering that the resolution of the lower ranked grid by definition is lower that the resolution of the higher ranked grid, a natural way to estimate the wave energy in the lower ranked grid $E_{l,i}$ from energy in the higher ranked grid $E_{h,j}$ is

$$E_{l,i} = \sum w_{i,j} E_{h,j} \quad , \tag{3.94}$$

where i and j are grid counters in the two grids, and where $w_{i,j}$ are averaging weights. The weights can be defined consistent with conservation of wave energy as the surface of the grid box j in the higher ranked grid that covers

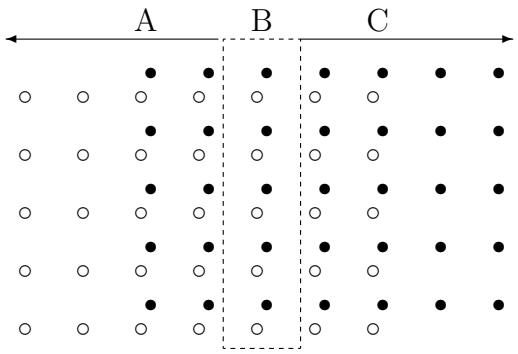

Figure 3.11: Concept for reconciling grids with identical rank and therefore similar resolution. ∘ represents points of grid 1, • represents grid 2.

---

the grid box i in the lower ranked grid, normalized with the surface of the lower ranked grid box i. This is illustrated in Fig. 3.10. To avoid circular reconciliation, grid points in the lower ranked grid that contribute to the boundary data in the higher ranked grid are not updated in this manner.

Overlapping grids with similar rank cannot use the above two-way nesting technique to consistently exchange data. Instead, all such grids are propagated one time step, after which the grids are reconciled as is illustrated in Fig. 3.11. For grid 1 (∘ in Fig. 3.11) two areas can be distinguished. In area C, the influence of the boundary has propagated into the grid since the last reconciliation. The actual depth of penetration depends on the stencil width of the numerical scheme, and the number of propagation time steps. In areas A and B, information from the boundary has not yet penetrated, and this area can be considered as the 'interior' of grid 1. Similarly, area A represents the boundary penetration depth for grid 2 (• in Fig. 3.11) whereas B and C represent the interior of grid 2. A simple and consistent reconciliation between grid 1 and 2 uses data from grid 1 exclusively in area A (interpolating data from grid 1 to grid points in grid 2 as necessary), and uses data from grid 2 exclusively in area C. In area B, where interior parts of both grids overlap, a consistent solution can be found by using weighted averages from both grids. Note that this approach is easily extended to multiple overlapping grids.

Note that for explicit numerical propagation schemes and overlapping

grids with identical resolution and coinciding grid points, solutions for overlapping grids and the compatible single grid can be identical, as long as the overlap areas are sufficiently wide.

The two-way nesting techniques in ww3_multi are largely automated. Each grid is prepared individually, with its own preferred time stepping information. Locations where each grid expects to get boundary data from lower ranked grids are marked as in the one-way nesting approach. All other bookkeeping needed to implement the two-way nesting techniques are automated, although some iterations may be needed to assure that all input boundary points defined in each grid can be provided with boundary data from other grids in the multi-grid application. Alternatively, each grid can obtain data from an external data file as in the traditional nesting approach. In the present implementation, each grid has to obtain all boundary data from a single file, or from other grids in the multi-grid application, but cannot receive data from file and grids simultaneously. Details on the management algorithm developed to run all grid simultaneously can be found in Tolman (2007, section 3.4) and Tolman (2008b), and will not be reproduced here.

Note that the grids used in ww3_multi do not need to have the same spectral discretization. Spectra are converted on the fly in ww3_multi. Details on the numerical techniques used for this approach can be found in Tolman (2007, section 3.5.5). Grid generation for multiple grids in such an approach can be cumbersome, and consistency between grids is required for consistent model results. For this reason automated grid generation utilities have been developed by Chawla and Tolman (2007, 2008).

4 Wave Model Structure and Data Flow

4.1 Program design

The core of WAVEWATCH III is the wave model subroutine, which can be called by either a stand-alone program shell or any other program that requires dynamically updated wave data. Two such programs are provided with the WAVEWATCH III release (e.g., `ww3_shel` and `ww3_multi`). Auxiliary programs include a grid preprocessor `ww3_grid`, a program to generate artificial initial conditions `ww3_strt`, generic program shells for individual `ww3_shel` or multi-grid `ww3_multi` applications, two input pre-processors (`ww3_prep` and `ww3_prnc`), and post-processors for gridded (`ww3_outf` and `ww3_ounf`) and point (`ww3_outp` and `ww3_ounp`) output data.

In this section, note that file names will be identified by the file type font, the contents of a file by the `code` type font and FORTRAN program elements by the FORTRAN type font. The main wave model subroutine is W3WAVE. Data files are identified with the file extension .ww3, except in the multi-grid wave model ww3_multi, where the file extension identifies individual grids part of a chosen multi-grid mosaic. For simplicity, the file extension .ww3 will be used throughout this chapter.

A relational diagram including the basic data flow is presented in Fig. 4.1. The figure illustrates a typical workflow, as follows. The grid pre-processor `ww3_grid` writes a model definition file mod_def.ww3 with bottom and obstruction information and parameter values defining the physical and numerical approaches. The wave model may have cold or hot starts. Hot starting requires a restart file restart.ww3, created either by the wave model itself in a previous run, or by the initial conditions program `ww3_strt`. If a restart file is not available, the wave model will be initialized automatically. If linear growth or spectral seeding is switch on, the model may start from a flat ocean ($H_s = 0$), otherwise the initial conditions will consist of a parametric fetch-limited spectrum based on the initial wind field (see the corresponding option in the initial conditions program).

The wave model routine (W3WAVE) optionally generates up to 9 restart files restart*n*.ww3, where *n* represents a single digit integer number. For telescoping nest applications, the wave model also optionally reads boundary conditions from the file nest.ww3 and generates boundary conditions for con-

secutive nested runs in nest*n*.ww3. The model furthermore dumps raw data to the output files out_grd.ww3 , out_pnt.ww3, track_o.ww3 and partition.ww3 (gridded mean wave parameters, spectra at locations, spectra along tracks, and partitioned wave data, respectively). The tracks along which spectra are to be presented is defined in the file track_i.ww3. Note that the wave model does not write to standard output, because this would be inconvenient if WAVEWATCH III is part of an integrated model. Instead, it maintains its own log file log.ww3 and optionally a test output files test.ww3 for a shared memory version of the model, or test*nnn*.ww3 for distributed memory versions, where *nnn* is the processor number starting with 1. Finally, various output post-processors are available (binary post-processing of raw gridded fields, point output and track output files; NetCDF and GRIB(2) packing of wave data; post-processing for later GrADS graphical processing of gridded and spectral data). A more detailed description of all program elements and their 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 WAVEWATCH III.

Files specific to WAVEWATCH III are opened by name within program subroutines. The unit numbers, however, are defined by the user[11] within each inp, guaranteeing the largest possible flexibility for implementation in integrated models.

In addition to the wave model subroutine, an initialization routine and an interface routine for data assimilation are provided. The routine includes a generic interface that provides all necessary model components to perform full spectral data assimilation. This routine is integrated into the generic wave model shell, which is set up to perform time step managements for a wave model with or without data assimilation. The shell also provides a simple yet flexible way to provide the data assimilation scheme with various types of data. Data assimilation has not yet been included in the multi-grid wave model shell.

[11] Except for ww3_multi.

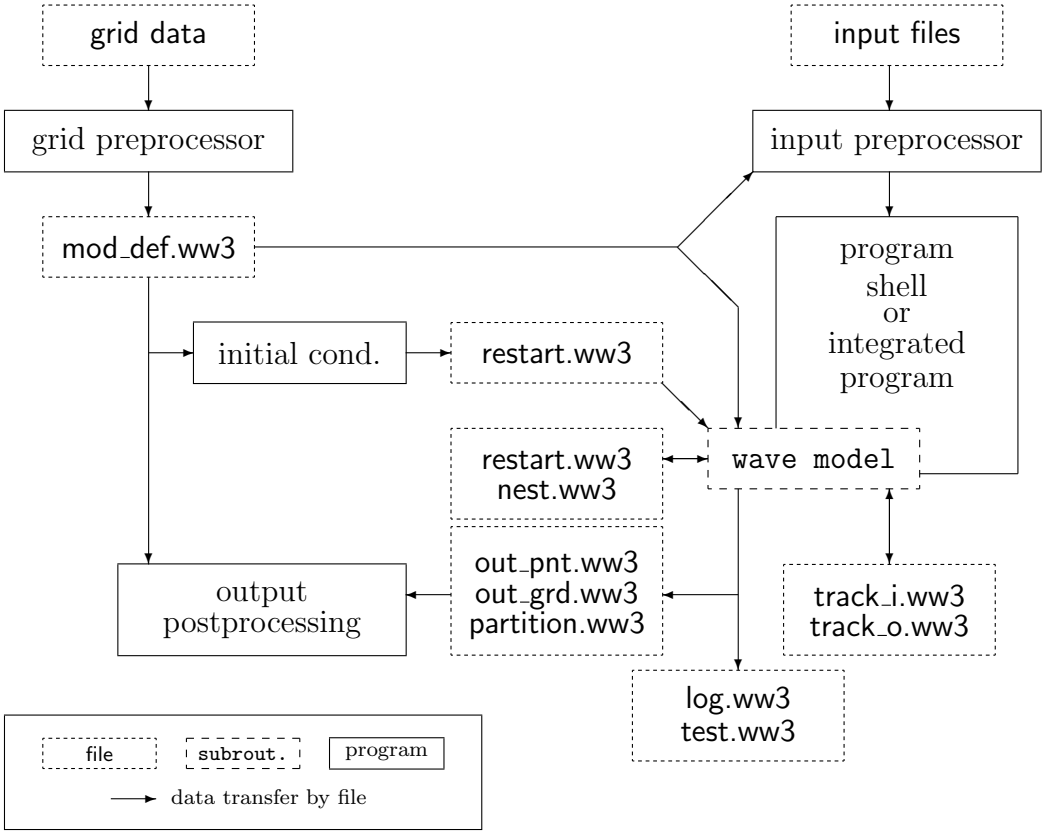

Figure 4.1: Basic program elements and data flow.

4.2 The wave model routines

The wave model driver is a subroutine within the WAVEWATCH III framework package. To run the model driver subroutine, a program shell is needed. WAVEWATCH III is provided with a simple stand-alone shell as will be discussed in Section 4.4.10, and with a more complex multi-grid model shell as will be discussed in Section 4.4.12. The present section concentrates on the wave model driver subroutines.

The wave model initialization routine W3INIT performs model initialization for a single wave model grid. This includes setting up part of the I/O system by defining unit numbers, initializing internal time management, processing the model definition file (mod_def.ww3), processing initial conditions (restart.ww3), preparing model output, and calculating grid-dependent parameters. If the model is compiled for an MPI environment, all necessary communication for both calculations and output are determined and initialized (the model uses persistent MPI communication throughout).

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

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

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

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

first field read, and an `L` identifies the last output. The seven input columns identify boundary conditions (`b`), wind fields (`w`), water levels (`l`), current fields (`c`), ice concentrations (`i`), and data for assimilation (`d`), respectively. Note that data assimilation takes place at the end of the time step after the wave routine call. The seven output columns identify gridded output (`g`), point output (`p`), output along tracks (`t`), restart files (`r`), boundary data (`b`), and partitioned spectral data (`f`), and output for coupling (`c`), respectively.

For the multi-grid wave model (Tolman, 2008b, ww3_multi) a set of routines is build around the basic wave model routines. The three main routines are the initialization routine WMINIT, a time stepping routine WMWAVE and a finalization routine WMFINL, with similar functions as the routines for a single grid as described above. Note that the raw input and output files are generated for separate grid in the mosaic, and are identified by replacing the standard file extension '.ww3' with a unique identifier for each individual grid as chosen by the user in the ww3_grid.inp file. Log files are maintained for each individual grid, as well as an overall log file log.mww3.

4.3 The data assimilation interface

As discussed above, the wave model subroutine is supplemented with a data assimilation interface routine (W3WDAS in w3wdasmd.ftn). This routine is integrated in the stand-alone shell (see Section 4.4.10) to provide time step management of a combined wave model / data assimilation scheme. It has not yet been integrated in the multi-grid model driver, although it is accounted for in the multi-grid model management algorithm.

In this a fairly simple approach is assumed where data assimilation is performed at selected times, while the wave model marches forward in time. In the setup of the shell, the data assimilation is performed after the model has reached the target time, but has not yet produced output. After the data assimilation is performed, the wave model routine is called again only to generate output as requested. Thus, the wave model output for a given time will include the effects of data assimilation for that specific target time.

The generic program shell also processes several types of data to be assimilated, and passes it on to the data assimilation interface routine. All data needs to be preprocessed using the wave model input preprocessor (see Section 4.4.7), and will be recognized by the generic shell by file name. Presently, up to three different data files can be used. Tentatively, these could be mean wave parameters, one dimensional spectral data, and two dimensional spectral data, respectively. This is, however, not hardwired to the model and in fact needs to be defined by the user.

Presently, no data assimilation packages are available. Therefore, user supplied data assimilation schemes are required, and may be included in the wave model using the interface routine (W3WDAS in w3wdasmd.ftn), the documentation of which should be sufficient for the necessary programming. Details on how to add user supplied software to the WAVEWATCH III compilation system can be found in the following chapter. Although NCEP is presently working on wave data assimilation techniques, there are no plans to distribute wave data assimilation software as part of the WAVEWATCH III package.

4.4 Auxiliary programs

4.4.1 General concepts

All auxiliary programs presented here, with the exception of the track output post-processor, read input from a pre-defined input file. Contents of that file determine user choices for each auxiliary program. Comment lines are allowed using a character determined by the user as follows: the first character on the first line of the input file will be considered to be the comment character, identifying comment lines throughout the input file. This comment character has to appear on the first position of input lines to be effective. In all examples in the following sections, the first character of the first line is '$'. Therefore, all lines starting with '$' contain only comments that are not parsed into the auxiliary program. As a standard, auxiliary programs all write formatted output to the standard output unit.

In the following sections, all available auxiliary programs are described using an example input file. These are found in the directory `inp` within the WAVEWATCH III package. Inside each sample input file, all possible options that can be activated by the user are included, most of them feature as comment lines starting with '$'. Files in the current section reflect actual contents of sample files. The sections below also show the name of the executable program associated with the displayed input file, the program name (as appears in the program statement), the source code file and input and output files and their unit numbers (in brackets behind the file name). Input and output files marked with *are optional. The intermediate files mentioned below are all UNFORMATTED, and are not described in detail here. Each file is written and read by a single routine, to which reference is made for additional documentation.

mod_def.ww3	Subroutine W3IOGR (w3iogrmd.ftn).
out_grd.ww3	Subroutine W3IOGO (w3iogomd.ftn).
out_pnt.ww3	Subroutine W3IOPO (w3iopomd.ftn).
track_o.ww3	Subroutine W3IOTR (w3iotrmd.ftn).
restart.ww3	Subroutine W3IORS (w3iorsmd.ftn).
nest.ww3	Subroutine W3IOBC (w3iobcmd.ftn).
partition.ww3	Subroutine W3IOSF (w3iosfmd.ftn).

Preprocessing and compilation of the programs is discussed in the following

two chapters. Examples of test runs of the model are provided with the source code.

4.4.2 Configuration file

All auxiliary programs presented here read parameters and configuration from either a namelist file (new form) or an input file (legacy form). Although most auxiliary programs provide the option of a namelist file, some are still limited to using the legacy input file form. Namelist files for the latter programs will become available in future public releases. For programs which already use namelists, template files are stored in model/nml directory. It is also possible to convert the traditional inp file to new nml format using bash tools stored in the model/aux/bash directory. If inp and nml files are both present in the run directory, the priority will be given to the namelist file. For regtests, the default behavior is to run all the programs with inp file, the option -N must be added to rather use nml file. Some new features are only available with namelists and the future developments will mainly be designed for namelist use.

The namelist configuration file dedicated to a program provides a more readable and adaptative file. A nml file contains one or more namelists with default values for each parameter with the possibility to overwrite them by a user-defined value. There is no mandatory order to defined namelists in the nml file. A namelist starts by '`&SOMETHING_NML`' and stops by '`/`'. If a section is skipped from the namelist file, all the default values will be used. When a program read its namelist file, all the default and user-defined values for all namelists will be outputted in a log file for more tracability.

The traditional way to read the program configuration is from a predefined input file. The first character on the first line of the input file will be considered to be the comment character, identifying comment lines in the input file. This comment character has to appear on the first position of input lines to be effective. In all examples in the following sections lines starting with '`$`' therefore only contain comment. The programs furthermore all write formatted output to the standard output unit.

Below is the part of an inp file with the corresponding part of nml file:

---------- start of example input file (traditional form)

```
$ ------------------------------------------------------------------- $
$ WAVEWATCH III Grid preprocessor input file                           $
$ ------------------------------------------------------------------- $
$ Grid name (C*30, in quotes)
$
  'TEST GRID (GULF OF NOWHERE)   '
$
$ Frequency increment factor and first frequency (Hz) ---------------- $
$ number of frequencies (wavenumbers) and directions, relative offset
$ of first direction in terms of the directional increment [-0.5,0.5].
$ In versions 1.18 and 2.22 of the model this value was by definiton 0,
$ it is added to mitigate the GSE for a first order scheme. Note that
$ this factor is IGNORED in the print plots in ww3_outp.
$
   1.1  0.04118  32  24  0.
$
```

end of example input file (traditional form)

start of example input file (namelist form)

```
! ------------------------------------------------------------------- !
! WAVEWATCH III - ww3_grid.nml - Grid pre-processing                  !
! ------------------------------------------------------------------- !


! ------------------------------------------------------------------- !
! Define the spectrum parameterization via SPECTRUM_NML namelist
!
! * namelist must be terminated with /
! * definitions & defaults:
!     SPECTRUM%XFR   = 0.  ! frequency increment
!     SPECTRUM%FREQ1 = 0.  ! first frequency (Hz)
!     SPECTRUM%NK    = 0   ! number of frequencies (wavenumbers)
!     SPECTRUM%NTH   = 0   ! number of direction bins
!     SPECTRUM%THOFF = 0.  ! relative offset of first direction [-0.5,0.5]
! ------------------------------------------------------------------- !
&SPECTRUM_NML
  SPECTRUM%XFR           =  1.1
  SPECTRUM%FREQ1         =  0.04118
  SPECTRUM%NK            =  32
  SPECTRUM%NTH           =  24
```

```
/
```

---

end of example input file (namelist form)

4.4.3 The grid preprocessor

Program	:	ww3_grid		(W3GRID)
Code	:	ww3_grid.ftn		
Input	:	ww3_grid.nml	(10)	Namelist configuration file. (App. G.1.2)
		ww3_grid.inp	(10)	Traditional configuration file. (App. G.1.1)
		'grid file' *	(user)	File with bottom depths.
		'obstr. file' *	(user)	File with sub-grid obstructions.
		'mask file' *	(user)	File with grid mask.
Output	:	standard out	(6)	Formatted output of program.
		mod_def.ww3	(20)	Model definition file in WAVEWATCH III format.
		mask.ww3 *	(20)	Land-sea mask file (switch O2a).
Scratch	:	ww3_grid.scratch	(90)	Formatted scratch file.

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

4.4.4 The initial conditions program

Program	:	ww3_strt		(W3STRT)
Code	:	ww3_strt.ftn		
Input	:	ww3_strt.inp	(10)	Traditional configuration file. (App. G.2.1)
		mod_def.ww3	(20)	Model definition file.
Output	:	standard out	(6)	Formatted output of program.
		restart.ww3	(20)	Restart file in WAVEWATCH III format.

4.4.5 The boundary conditions program

Program	:	ww3_bound		(W3BOUND)
Code	:	ww3_bound.ftn		
Input	:	ww3_bound.inp	(10)	Traditional configuration file. (App. G.3.1)
		mod_def.ww3	(20)	Model definition file.
		'spectra file' *	(user)	File(s) with wave spectra.
Output	:	standard out	(6)	Formatted output of program.
		nest.ww3	(33)	Boundary conditions file.

4.4.6 The NetCDF boundary conditions program

Program	:	ww3_bounc		(W3BOUNC)
Code	:	ww3_bounc.ftn		
Input	:	ww3_bounc.nml	(10)	Namelist configuration file. (App. G.4.2)
		ww3_bounc.inp	(10)	Traditional configuration file. (App. G.4.1)
		mod_def.ww3	(20)	Model definition file.
		'spectra file' *	(user)	File(s) with wave spectra, in NetCDF.
Output	:	standard out	(6)	Formatted output of program.
		nest.ww3	(33)	Boundary conditions file.

4.4.7 The input field preprocessor

Program	:	ww3_prep		(W3PREP)
Code	:	ww3_prep.ftn		
Input	:	ww3_prep.inp	(10)	Traditional configuration file. (App. G.5.1)
		mod_def.ww3	(11)	Model definition file.
		'user input'*	(user)	See example below.
Output	:	standard out	(6)	Formatted output of program.
		level.ww3*	(12)	Water levels file.
		current.ww3*	(12)	Current fields file.
		wind.ww3*	(12)	Wind fields file.
		ice.ww3*	(12)	Ice fields file.
		data0.ww3*	(12)	Assimilation data ('mean').
		data1.ww3*	(12)	Assimilation data ('1-D spectra').
		data2.ww3*	(12)	Assimilation data ('2-D spectra').

Note that the optional output files are specific to ww3_shel and ww3_multi, but are not processed by the actual wave model routines. These files are consequently not needed if the wave model routines are used in a different shell or in an integrated program. However, the routines reading and writing these files are system-independent and could therefore be used in customized applications of the basic wave model. The reading and writing of these files is performed by the subroutine W3FLDG (w3fldsmd.ftn). For additional documentation and file formats reference if made to this routine.

4.4.8 The NetCDF input field preprocessor

Program	:	ww3_prnc		(W3PRNC)
Code	:	ww3_prnc.ftn		
Input	:	ww3_prnc.nml	(10)	Namelist configuration file. (App. G.6.2)
		ww3_prnc.inp	(10)	Traditional configuration file. (App. G.6.1)
		mod_def.ww3	(11)	Model definition file.
		'user input'*	(user)	See example below.
Output	:	standard out	(6)	Formatted output of program.
		level.ww3*	(12)	Water levels file.
		current.ww3*	(12)	Current fields file.
		wind.ww3*	(12)	Wind fields file.
		ice.ww3*	(12)	Ice fields file.
		data0.ww3*	(12)	Assimilation data ('mean').
		data1.ww3*	(12)	Assimilation data ('1-D spectra').
		data2.ww3*	(12)	Assimilation data ('2-D spectra').

See note at the end of the previous section (4.4.7) for tools that can be used to pack input files in custom programs.

4.4.9 The tide prediction program

Program	:	ww3_prtide		(W3TIDE)
Code	:	ww3_prtide.ftn		
Input	:	ww3_prtide.inp	(10)	Traditional configuration file. (App. G.7.1)
		mod_def.ww3	(20)	Model definition file.
		current.ww3_tide or level.ww3_tide	(user)	File with tidal constituents.
Output	:	standard out	(6)	Formatted output of program.
		current.ww3 or level.ww3	(33)	Level or current forcing.

The user-provided file current.ww3_tide or level.ww3_tide is a binary file that can be obtained by running ww3_prnc with the 'AT' option and then renaming the resulting file current.ww3 or level.ww3 into current.ww3_tide or level.ww3_tide . The choice of tidal constituents used for the tidal prediction can be a subset of the ones present in these files or all of them.

Because of wetting and drying or grid mismatches, the tidal constituents may be erroneous or absent for some of the WAVEWATCH III nodes. The erroneous ones can be detected using a maximum amplitude on particular components. When the amplitudes exceeds these maxima, then the tidal constituents are extrapolated from the nearest nodes. This feature has only been tested on triangular meshes.

4.4.10 The generic shell

Program	:	ww3_shel		(W3SHEL)
Code	:	ww3_shel.ftn		
Input	:	ww3_shel.nml	(10)	Namelist configuration file. (App. G.8.2)
		ww3_shel.inp	(10)	Traditional configuration file. (App. G.8.1)
		mod_def.ww3	(30)	Model definition file.
		restart.ww3	(30)	Restart file.
		nest.ww3*	(33)	Boundary conditions file.
		level.ww3*	(11)	Water levels file.
		current.ww3*	(12)	Current fields file.
		wind.ww3*	(13)	Wind fields file.
		ice.ww3*	(14)	Ice fields file.
		data0.ww3*	(15)	Assimilation data.
		data1.ww3*	(16)	Assimilation data.
		data2.ww3*	(17)	Assimilation data.
		track_i.ww3*	(22)	Output track information.
Output	:	standard out	(6)	Formatted output of program.
		log.ww3	(20)	Output log of wave model (see Section 4.2).
		test.ww3*	(6/21)	Test output of wave model.
		restart*n*.ww3*	(30)	Restart file(s).
		nest*n*.ww3*	(34-42)	Nesting file(s).
		out_grd.ww3*	(31)	Raw output of gridded fields.
		out_pnt.ww3*	(32)	Raw output of spectra.
		track_o.ww3*	(23)	Raw output of spectra along tracks.
Scratch	:	ww3_shel.scratch	(90)	Formatted scratch file.

4.4.11 Automated grid splitting for ww3_multi (ww3_gspl)

Program	:	ww3_gspl		(W3GSPL)
Code	:	ww3_gspl.ftn		
Input	:	ww3_gspl.inp	(10)	Traditional configuration file. (App. G.9.1)
		mod_def.*xxx*	(11)	Model definition file of grid to be split.
Output	:	standard out	(6)	Formatted output of program.
		xxx.bot	(11)	File with bathymetry for sub-grid.
		xxx.obst	(11)	File with obstructions for sub-grid.
		xxx.mask	(11)	File with mask for sub-grid.
		xxx.tmpl	(11)	ww3_grid.inp for sub-grid.
		ww3_multi.*xxx.n*	(11)	Template for part of ww3_multi.inp that needs to be modified.
		ww3.ww3_gspl	(35)	GrADS file with map of sub-grids (with switch O16).
		ww3.ctl	(35)	GrADS map control file (O16).

To further automate the splitting of the grid, a script ww3_gspl.sh is provided. This script runs ww3_gspl, and subsequently generated the mod_def files for all sub-grids. If a file ww3_multi.inp is provided, then this file is updated too. The workings of the script are shown with the -h command line flag, which results in the output of the script as shown in Fig. 4.3.

```
Usage: ww3_gspl.sh [options]  gridID nr_grid
Required:
  gridID     : name of master grid to be split up
  nr_grid    : number of sub-grids to be generated
Options:
  -a               : use entire assigned cummunicator for each grid
  -h               : help, print this.
  -i               : create template file ww3_gint.inp_tmpl for
                     later integration of output into single grid.
  -d data_dir      : directory with ww3_grid.inp and ancilary data
                       * default is working directory
                       * relative unless starting with '/'
  -e halo_ext      : set halo extension, default is 2
  -o output_dir    : directory for std out redirects
                       * default is working directory
                       * relative unless starting with '/'
  -n n_iter        : maximum number of  interations in ww3_gspl
                       * default = 350
  -t target        : target accuracy in ww3_gspl (%)
                       * default = 0.75
  -f comm_first    : communicator fraction (first).
                       * default = 0.
  -l comm_last     : communicator fraction (last).
                       * default = 1.
  -s ww3_multi.inp : name of input file to be modified.
                       * Not set as default.
  -r               : replace file defined under -s, otherwise add .new
  -v               : verbose, show program output
```

Figure 4.3: Options for ww3_gspl.sh, as obtained by running it with the -h command line option.

4.4.12 The multi-grid shell

Program	:	ww3_multi		(W3MLTI)
Code	:	ww3_multi.ftn		
Input	:	ww3_multi.nml	(8)	Namelist configuration file. (App. G.10.2)
		ww3_multi.inp	(8)	Traditional configuration file. (App. G.10.1)
Output	:	standard out	(6)	Formatted output of program.
		log.mww3	(9)	Output log of wave model driver.
		test.mww3*	(auto)	Test output of wave model.

This wave model program requires and produces a plethora of input and output files consistent with those of ww3_shel in Section 4.4.10, where file extensions .ww3 are replaced by an identifier for a specific grid. Note that all files are opened by name, and that the unit number assignment is dynamic and automatic.

In order to make all existing features available there is a new version of the input file that uses namelists. This is the version that will be supported in the future as it allows a more flexible addition of new features. **Please note that the namelist form is not supported by GCC compilers before version 4.8.2.**

4.4.13 Grid Integration

Program :	ww3_gint		(W3GINT)
Code :	ww3_gint.ftn		
Input :	ww3_gint.inp	(10)	Traditional configuration file. (App. G.11.1)
	mod_def.*	(20)	Model definition files in WAVEWATCH III format for base and target grids
	out_grd.*	(30+)	Gridded field files in WAVEWATCH III format for base grids
Output :	standard out	(6)	Formatted output of program.
	out_grd.*	(30+)	Gridded field files in WAVEWATCH III format for target grid

This post processor program takes field data from several overlapping grids and produces a unified output file. The different model definition and field output files are identified by the unique identifier associated with each specific grid. At this moment the program works with curvilinear and rectilinear grids. A weights file WHTGRIDINT.bin is written that can be read in subsequent runs using identical origin-destination grids, saving substantial time in cases using large number of input grids and/or high-resolution target grids.

Note that this program can be used in concert with the grid splitting program ww3_gspl, and that ww3_gspl.sh has an option to produce a template input file for his program (see Section 4.4.11).

4.4.14 Gridded output post-processor

Program	:	ww3_outf		(W3OUTF)
Code	:	ww3_outf.ftn		
Input	:	ww3_outf.inp	(10)	Traditional configuration file. (App. G.12.1)
		mod_def.ww3	(20)	Model definition file.
		out_grd.ww3	(20)	Raw gridded output data.
Output	:	standard out	(6)	Formatted output of program.
		...*	(50)	Transfer file.

The extension of the file name of transfer files for ITYPE = 3 identifies the content of the file. The file extension for each data type is given in Table 4.1 on page 221.

4.4.15 Gridded output NetCDF post-processor

Program	:	ww3_ounf		(W3OUNF)
Code	:	ww3_ounf.ftn		
Input	:	ww3_ounf.nml	(10)	Namelist configuration file. (App. G.13.2)
		ww3_ounf.inp	(10)	Tradition configuration file. (App. G.13.1)
		mod_def.ww3	(20)	Model definition file.
		out_grd.ww3	(20)	Raw gridded output data.
		NC_globatt.inp	(994)	Additional global attributes.
Output	:	standard out	(6)	Formatted output of program.
		*.nc	()	NetCDF file

When a single field is put in the file, the abbreviated field name (file extensions from ww3_outf) for each data type is given in Table 4.1 on page 221.

4.4.16 Gridded output post-processor for GrADS

Program	:	gx_outf		(GXOUTF)
Code	:	gx_outf.ftn		
Input	:	gx_outf.inp	(10)	Traditional configuration file. (App. G.14.1)
		mod_def.ww3	(20)	Model definition file.
		out_grd.ww3	(20)	Raw gridded output data.
Output	:	standard out	(6)	Formatted output of program.
		ww3.grads	(50)	GrADS data file.
		ww3.ctl	(51)	GrADS control file.

This post-processor generates input files with gridded model parameters for the Grid Analysis and Display System (GrADS, Doty, 1995). Although GrADS can also work with GRIB files, the present preprocessor is preferable, as the data file also gives access to a land-sea-ice map.

4.4.17 Gridded GRIB output post-processor

Program	:	ww3_grib		(W3GRIB)
Code	:	ww3_grib.ftn		
Input	:	ww3_grib.inp	(10)	Traditional configuration file. (App. G.15.1)
		mod_def.ww3	(20)	Model definition file.
		out_grd.ww3	(20)	Raw gridded output data.
Output	:	standard out	(6)	Formatted output of program.
		gribfile	(50)	GRIB file.

This post-processor packs fields of mean wave parameters in GRIB format, using GRIB version II and NCEP's w3 and bacio library routines, or in GRIB2, using NCEPS's operational package. Additional packing data can be found in Table 4.1 on page 221.

The GRIB packing is performed using the NCEP's GRIB tables as described in NCEP (1998). Because the w3 and bacio routine are not fully portable, they are not supplied with the code. The user will have to provide corresponding routines. It is suggested that such routines are activated with additional WAVEWATCH III switches in the mandatory switch group containing the 'NOGRB' switch, as if presently the case with the NCEP routines. The GRIB2 packing is performed according to WMO (2001), and is performed with NCEP's standard operational packages.

Table 4.1 shows the KPDS(5) data values for GRIB packing. For the partitioned data, the first number identifies the wind sea, the second number identifies swell. Most data are packed as surface data (KPDS(6) = 0). For the partitioned swell fields, however, consecutive fields are packed at consecutive levels, with the level type indicator set to (KPDS(6) = 241). KPDS(7) identifies the actual level or swell field number.

Table 4.1 shows several KPDS data values for GRIB2 packing. The first number in the table represents LISTSEC0(2), which identifies the discipline type (e.g., oceanography, meteorology, etc.) The second number represents KPDS(1), which identifies the parameter category (e.g., waves, circulation, ice, etc.) within the discipline type. The third number represents KPDS(2), which identifies the actual parameter. For the partitioned data, A/B means A for wind sea and B for swell. Additionally KPDS(10) = 0 for surface data, and KPDS(10) = 241 to pack consecutive swell fields at consecutive levels.

KPDS(12) identifies the actual level or swell field number.

Although the above input file contains flags for all 31 output fields of WAVEWATCH III, not all fields can be packed in GRIB. If a parameter is chosen for which GRIB packing is not available, a message will be printed to standard output. Table 4.1 shows which parameter can be packed in GRIB. Note that at NCEP the conversions from GRIB to GRIB2 coincided with the introduction of partitioned wave model output. This required some duplicate definitions in GRIB and some apparent inconsistencies between GRIB and GRIB2 packing.

4.4.18 Point output post-processor

Program	:	ww3_outp		(W3OUTP)
Code	:	ww3_outp.ftn		
Input	:	ww3_outp.inp	(10)	Traditional configuration file. (App. G.16.1)
		mod_def.ww3	(20)	Model definition file.
		out_pnt.ww3	(20)	Raw point output data.
		NC_globatt.inp	(994)	Additional global attributes.
Output	:	standard out	(6)	Formatted output of program.
		tab*nn*.ww3 *	(*nn*)	Table of mean parameters where *nn* is a two-digit integer.
		. . . *	(user)	Transfer file.

In previous releases of WAVEWATCH III spectral bulletins were generated using spectral data transfer file generated with ITYPE = 1 and OTYPE = 3 and the w3split program (see section 5.2). This is an obsolescent code that is produced here for backward compatibility only. This program reads the following five records from standard input (no comment lines allowed) :

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

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

4.4.19 Point output NetCDF post-processor

Program	:	ww3_ounp		(W3OUNP)
Code	:	ww3_ounp.ftn		
Input	:	ww3_ounp.nml	(10)	Namelist configuration file. (App. G.17.2)
		ww3_ounp.inp	(10)	Traditional configuration file. (App. G.17.1)
		mod_def.ww3	(20)	Model definition file.
		out_pnt.ww3	(20)	Raw point output data.
Output	:	standard out	(6)	Formatted output of program.
		...*	(user)	Transfer file.

4.4.20 Point output post-processor for GrADS

Program	:	gx_outp		(GXOUTP)
Code	:	gx_outp.ftn		
Input	:	gx_outp.inp	(10)	Traditional configuration file. (App. G.18.1)
		mod_def.ww3	(20)	Model definition file.
		out_pnt.ww3	(20)	Raw point output data.
Output	:	standard out	(6)	Formatted output of program.
		ww3.spec.grads	(30)	GrADS data file with spectra and source terms.
		ww3.mean.grads	(31)	File with mean wave parameters.
		ww3.spec.ctl	(32)	GrADS control file.

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

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

4.4.21 Track output post-processor

Program	:	ww3_trck		(W3TRCK)
Code	:	ww3_trck.ftn		
Input	:	ww3_trck.inp	(10)	Traditional configuration file. (App. G.19.1)
		track_o.ww3	(11)	Raw track output data.
Output	:	standard out	(6)	Formatted output of program.
		track.ww3	(51)	Formatted data file.

This post-processor will convert the raw track output data 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, oceanographic 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 varying in time and position, the following records are printed :

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

4.4.22 Track output NetCDF post-processor

Program :	ww3_trnc		(W3TRNC)	
Code :	ww3_trnc.ftn			
Input :	ww3_trnc.nml	(10)	Namelist configuration file.	
(App. G.20.2)				
	ww3_trnc.inp	(10)	Traditional configuration file.	
(App. G.20.1)				
	track_o.ww3	(11)	Raw track output data.	
Output :	standard out	(6)	Formatted output of program.	
	*.nc	()	NetCDF file.	

This post-processor will convert the raw track output data to a NetCDF file. The output NetCDF file contains the following variables :

- time in days since 1990-01-01, Julian days at UTC time.
- frequencies of each frequency bin. (radian)
- frequencies of lower band of each frequency bin. (radian)
- frequencies of upper band of each frequency bin. (radian)
- frequencies width of each frequency bin. (radian)
- directions of each direction bin. (oceanographic convention)
- latitudes and longitudes along time dimension. (degree)

For each output point varying in time and position, the following records are printed :

- track name (32 characters)
- the entire spectrum. (m2 s rad-1)
- water depth (m), current and wind u and v components (m s-1), friction velocity (m s-1), air-sea temperature difference (degree centigrade).

4.4.23 Spatial and temporal tracking of wave systems

Program	:	ww3_systrk		(W3SYSTRK)
Code	:	ww3_systrk.ftn		
Input	:	ww3_systrk.inp	(10)	Traditional configuration file. (App. G.21.1)
		partition.ww3	(11)	Spectral partition file.
		sys_restart.ww3*	(12)	Restart file with system memory.
		sys_mask.ww3*	(13)	Mask file.
Output	:	sys_log.ww3	(20)	Output log (appended with processor number in parallel run).
		sys_coord.ww3	(21)	Lat/lon coordinates of fields.
		sys_hs.ww3	(22)	Significant wave height fields of individual wave systems.
		sys_tp.ww3	(23)	Peak period fields of individual wave systems.
		sys_dir.ww3	(24)	Peak direction fields of individual wave systems
		sys_dspr.ww3	(25)	Direction spread fields of individual wave systems.
		sys_pnt.ww3	(26)	Point output file for significant wave height, peak period, and peak direction.
		sys_restart1.ww3	(27)	Restart file.
		*.nc	( )	NetCDF file.

Program currently implemented for regular grids only. The spatial and temporal tracking is performed on the basis of the spectral partition data file. Both the time interval and geographic domain over which wave systems are tracked can be subsets of the data contained in the partition file. The combining parameters `dirKnob` and `perKnob` are used to influence the strictness of the system combining algorithm in geographic space, and `dirTimeKnob` and `perTimeKnob` are the corresponding parameters in temporal space. Lower values imply stricter criteria, which results in smaller, more numerous systems. This also typically increases the processing time. Recommended values are given above. These values can be influenced locally, for example around an island, by defining a mask file sys_mask.ww3. Parameters `hsKnob` and `wetPts` are a low-energy and small system filters—all wave systems with an average

H_{m0} below `hsKnob` or with a size of less than `wetPts`*100% of the overall domain size are purged. Parameters `seedLat` and `seedLon` influence the origin of the wave system search spiral, with default at the center of model domain (indicated by `0. 0.`). At the end of a tracking run, the end state of system memory is stored in sys_restart1.ww3. This file, renamed as sys_restart.ww3, can be used to restart a tracking sequence from this previous system memory state.

4.4.24 The Restart File Processor

Program	:	ww3_uprstr		(WW3UPRSTR)
Code	:	ww3_uprstr.ftn		
Input	:	ww3_uprstr.inp	(10)	Traditional configuration file. (App. G.22.1)
		mod_def.ww3	(20)	Model definition file.
		restart.ww3	( )	Restart File
		XXXX.grbtxt	(123)	SWH Analysis file in grbtxt format.
Output	:	restart001.ww3	( )	Updated restart file.

Introduction

The majority of the observations for the sea surface wave field is observations of the diagnostic variable: Significant Wave Height (SWH). Therefore, the wave data assimilation (WDA) often takes space in the SWH space and subsequently this information has to be transfered to the prognostic space, wave spectrum (WS) to be imposed as boundary and/or initial condition (BIC).

Purpose of the ww3_uprstr

Redistribution of the energy from the analysis of the SWH field to the field of WS saved in the restart file.

Core algorithm

The **ww3_uprstr** sets the SWH of the background spectra equal to the SWH of the analysis and modifies the shape of the spectrum according to the user's prescibed spectrum shape. The **ww3_uprstr** has been implemented as an extension of the restart reader and it requires as inputs: the restart file, the SWH of the analysis, and the **ww3_uprstr.inp** (see above) with the user's defined options; additional files may be required in order to reduce the calculations on the fly.

How to Use the ww3_uprstr

To use the **ww3_uprstr**, the users have to follow the same logic as for all the WAVEWATCH III programs. In summary:

1. **Download the source code**
   The **ww3_uprstr** source code is included to the official WAVEWATCH III release, starting with the version x.xx.

2. **Compile the code**
   The **ww3_uprstr** is compiled the same way as all the auxiliary programs of WAVEWATCH III, see the appropriate section of the manual. For debugging outputs, use the **T** flag at the switch file.

3. **Test the executable**
   The regression test **ww3_ta1** can be used for testing the different options of **ww3_uprstr**.

4. **Run the ww3_uprstr**
   Description of steps to run the **ww3_uprstr**:

   Required Input files:

   - **Restart file.** This file has been created by WW3 during the hindcast run of the model or during the previous cycle. Expected filename: **restart.ww3**.
   - **mod_def.ww3**
   - **ww3_uprstr.inp.** This is the input file for the **ww3_uprstr**. The users have to define i. the date of the assimilation, ii. the method of energy redistribution and iii. depending on the method: a percentage or an inputfile.
   - **Input file of analysis.** The file can have any name, but the suffix defines the reader used for importing the data. Currently, the reader supports only **grbtxt** format. This is a text file created with wgrib2 from the grib2 file of the analysis and it has the following structure:

NX NY
VAL0001
VAL0002
...
VAL(NX*NY)

- To run the executable:
  >$EXE/ww3_uprstr
  If all the inputs are correctly prepared, a new restart file (**restart001.ww3**) will be created. The **ww3_uprstr** exports the updated spectra in the same format as the restart.ww3. To be applied as BIC for the initialization of the next prediction cycle, it has to be renamed:
  >mv **restart001.ww3 restart.ww3**)
  The updated restart file is used as normally.

Tips

- The restart file has to be created with the same WW3 version as the **ww3_uprstr**; there is not backwards compatibility.
- The starting time of the assimilation defined at **ww3_uprstr.inp** has to be the same with the time at the restart file.
- By using the **T**, the **ww3_uprstr.inp** exports the fields of SWH from the background restart file, the analysis and of the updated restart file. In addition, the spectra from the restart files before and after and the update are exported as text files.

Update method

The users have to define the update algorithm of their choice at the **ww3_uprstr.inp**. The options for updating the restart file are defined at the ww3_uprstr.inp with the flag UPD[N], where N could be 0F, 0C, 1, 2,... For UPDN, with N <2, the same correction is applied to the whole grid; Expected input: PRCNTG, as defined at fac. For UPDN, with N >1 each gridpoint has its own update factor and the input is at grb2txt format. For more details about the current implementation see the 4.4.

The following UPD options are available:

1. UPD0C:: **ELIMINATED from version x.xx** Option 0C All the spectra are updated with a constant:
   $fac = (SWH\_Bckg - SWH\_Anl)/SWH\_Anl.$

2. UPD0F:: Option 0F All the spectra are updated with a constant:
   $fac = SWH\_Anl/SWH\_Bckg.$

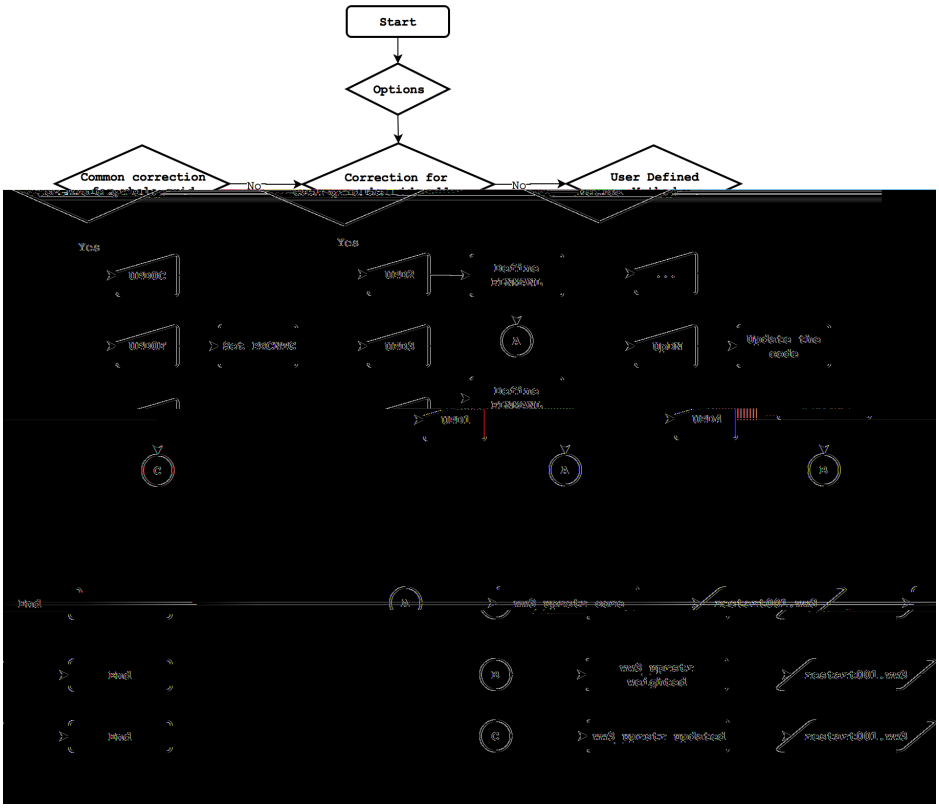

Figure 4.4: Flowchart of the implemented methods for updating the wave spectra at the WW3 restart file. Additional methods can be implemented by adding UPD options to the namelist.

3. UPD1 :: **ELIMINATED from version x.xx** Option 1 The fac(x,y,frq,theta), is weighted according to the % of energy at each spectral bin; fac the same as UPDOF.

4. UPD2 :: Option 2 The fac(x,y,frq,theta), is calculated at each grid point according to SWH_Bckg and SWH_Anl

5. UPD3 :: Option 3 The update factor is a surface with the shape of the background spectrum.

6. UPD4 :: [NOT INCLUDED in the current version, just keeping the spot] Option 4 The generalization of the UPD3. The update factor is the sum of surfaces which are applied on the background spectrum. The algorithm requires the mapping of each partition on the individual spectra; the map is used to determine the weighting surfaces.

Any additional method for the redistribution of the energy to the WS could be added by extending the input file and adding the source code to the **ww3_uprstr.ftn**.

Example

In this section, an example of the simplest WDA application is discussed. The figure 4.5 shows how the **ww3_uprstr** is used in the framework of a simple wave analysis system.

A WW3 run (from the previous cycle or from the hindcast) provides the background field of SWH and the corresponding restart file at the appropriate time. The format of the background SWH field has to be compatible with the WDA module inputs.

The WDA module uses the background field and the available observations for the time of analysis, produces the analysis and exports the field of SWH in grbtxt format (**XXXX.grbtxt**) .

The analysis file, the **mod_def.ww3**, the **restart.ww3** file and the **ww3_uprstr.inp** are the input files for the **ww3_uprstr**. If all the options and input files are correctly prepared, it takes approxmately one minute to update a grid of 260000 grid nodes and generate the output on a single processor. The updated restart file has to be renamed, at the expected file name, in the case of this example to **restart.ww3**.

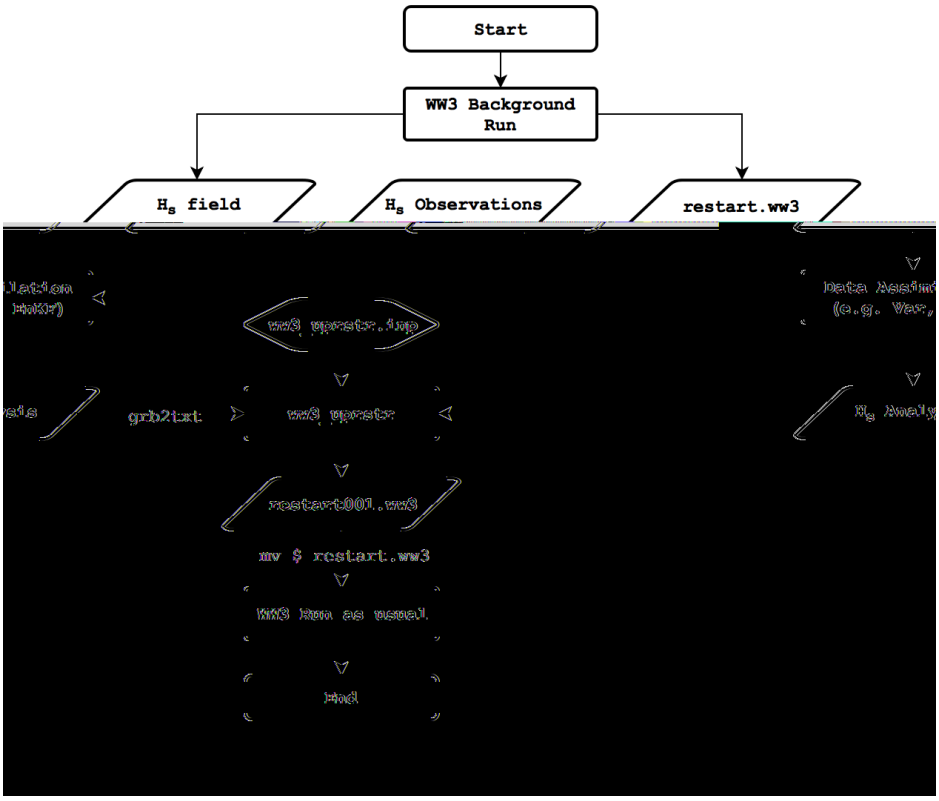

Figure 4.5: Flowchart of simplified wave data assimilation system, showing the role of the ww3_uprstr, the required input files, and the resulted output of the updated restart file.

Note: All NCEP's WDA systems use GRIB2 format, thereore there is always an intermediate step to transfer the grib files to the appropriate format. The used software is WGRIB2 and more information can be retrieve from the official website.

group	field	description extension	file data	GRIB1 data	GRIB2
1	1	depth	.dpt	–	–
1	2	mean current components	.cur	–	–
1	3	wind speed	.wnd	32	0,2,1
		wind direction		31	0,2,0
		wind u		33	0,2,2
		wind v		34	0,2,3
1	4	air-sea temp. dif.	.dt	–	–
1	5	water level	.wlv	–	10,3,1
1	6	ice coverage	.ice	91	10,2,0
2	1	wave height H_s	.hs	100	10,0,3
2	2	mean wave length	.l	–	–
2	3	mean wave period $T_{m0,2}$	.t02	–	–
2	4	mean wave period $T_{m0,1}$	.t	103	10,0,15
2	5	mean wave period $T_{m0,-1}$	.tm1	–	–
2	6	peak frequency f_p	.fp	108	10,0,11
2	7	mean wave direction θ_m	.dir	101	–
2	8	directional spread σ	.spr	–	–
2	9	peak direction θ_p	.dp	107	10,0,10
4	1	H_s of partition	.phs	102,105	10,0,5/8
4	2	T_p of partition	.ptp	110,106	10,0,6/9
4	3	L_p of partition	.plp	–	–
4	4	θ_m of partition	.pdir	109,104	10,0,4/7
4	5	σ of partition	.psi	–	–
4	6	wind sea fraction of part.	.pws	–	–
4	7	total wind sea fraction	.wsf	–	–
4	8	number of partitions	.pnr	–	–
5	1	friction velocity comp.	.ust	–	–
5	2	Charnock par for air side	.cha	–	–
5	3	Energy flux $\int C_g E(f) df$	.CgE	–	–
5	4	Wind to wave energy flux	.faw	–	–
5	5	Wave-supported stress	.taw	–	–
5	6	Upward wave-supported stress	.twa	–	–
5	7	Whitecap coeverage	.wcc	–	–
5	8	Avg whitecap foam thickness	.wcf	–	–
5	9	Sign breaking wave height	.wch	–	–
5	10	Whitecap moment	.wcm	–	–

Table 4.1: Field output post processors ancillary data.

group	field	description	file extension	GRIB1 data	GRIB2
6	1	radiation stress	.Sxy	–	–
6	2	Breaking wave momentum flux	.two	–	–
6	3	Bernoulli head	.J	–	–
6	4	Breaking wave energy flux	.foc	–	–
6	5	Stokes transport	.tus	–	–
6	6	Surface Stokes drift	.uss	–	–
6	7	Second order pressure at $k = 0$	.p2s	–	–
7	1	near-bottom amplitude	.cfd	–	–
7	2	near-bottom velocity	.ubr	–	–
7	3	bedform parameters	.bed	–	–
7	4	Energy flx to bot bound layer	.fbb	–	–
7	5	Momentum flx to bot bound layer	.tbb	–	–
8	1	mean square slopes	.mss	–	–
8	2	Phillips constant	.msc	–	–
9	1	average time step	.dtd	–	–
9	2	cut-off frequency f_c	.fc	–	–
9	3	cut-off frequency f_c	.fc	–	–
9	4	maximum CFL for X-Y advection	.cfx	–	–
9	5	maximum CFL for θ advection	.cfd	–	–
9	6	maximum CFL for k advection	.cfk	–	–
10	1	user defined #1	.us1	–	–
10	2	user defined #2	.us2	–	–

Table 4.1, continued.

5 Install, Compile and Run the wave model

5.1 Introduction

WAVEWATCH III is written in ANSI standard FORTRAN-90, with no machine-dependent elements, so that WAVEWATCH III can be installed without modifications on most platforms. WAVEWATCH III utilizes its own preprocessor 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 WAVEWATCH III, but complicates its installation. To minimize complications, a set of UNIX/Linux scripts is provided to automate the installation in general and the use of the preprocessor in particular. Note this option is not supported for other operation systems like MS products. If the code is to be compiled on one of the latter platforms, it is suggested to extract a working code in a UNIX/Linux environment using the utility `w3_source` (see below), and then to port this clean code to the platform of choice.

WARNING

If version 6.07 is implemented as an upgrade to previous versions of WAVEWATCH III, please note that this version may not be compatible with previous model versions. It is therefore prudent *NOT* to install the new version of WAVEWATCH III on top of the old version.

WARNING

5.2 Distribution

Through version 5.16, WAVEWATCH III public releases were released as tarballs. After the public release of version 6.07, WAVEWATCH III will move to an open development paradigm and source code is distributed via Github.

5.3 Installing

To clone the source code

```
git clone https://github.com/NOAA-EMC/WW3
```

Public releases can be accessed by checking out tags:

```
git checkout VERTAG
```

where VERTAG is the public release tag, for example v6.07.

5.4 Setting up

After cloning the WAVEWATCH III repo, there are two steps to follow. The first step is to run the script model/bin/w3_setup which will compile auxiliary programs and setup the environment file wwatch3.env, which will be stored locally in the bin directory. This file sets options for default C and FORTRAN compilers which will only be used by the preprocessor. Except in specific cases, this default setting can be used as is. Running the script alone will give full usage, or to execute also supply the source_dir, which is the location of the model directory. If you run from the top level directory:

```
./model/bin/w3_setup model
```

The setup can be modified by rerunning the w3_setup program, or by manually editing the setup file. The 'home' directory of WAVEWATCH III can only be changed by editing or removing the local wwatch3.env.

Note that previous versions of WAVEWATCH III required defining the An advanced option is to define the path of wwatch3.env in the user environment variable `WWATCH3_ENV`. This is no longer needed. In addition, there is no longer an option of a global installation.

The second set of setting up WAVEWATCH III is running the script /model/bin/ww3_from_ftp.sh to obtain binary or large files that are not stored in the git repository. From the top directory:

```
./model/bin/ww3_from_ftp.sh
```

There are three prompts requiring user input in this script. The first asks for the path the the top level directory, the second asks if you want to delete the tar file that is copied (likely the answer is n for no) and the last asks to keep a folder that is used by this script (likely the answer is n for no).

5.5 Directory Structure

After cloning the WAVEWATCH III repository, the following directories are found on the top level.

manual	Manual directory.
model	Model source code, build scritps, etc.
regtests	WAVEWATCH III executables.
guide	Guide for WAVEWATCH III .
smc_docs	Documentation and files for SMC grids.

The following directories exist in the 'model' directory of WAVEWATCH III.

aux	Raw auxiliary programs (source codes etc.).
bin	Executables and shell scripts for compiling and linking.
ftn	Source code and makefile.
inp	Input files.
nml	Namelist input files.

The following directories are generated in the 'model' directory after compiling the code.

exe	WAVEWATCH III executables.
mod	Module files.
obj	Object files.

The aux directory has various additional tools, see the actual directory for its contents. These include contributed *Matlab* codes, *IDL* tools, *bash* scripts and *GrADS* and *FORTRAN* programs.

Setting up WAVEWATCH III runs the installation the auxiliary programs will first process *FORTRAN* codes, using the compiler as defined in the setup

file wwatch3.env. Note that these codes are still in fixed format FORTRAN-77. The executables are stored in the directory bin. A more detailed description of these programs (including instructions on running the executables) can be found in the documentation included in the above source code files. After the compilation of these programs, several UNIX shell scripts and auxiliary files are installed in the bin directory.

w3adc.f	WAVEWATCH III FORTRAN preprocessor.
w3prnt.f	Print files (source codes) including page and line numbers.
w3list.f	Generate a generic source code listing.
w3split.f	Generate spectral bulletin identifying individual wave fields within a spectrum from the spectral output of the point output post-processor (see Section 4.4.18). This is a legacy code superseded by generating bulletins directly from ww3_outp. It is retained here for historical reasons only.

The bin directory has various UNIX scripts to manage the WAVEWATCH III framework. The use of these scripts is explained in Section 5.7. Note that the following scripts acquire setup information from the WAVEWATCH III environment setup file defined by `WWATCH3_ENV`. Either defined by w3_setenv which read the local wwatch3.env or by the user environment file.

Main programs used :

install_ww3_tar	Script to install WAVEWATCH III from tar files.
w3_setup	Script for creating/editing the WAVEWATCH III environment setup file. The default setup file is bin/.wwatch3.env. switch and compiler are given in arguments. (see options)
w3_clean	Script to clean up WAVEWATCH III directories by removing files generated during compilation or test runs. 3 levels of clean-up. (see options)
w3_make	Script to compile and link components of WAVEWATCH III using a makefile. A list of programs can be given in arguments.

w3_automake	Script to automatically compile separately sequential and distributed programs in MPI, OMP or HYB depending on the switch file content. A list of programs can be given in arguments.
w3_new	Script to touch all correct source code files to account for changes in compiler switches in combination with the makefile.
ww3_gspl.sh	Script to automate use of ww3_gspl program (see Section 4.4.11).
arc_wwatch3_tar	Program to archive versions of WAVEWATCH III in the directory arc.
ww3_from_ftp.sh	Script to download all the binary files not provided by git.

Examples files provided :

switch_*xxx*	Examples of preprocessor switches provided by users or developers. (Section 5.9)

Subprograms called :

w3_setenv	Script to setup the environment based on bin/.wwatch3.env. Called by all auxiliary programs.
comp.tmpl	Compiler script template comp used with cmplr.env.
link.tmpl	Link script template link used with cmplr.env.
cmplr.env	compiler options tested on intel, mpt, gnu, pgi with optimized and debugging options. used with comp.tmpl and link.tmpl
make_makefile.sh	Script to generate the makefile based on selections in the file switch.
ad3	Script to run the preprocessor w3adc and the compile script comp for a given source code file.
ad3_test	Test version of ad3, showing modifications to original source file. This script does not compile code.
all_switches	Generates a list of all w3adc switches present in the source code files.
find_switch	Script to find WAVEWATCH III source code files containing compiler switches (or arbitrary strings).

sort_switch	Order the switches present in the switch file.
sort_all_switches	Script to search all the switch files in WAVEWATCH III and call sort_switch.

Extra programs :

comp.*xxx*	Compiler script comp for advanced setting.
link.*xxx*	Link script link for advanced setting.
make_MPI	Script to separately compile MPI and non-MPI programs. (obsolete in future releases - use w3_automake)
make_OMP	Script to separately compile OpenMP and single threaded programs. (obsolete in future releases - use w3_automake)
make_HYB	Script to separately compile hybrid MPI-OpenMP and single threaded programs. (obsolete in future releases - use w3_automake)
ln3	Script to make symbolic link of source code file to work directory. (not used)
list	Script to print source code listing using w3prnt. (not used)
w3_source	Script to generate a true FORTRAN source code for any of he WAVEWATCH III program elements. (not used)
WW3_switch_process	Perl script to process switch from a source file. (not used)

After installation in the bin directory, several GrADS scripts are installed in the aux directory.

cbarn.gs	Semi-standard GrADS script for displaying color bars.
colorset.gs	Script to define colors used in shading.
profile.gs	Script to display profiling data generated by ww3_multi.
source.gs	Script for composite plot of spectra and source terms (2-D polar or Cartesian plots in color or in black and white).

1source.gs	Script to plot single source term.
spec.gs	Script to plot spectra.
spec_ids.gen	Data file used by spectral / source scripts.

5.6 Optional environment settings

Compilation of the WAVEWATCH III NetCDF enabled programs requires the environment variable `WWATCH3_NETCDF` be set to either `NC3` (compile with NetCDF version 3.x) or `NC4` (compile with NetCDF version 4.x). If the script variable is set to `WWATCH3_NETCDF = NC3`, then the following environment variables are required

`NETCDF_LIBDIR`	Path to where the NetCDF-3 libraries are installed.
`NETCDF_INCDIR`	Path to where the NetCDF-3 include files are installed.

If `WWATCH3_NETCDF = NC4`, then the following environment variable is required.

`NETCDF_CONFIG`	Path to the NetCDF-4 nc-config utility program.

The nf-config utility program (part of the NetCDF-4 install or nc-config for old versions) is used to determine the appropriate compile and link flags for the `WWATCH3_NETCDF = NC4` compile. The NetCDF-4 compile requires NetCDF version 4.1.1 or higher. Use the command

```
nf-config --version
```

to check the version of the installed Fortran-NetCDF library. Compiling with the `NC4` switch requires `WWATCH3_NETCDF = NC4` and the NetCDF-4 installation compiled with the NetCDF-4 API enabled. Use

```
nf-config --has-nc4
```

to check if the installed NetCDF has the NetCDF-4 API enabled.

Compilation of the WAVEWATCH III with PDLIB for Domain Decompstion option for (Explicit/Implicit) triangular unstructed grids requires the environment variable `METIS_PATH` be set to the path where `Metis` and `ParMetis` are compiled using the same compiler used for WW3.

Two additional remarks need to be made regarding parallel versions of the model (OpenMP and MPI versions). First, complications may occur when

preparing executables for running in an MPI environment. Such complications are discussed in Appendix C. Secondly, the OpenMP code should be compiled using directives only, i.e., do not use compiler options that automatically thread the code.

5.7 Compiling and linking

For the first compilation, the WAVEWATCH III environment must be set up using the script w3_setup with the model directory path in argument. Some options are available to define the compiler options and the switch file located in the bin directory. For instance,

```
w3_setup /home/user/WW3/model -c <comp> -s <switch>
```

the `<comp>` keyword can be `mpt`, `intel`, `gfortran`, `pgi` for optimized compilation options or could be `mpt_debug`, `intel_debug`, `gfortran_debug`, `pgi_debug` for debugging compilation options. If system-dependant options are needed, it can be done by modifying the script cmplr.env. Some old comp/link templates from different clusters are still available but not recommended. The `<switch>` keyword can be the suffix of a provided switch file or your own one. Running this script will create these three scripts/files :

comp	Compiler script. This script is based on comp.tmpl with the provided definition of the compiler and its options.
link	Linker script. This script is based on link.tmpl with the provided definition of the linker and its options.
switch	File containing a list of switches as recognized by the preprocessor w3adc. Copied from `switch_<switch>`.

The environment file wwatch3.env will be created or updated if it already exists. The auxiliary FORTRAN programs for code prepocessor will be compiled using by default the GNU FORTRAN compiler which can be different to the provided compiler for the WAVEWATCH III programs.

The easiest way to compile WAVEWATCH III is using the script w3_automake to automatically detect which programs to compile and to compile it. It will force pre- and post-processing programs to be compiled as sequential implementation and others depending on the switches from openMP, MPI or hybrid configurations. If the netCDF library is correctly set up, the netCDF dedicated programs will be also compiled. The same test is done for SCRIPNC, PDLIB, OASIS, ESMF, TRKNC and TIDE keywords to manage the program compilation in the best way. For instance,

```
w3_automake
```

or for a few programs

```
w3_automake ww3_grid ww3_shel ww3_ounf
```

the compiled programs will all be stored in the exe directory with a copy of the switch, comp and link files used.

In case of troubles during the compilation of a program, the log files will be stored in a temporary directory which will differs between sequential mode (tmp_SEQ) and distributed mode like MPI (tmp_MPI), OMP (tmp_OMP) or hybrid (tmp_HYB). There are usually three log files per program:

`ww3_<prog>.1`	full program with only the matching lines of codes based on the switches and at the end the compilation command line.
`ww3_<prog>.out`	warning messages
`ww3_<prog>.err`	error messages

To remove all user-created files from the WAVEWATCH III framework, the script w3_clean can be used by only cleaning up the model directory

```
w3_clean -m
```

or the full ww3 repository

```
w3_clean -c
```

5.8 Detailled compilation

Compilation of WAVEWATCH III is performed using the script w3_make in the bin directory[12]. If this script is used without parameters, all basic programs of WAVEWATCH III are compiled. Optionally, names of programs to be compiled can be given as part of the compile command. For instance

```
w3_make ww3_grid ww3_strt
```

will compile the grid preprocessor and the initial conditions program only. w3_make uses several of the scripts described in the previous section. A graphical representation is given in Fig. 5.1. If necessary, the script w3_make uses the scripts make_makefile.sh to generate a makefile. make_makefile.sh generates a list of modules to be linked, based on the program switches in the file switch (see Section 5.9), and checks all needed sources for module dependencies. If switches have been changed since the last call to w3_make, w3_new is used to 'touch' relevant source code or to delete relevant object files. After the makefile has been completed, the standard UNIX make utility is used to compile and link the programs. Instead of directly using the FORTRAN compiler, the makefile invokes the preprocessor and compile scripts ad3 and comp, and the link script link. The script ad3 uses the extension of the file name to determine the necessary action. Files with extension .ftn are processed by w3adc, files with extension .f or .f90 are send to the script comp directly. Although a user could try out several of these scripts interactively, he or she generally needs to run w3_make only.

WARNING

The auxiliary scripts w3_make etc. use the switch, comp and link files from the ./bin directory under the WAVEWATCH III home directory, *NOT* from the local directory.

WARNING

After the appropriate changes have been made, or the appropriate example scripts have been copied in, (parts of) WAVEWATCH III can be compiled

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

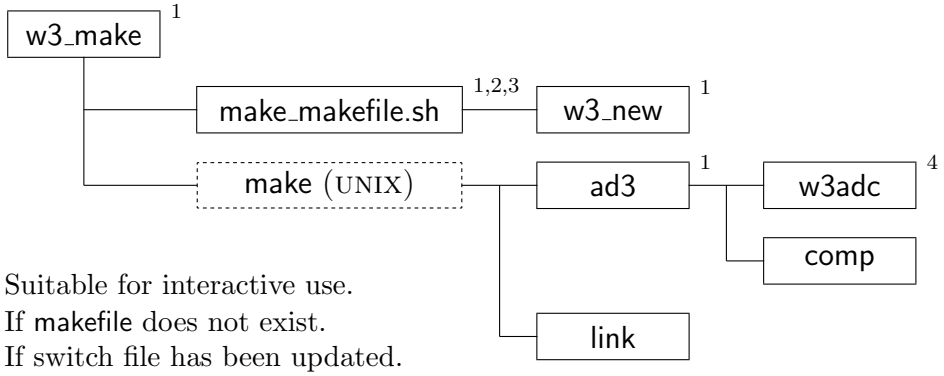

Figure 5.1: General layout of the compiler program w3_make.

---

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

```
ln3 ctest
```

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

```
ad3_test ctest
```

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

```
ad3 ctest 1
```

which invokes both the preprocessor w3adc and the compile script comp. The 1 at the end of this line activates test output. If it is omitted, this command should result in a single line of output, identifying that the routine is being processed. If ad3 works as expected, an object file obj/ctest.o is generated. If requested during the initial set up, a source code and listing file (ctest.f and

ctest.l) can be found in the scratch directory. The listing file is also retained if compilation errors are detected by comp. At this time, it is prudent to test error capturing in the script comp by adding errors and warnings to ctest.ftn in the work directory. The error capturing is discussed in some detail in the documentation of comp. After comp has been tested, and the errors in ctest.ftn have been removed, the link to the work directory and the file obj/ctest.o can be deleted.

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

```
w3_make ww3_grid
```

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

```
ww3_grid
```

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

```
rehash
```

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

```
w3_clean
```

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

```
w3_new all or w3_new
```

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

5.9 Selecting model options

The file switch in the bin directory 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 5.9.1. Other switches are optional, and are described in Section 5.9.2. Default model setting are identified in Section 5.9.3. The order in which the switches appear in switch is arbitrary. How these switches are included in the source code files is described in Section 6.2.

5.9.1 Mandatory switches

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

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

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

SHRD	Shared memory model.
DIST	Distributed memory model.

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

Selection of propagation schemes and GSE alleviation method. These represent two sets of switches with some shared switches between the groups. Note that the second set of switches is secondary to the selection of program modules in the first set of switches, and therefore, does not have a user-defined option.

PR0	No propagation scheme / GSE alleviation used.
PR1	First order propagation scheme, no GSE alleviation.
PR2	Higher-order schemes with Booij and Holthuijsen (1987) dispersion correction.
PR3	Higher-order schemes with Tolman (2002a) averaging technique.
PRX	Experimental (user supplied).

PR0	No propagation scheme used.
PR1	First-order propagation scheme.
UNO	Second-order (UNO) propagation scheme.
UQ	Third-order (UQ) propagation scheme.

Selection of flux computation:

FLX0	No routine used; flux computation included in source terms,
FLX1	Friction velocity according to Eq. (2.56).
FLX2	Friction velocity from Tolman and Chalikov input.
FLX3	Idem, with cap of Eq. (2.78) or (2.79).
FLX4	Friction velocity according to Eq. (2.138).
FLXX	Experimental (user supplied).

Selection of linear input:

LN0	No linear input.
SEED	Spectral seeding of Eq. (3.70).
LN1	Cavaleri and Malanotte-Rizzoli with filter.
LNX	Experimental (user supplied).

Selection of input and dissipation. STABn switches are optional and additional to corresponding STn switch:

ST0	No input and dissipation used.
ST1	WAM3 source term package.
ST2	Tolman and Chalikov (1996) source term package. See also the optional STAB2 switch.
STAB0	No stability correction. Compatible with any source term (ST) package. Including this switch has no effect.

STAB2	Enable stability correction (2.95) - (2.98). Compatible with ST2 only.
ST3	WAM4 and variants source term package.
STAB3	Enable stability correction from Abdalla and Bidlot (2002). Compatible with ST3 and ST4 only.
ST4	Ardhuin et al. (2010) source term package.
ST6	BYDRZ source term package.
STX	Experimental (user supplied).

Selection of nonlinear interactions:

NL0	No nonlinear interactions used.
NL1	Discrete interaction approximation (DIA).
NL2	Exact interaction approximation (WRT).
NL3	Generalized Multiple DIA (GMD).
NL4	Two-scale approximation (TSA).
NLX	Experimental (user supplied).

Selection of bottom friction:

BT0	No bottom friction used.
BT1	JONSWAP bottom friction formulation.
BT4	SHOWEX bottom friction formulation.
BT8	Dalrymple and Liu formulation (fluid mud seafloor).
BT9	Ng formulation (fluid mud seafloor).
BTX	Experimental (user supplied).

Selection of term for damping by sea ice:

IC0	No damping by sea ice.
IC1	Simple formulation.
IC2	Liu et al. formulation.
IC3	Wang and Shen formulation.
IC4	Frequency-dependent damping by sea ice.

Selection of term for scattering by sea ice:

IS0	No scattering by sea ice.
IS1	Diffusive scattering by sea ice (simple).
IS2	Floe-size dependent scattering and dissipation.

Selection of term for reflection:

REF0	No reflection.
REF1	Enables reflection of shorelines and icebergs

Selection depth-induced breaking of :

DB0	No depth-induced breaking used.
DB1	Battjes-Janssen.
DBX	Experimental (user supplied).

Selection of triad interactions:

TR0	No triad interactions used.
TR1	Lumped Triad Interaction (LTA) method.
TRX	Experimental (user supplied).

Selection of bottom scattering:

BS0	No bottom scattering used.
BS1	Magne and Ardhuin.
BSX	Experimental (user supplied).

Selection of supplemental source term:

XX0	No supplemental source term used.
XXX	Experimental (user supplied).

Selection of method of wind interpolation (time):

WNT0	No interpolation.
WNT1	Linear interpolation.
WNT2	Approximately quadratic interpolation.

Selection of method of wind interpolation (space):

WNX0	Vector interpolation.
WNX1	Approximately linear speed interpolation.
WNX2	Approximately quadratic speed interpolation.

Selection of method of current interpolation (time):

CRT0	No interpolation.

CRT1	Linear interpolation.
CRT2	Approximately quadratic interpolation.

Selection of method of current interpolation (space):

CRX0	Vector interpolation
CRX1	Approximate linear speed interpolation.
CRX2	Approximate quadratic speed interpolation.

Switch for user supplied GRIB package.

NOGRB	No package included.
NCEP1	NCEP GRIB1 package for IBM SP.
NCEP2	NCEP GRIB2 package for IBM SP.

5.9.2 Optional switches

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

O0	Output of namelists in grid preprocessor.
O1	Output of boundary points in grid preprocessor.
O2	Output of the grid point status map in grid preprocessor.
O2a	Generation of land-sea mask file mask.ww3 in grid preprocessor.
O2b	Output of obstruction map in grid preprocessor.
O2c	Print status map in format as read by ww3_grid.
O3	Additional output in loop over fields in field preprocessor.
O4	Print plot of normalized one-dimensional energy spectrum in initial conditions program.
O5	Id. two-dimensional energy spectrum.
O6	Id. spatial distribution of wave heights (not adapted for distributed memory).
O7	Echo input data for homogeneous fields in generic shell.
O7a	Diagnostic output for output points.
O7b	Idem in ww3_multi.
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.
- o10 Identify main elements of multi-grid model extensions in standard output.
- o11 Additional log output on management algorithm in log.mww3.
- o12 Identify removed boundary points in overlapping grids (center).
- o13 Identify removed boundary points in overlapping grids (edge).
- o14 Generate log file with buoy data buoy_log.ww3 for output type `ITYPE = 0` in ww3_outp.
- o15 Generate log file with time stamps of input data file times.XXX in ww3_prep.
- o16 Generate GrADS output of grid partitioning in ww3_gspl.

The following switches enable parallelization of the model using OpenMP directives, also known as 'threading'. Before model version 5.01, threading and parallelization using the MPI switch could no be used simultaneously. With version 5.01, pure MPI,pure OMP and hybrid MPI-OMP approaches became available. Switches used in version 5.01 and higher are not compatible with switches used in previous model versions.

- OMPG General loop parallelization directives used for both exclusive OpenMP parallelization and hybrid MPI-OpenMP parallelization.
- OMPX Idem, but for directives used only for exclusive OpenMP parallelization.
- OMPH Idem, but for directives used only for hybrid MPI-OpenMP parallelization.
- PDLIB Domain Decomposition for Explicit and Implicit Solver on triangular unstructured grids. (`ParMetis` is required for this option)

Note that these switches can only be used in certain combinations, as enforced in the model installation scripts (particularly make_makefile.sh. A pure MPI approach requires the DIST and MPI switches. A pure OpenMP approach requires the SHRD, OMPG and OMPX switches, and the hybrid approach requires the DIST, MPI, OMPG, and OMPH switches.

The following switches are associated with the continuously moving grid options. The first switch activates the option, the other two are optional additions.

MGP	Activate propagation correction in Eq. (3.45).
MGW	Apply wind correction in moving grid approach.
MGG	Activate GSE alleviation correction in Eq. (3.48).

The following compiler dependent switches are available. They may not have been maintained for recent compiler versions.

C90	Compiler directives for Cray C90 (vectorization).
NEC	Compiler directives for NEC SX6/SX8 (vectorization).

Furthermore the following miscellaneous switches are available:

ARC	Arctic grid option for SMC grid[13].
COU	Activates the calculation of variables required for coupling
DSS0	Switch off frequency dispersion in diffusive dispersion correction.
FLD1	Sea-state dependent τ Reichl et al. (2014) (Section 2.5.2).
FLD2	Sea-state dependent τ Donelan et al. (2012) (Section 2.5.3).
IG1	Second-order spectrum and free infragravity waves (Section 2.4.9.
MLIM	Use Miche-style shallow water limiter of Eq. (3.71).
MPIBDI	Experimental parallelization of multi-grid model initialization.
MPIT	Test output for MPI initializations.
MPRF	Profiling of individual models and nesting in ww3_multi.
NC4	Activates the NetCDF-4 API in the NetCDF pre- and post-processing programs.
NCC	NCEP coupler.
NCO	Code modifications for operational implementation at NCO (NCEP Central Operations). Mostly changes unit numbers and file names. Not recommended for general use.
NLS	Activate nonlinear smoother (Section 2.3.6).

[13] Not yet fully tested according to author.

NNT	Generate file test_data_*nnn*.ww3 with spectra and nonlinear interactions for training and testing of NNIA.
OASIS	Initializes OASIS Coupler (App. E.3).
OASACM	OASIS atmospheric model coupling fields(App. E.3).
OASOCM	OASIS oceanic model coupling fields (App. E.3).
OASICM	OASIS sea ice model coupling fields (App. E.3).
REFRX	Enables refraction based on spatial gradients in phase velocity (Section 2.4.3)
REFT	Test output for shoreline reflection (which is activated with REF1).
RTD	Rotated grid option.
RWND	Correct wind speed for current velocity.
S	Enable subroutine tracing in the main WAVEWATCH III subroutines by activating calls to the subroutine STRACE.
SCRIP	Enable SCRIP remapping routines (App. D.3)
SCRIPNC	Enable storage of remapping weights in NetCDF files (App. D.3)
SEC1	Enable the use of global time steps less than 1 s, but does not allow output at time steps less than 1 s.
SMC	Activate SMC grid.
T	Enable test output throughout the program(s).
Tn	Id.
TDYN	Dynamic increment of swell age in diffusive dispersion correction (test cases only).
TIDE	Enables tidal analysis: used for pre-processing of input files, run-time tidal prediction in ww3_shel or tidal prediction with ww3_prtide.
TIDET	test output for tidal analysis.
TRKNC	Activates the NetCDF API in the wave system tracking post-processing program. Selecting TRKNC alone will generate NetCDF-3 files. Selecting both TRKNC and NC4 will generate NetCDF-4 files.
UOST	Enable the unresolved obstacles source term.
XW0	Swell diffusion only in ULTIMATE QUICKEST scheme.
XW1	Id. wave growth diffusion only.

5.9.3 Default model settings

Up to model version 3.14, the NCEP operational model setup was considered as the default model setup. However, with subsequent versions of WAVEWATCH III, the model has evolved into a modeling framework rather than a single model. With this, WAVEWATCH III is run differently at various centers, and a clear "default" model version can no longer be identified. Nevertheless, in order to be able to concisely identify in publications exactly which model setup is used, "default" configurations of various centers are now provided in the bin directory. These configurations are provided in example switch files and README files, such as switch_NCEP_st2 and README.NCEP. Note that these files are provided to simplify referring to model version, but do not imply an endorsement of the specific model configuration.; in this context, it should be noted that by nature, model versions at operational centers are in a continuous state of development.

5.10 Modifying the source code

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

```
ln3 filename
```

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

Modifying source codes is straightforward. Adding new switches to existing subroutines, or adding new modules requires modification of the automated compilation scripts. If a new subroutine is added to an existing

module, no modifications are necessary. If a new module is added to WAVEWATCH III, the following steps are required to include it in the automatic compilation:

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

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

1) Put switch in required source code files.
2) If the switch is part of a new group of switches, add a new 'keyword' to w3_new.
3) Update files to be touched in w3_new if necessary.
4) Update make_makefile.sh with the switch and/or keyword.

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

1) Update files to be touched in w3_new.

If WAVEWATCH III 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 WAVEWATCH III 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 WAVEWATCH III home directory, renaming them to the file names expected by the install program, and running the install program.

For co-developers using the NCEP svn repository, changes in the code should be made using the best practices as outlined in (Tolman, 2014c).

5.11 Running test cases

If WAVEWATCH III is installed and compiled successfully, it can be tested by running different program elements interactively. To do this, create a top level directory called work and copy the input files in the model/inp or model/nml directories. A generic switch file is no longer provided, but a switch file can be chosen from the bin directory and then test the example input files. It should therefore be possible to run all model elements by typing

```
ww3_grid | more
ww3_strt | more
ww3_bound | more
ww3_prep | more
ww3_shel | more
ww3_outf | more
ww3_outp | more
ww3_ounf | more
ww3_ounp | more
ww3_trck | more
ww3_grib | more
gx_outf | more
gx_outp | more
```

where the `more` command is added to allow for on-screen inspection of the output. This `| more` can be replaced by redirection to an output file, e.g.

```
ww3_grid > ww3_grid.out
```

Note that `ww3_grib` will only provide GRIB output if a user-supplied packing routine is linked in. Note furthermore that no simple interactive test case

for ww3_multi is provided. GrADS can then be run from the work directory to generate graphical output for these calculations. All intermediate output files are placed in the work directory, and can be removed conveniently by typing

```
w3_clean
```

Up to version 3.14, WAVEWATCH III was provided with a set of simple tests to established assess the proper behavior of the basic functionality of the model. In the early development of the next release of the model, Erick Rogers and Tim Campbell converted these in regression tests that could be run more easily in an automated version. Up to model version 4.06, these modified tests were gathered in the nrltest directory, while keeping the old tests in the test directory. In model version 4.07, the nrltest were adopted as the new test cases for WAVEWATCH III in a new regtests directory, while eventually the remaining real-world test cases in test were moved to the cases directory, while discontinuing the test directory completely. The following regression tests are available in the regtests directory.

ww3_tp1.1	1D propagation around the world along the equator (no land).
ww3_tp1.2	1D propagation, along meridian (no land).
ww3_tp1.3	1D propagation, shoaling test.
ww3_tp1.4	1D propagation, spectral refraction (x).
ww3_tp1.5	1D propagation, spectral refraction (y).
ww3_tp1.6	1D propagation, wave blocking by current.
ww3_tp1.7	1D propagation, IG wave generation.
ww3_tp1.8	1D propagation, wave breaking on a beach.
ww3_tp1.9	1D propagation, Beji and Battjes (1993) barred flume case.
ww3_tp2.1	2D propagation under angle with grid.
ww3_tp2.2	2D propagation over half the globe without land (with directional spread).
ww3_tp2.3	2D propagation, GSE test.
ww3_tp2.4	2D propagation, East Pacific curvilinear grid test.
ww3_tp2.5	2D propagation, Arctic Grid, curvilinear grid test.
ww3_tp2.6	2D propagation, Limon Harbor unstructured grid test.
ww3_tp2.7	Reflection on a 2D unstructured grid.

ww3_tp2.8	Tidal constituents on a 2D regular grid.
ww3_tp2.9	Tests for obstruction grids.
ww3_tp2.10	Tests for SMC grid.
ww3_tp2.11	Tests for rotated grid.
ww3_tp2.12	Test for system tracking.
ww3_tp2.13	Test for propagation under angle with grid (tripole)
ww3_tp2.14	Test for toy-model using OASIS coupler.
ww3_tp2.15	Test for space-time extremes parameters.
ww3_tp2.16	Two-dimensional propagation on SMC grid with ARC option
ww3_tp2.17	Unstructured grid with Card Deck vs Domain Decompostion for Explicit vs Implicit schemes
ww3_tp2.16	Test for two-dimensional SMC propagation with Arctic pole handling
ww3_ts1	Source term test, time limited growth.
ww3_ts2	Source term test, fetch limited growth.
ww3_ts3	Source term test, hurricane with single moving grid.
ww3_ts4	Source term test, unresolved obstacles.
ww3_tic1.1	Wave-ice interaction, 1D test of S_{ice}.
ww3_tic1.2	Wave-ice interaction, 1D test of "shoaling" effect.
ww3_tic1.3	Wave-ice interaction, 1D test of refraction effect.
ww3_tic1.4	Wave-ice interaction, 1D test with ice floes and ice thickness.
ww3_tic2.1	Wave-ice interaction, 2D test of S_{ice}.
ww3_tic2.2	Wave-ice interaction, 2D test with non-uniform ice.
ww3_tic2.3	Wave-ice interaction, 2D test with uniform ice with increasing thickness.
ww3_tbt1.1	Wave-mud interaction, 1D test of S_{mud}.
ww3_tbt2.1	Wave-mud interaction, 2D test of S_{mud}.
ww3_tpt1.1	Tests for alternative spectral partitioning methods.
ww3_ta1	ww3_uprstrt, update the restart file of homogeneous conditions (1 point model)
mww3_test_01	Test for expanded grid mask with wetting and drying, etc.
mww3_test_02	Two-way nesting test with single inner grid.
mww3_test_03	Overlapping grids and two-way nesting tests (6-grid version with beach in high-resolution grids.)
mww3_test_04	Current or sea-mount test for two-way nesting with

	stationary swell conditions.
mww3_test_05	Three nested hurricane grids with moving grids test.
mww3_test_06	Tests for irregular grid(s) w/ ww3_multi.
mww3_test_07	Tests for unstructured grid(s) w/ ww3_multi.
mww3_test_08	Tests with wind and ice input.

These regression tests are now run using the run_test script in the regtests/bin directory (primary author: Tim Campbell). How to run this script, including options, is shown by running

```
run_test -h
```

The output of running this command is shown here in Fig. 5.2. The test cases are stored in directories under the regtests directory, e.g. regtests/ww3_tp1.1. For example, the contents of /ww3_tp1.1 might be

info	A file containing information about the test case.
input	A permanent directory containing input files for the test case.
work_PR3	A scratch directory for model output (in this example, filename is such because the user had specified "run_test -w work_PR3 ...").

Also provided now is a matrix if regression tests, used by the code developers to assure that new model versions do not break older model versions. The core of this matrix is the file regtests/bin/matrix.base. An example of how to run this is given in regtests/bin/matrix_zeus_HLT, which is Hendrik's driver for the matrix at the NCEP Zeus R&D computer[14]. To run this, make a link to it in the regtests directory and execute after setting the desired option flags in the script. This will make a file matrix in retests, which can then be run interactively or in batch mode as desired. The file can also be manually edited further if so desired. The bin directory under regtests contains the following tools.

cleanup	Cleanup work directories.
comp_switch	Compare switches inside and across test cases. comp_switch -h provides documentation.

[14] Please build your own driver for your own setup using this as a blueprint, rather than editing this file.

matrix.base	Core script to generate matrix of test cases.
matrix.comp	Script to compare output of matrix of test cases between separately checked out model versions.
matrix_zeus_HLT	Example of driver for matrix.base.
run_test	Basic test script as described above.

Note that efficient running of the matrix of regression tests requires a minimization of the need to recompile code between regression tests. This is achieved by the ordering of the regression tests in matrix.base. A way to assure that identical switch files are identified as such is to systematically sort them. This can be done with the script sort_switch in the main bin directory. This script will add default values of missing switches and can also be used to remove or add switches from the file. Run

```
comp_switch -h
```

for documentation of the script.

Finally, the cases directory hold the real-world test cases as described below.

mww3_case_01	Atlantic case with five grids focusing on Trondheim.
mww3_case_02	Pacific case with three grids focusing on Alaska.
mww3_case_03	Original multi-grid case used as global model at NCEP.

Each of these cases is a single script executing the entire model run. Before executing the script, compile the model with the switches indicated in the documentation at the head of the script. Additional data used by these scripts is contained in the directories

mww3_data_00	Wind fields and ice data used by all example cases.
mww3_data_*nn*	Specific data needed for script mww3_case_*nn*.

These examples can be used as blueprints for setting up other real model applications.

```
Usage: run_test [options] source_dir test_name
Required:
  source_dir : path to top-level of WW3 source
  test_name  : name of test case (directory)
Options:
  -a ww3_env       : use WW3 environment setup file <ww3_env>
                   :   *default is <source_dir>/bin/wwatch3.env
                   :   *file will be created if it does not already exist
  -c cmplr         : setup comp & link files for specified cmplr
  -C coupl         : invoke test using <coupl> coupled application
                   :   OASIS : OASIS3-mct ww3_shel coupled application
                   :   ESMF  : ESMF ww3_multi coupled application
  -d               : invoke main program using gdb (non-parallel)
  -e               : prompt for changes to existing WW3 environment
  -f               : force pre- and post-processing programs to be compiled
                   : non-MPI (i.e., with SHRD switch); default is all programs
                   : compiled with unmodified switch settings
  -g grid_string   : use ww3_grid_<grid_string>.inp
  -G               : create GrADS data files using gx_outX.inp
  -h               : print usage and exit
  -i inpdir        : use inputs in test_name/<inpdir> (default  test_name/input)
  -m grid_set      : execute multi-model test
                   :   *grid names are obtained from input/<grid_set>
                   :   *ww3_multi_<grid_set> will execute instead of ww3_shel
                   :   *to execute a single model test case with ww3_multi use
                   :    grid_set = none
  -n nproc         : specify <nproc> processors for parallel run
                   :   *some <runcmd> programs do not require <nproc>
                   :   *ignored if -p <runcmd> or -O is not specified
  -N               : use namelist (.nml) input instead of .inp (if available)
  -o outopt        : limit output post-processing based on <outopt>
                   :   native : post-process only native output
                   :   netcdf : post-process only NetCDF output
                   :   both   : post-process both native and NetCDF output
                   :   * default is native
                   :   * note that required input files must be present for
                   :     selected output post-processing to occur
  -O               : parallel run using OpenMP paradigm and OMP_NUM_THREADS
                     environment variable and number of processors defined with
                     the -n np option
  -p runcmd        : run in parallel using <runcmd> to start program
                   :   *MPICH or OpenMPI: mpirun or mpiexec (default <nproc> = 1)
                   :   *IBM with Loadleveler: poe (no <nproc> required)
                   :   *LSF: mpirun.lsf (no <nproc> required)
  -q program       : exit script after program <program> executes
  -r program       : only execute program <program>
  -s switch_string : use switch_<switch_string>
  -S               : create stub file <finished>. with end data and time.
                     tests not executed if file is found.
  -t nthrd         : Threading option. (this is system dependant and can be used
                   : only for the hybrid option)
  -w work_dir      : run test case in test_name/work_dir (default test_name/work)
```

Figure 5.2: Options for run_test, as obtained by running it with the -h command line option.

6 System documentation

6.1 Introduction

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

6.2 The preprocessor

The WAVEWATCH III source code files are not ready to use FORTRAN files; mandatory and optional program options still have to be selected, and test output may be activated[15]. Compile level options are activated using 'switches'. The arbitrary switch 'SWT' is included in the WAVEWATCH III files as comment of the form !/SWT, where the switch name SWT is followed by a space or by a '/'. If a switch is selected, the preprocessor removes the comment characters, thus activating the corresponding source code line. If '/' follows the switch, it is also removed, thus allowing the selective inclusion of hardware-dependent compiler directives etc. The switches are case sensitive, and available switches are presented in Section 5.9. Files which contain the switch C/SWT can be found by typing

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

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

```
all_switches
```

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

```
0 1
constants.ftn'  constants.f'
'F90 NOGRB SHRD PR3 UQ FLX2 LN1 ST2 STAB2
   NL1 BT1 DB1 MLIM TR0 BS0 XX0 WNX1 WNT1 CRX1 CRT1
   O0 O1 O2 O3 O4 O5 O6 O7 O11 O14'
```

Figure 6.1: Example input for W3ADC.

Pre-processing is performed by the program `w3adc`. This program is found in the file w3adc.f, which contains a ready to compile FORTRAN source code and a full documentation[16]. 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 Fig. 6.1. Line-by-line, the input consists of

- → Test indicator and compress indicator.
- → File names of the input and output code.
- → Switches to be turned on in a single string (see Section 5.9).
- → Additional lines with include files can be given, but these are no longer used in the automated compile system.

A test indicator 0 disables test output, and increasing values increase the detail of the test output. A compress indicator 0 leaves the file as is. A compress indicator 1 results in the removal of all comment lines indicated by '!', except for empty switches, i.e., lines starting with '!/'. A compress indicator 2 results in the subsequent removal of all comments. Comment lines are not allowed in this input file. The above input for W3ADC is read using free format. Therefore quotes are needed around strings. Echo and test output is send to the standard output device. To facilitate the use of the preprocessor, several UNIX scripts are provided with WAVEWATCH III

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

as discussed in Section 5.7. Note that compiler directives are protected from file compression by defining them using a switch.

6.3 Program files

The WAVEWATCH III source code files are stored in files with the extension ftn[17]. Starting with version 2.00, the code has been organized in modules. Only the main programs are not packaged in modules. Originally, variables were bundled with the code modules, resulting in a single static data structure. In model version 3.06, a separate dynamical data structure was introduced, allow for the presence of multiple wave grids in a single program, as a preparation for the development of the the multi-grid model driver.

The subroutines contained in the modules are described in some detail below. The relation between the various subroutines is graphically depicted in Figs. 6.2 and 6.3. Three groups of codes are considered. The first are the main wave model subroutine modules, which are generally identified by the file name structure w3*xxxx*md.ftn. These modules are described in Section 6.3.1. The second group consists of modules specific to the multi-grid wave model driver, which are generally identified by the file name structure wm*xxxx*md.ftn. These modules are described in Section 6.3.2. The final group consists of auxiliary programs and wave model drivers, and is described in Section 6.3.4. Section 6.3.3 briefly describes the data assimilation module.

6.3.1 Wave model modules

At the core of the wave model are the wave model initialization module and the wave model module.

Main wave model initialization module — w3initmd.ftn

w3init	The initialization routine W3INIT, which prepares the wave model for computations (internal).
w3mpii	MPI initialization (internal).

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

w3mpio	MPI initialization for I/O (internal).
w3mpip	MPI initialization for I/O (internal, point output only).

Main wave model module w3wavemd.ftn

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

The main wave model routines and all other subroutines require a data structure to exist. The data structure is contained in the following modules.

Define model grids and parameter settings w3gdatmd.ftn

w3nmod	Set number of grids to be considered.
w3dimx	Set dimensions for spatial grid and allocate storage.
w3dims	Set dimensions for spectral grid and allocate storage.
w3setg	Set pointers to selected grid.
w3dimug	Set dimensions for arrays specific to the triangle-based grids (grid connectivity ...).
w3gntx	Develop unstructured grid structures.

Dynamic wave data describing sea state w3wdatmd.ftn

w3ndat	Set number of grids to be considered.
w3dimw	Set dimensions and allocate storage.
w3setw	Set pointers to selected grid.

Auxiliary storage w3adatmd.ftn

w3naux	Set number of grids to be considered.
w3dima, w3xdma, w3dmnl	Set dimensions and allocate storage.
w3seta, w3xeta	Set pointers to selected grid.

Model output w3odatmd.ftn

w3nout	Set number of grids to be considered.
w3dmo2, w3dmo3, w3dmo5	Set dimensions and allocate storage.
w3seto	Set pointers to selected grid.

Model input w3idatmd.ftn

w3ninp	Set number of grids to be considered.
w3dimi	Set dimensions and allocate storage.
w3seti	Set pointers to selected grid.

The configuration files such as traditional form (.inp) and namelist form (.nml) are processed differently in the program. The traditional configuration file is directly read in the main program. The namelist configuration file is read by the subroutines in the following modules.

Grid preprocessor namelist module w3nmlgridmd.ftn

w3nmlgrid	Read and report all the namelists.
read_spectrum_nml	Init and save values to a derived type.
read_run_nml	Init and save values to a derived type.
read_timesteps_nml	Init and save values to a derived type.
read_grid_nml	Init and save values to a derived type.
read_rect_nml	Init and save values to a derived type.
read_curv_nml	Init and save values to a derived type.
read_unst_nml	Init and save values to a derived type.
read_smc_nml	Init and save values to a derived type.
read_mask_nml	Init and save values to a derived type.
read_obst_nml	Init and save values to a derived type.
read_slope_nml	Init and save values to a derived type.
read_sed_nml	Init and save values to a derived type.
read_inbound_nml	Init and save values to a derived type.
read_excluded_nml	Init and save values to a derived type.
read_outbound_nml	Init and save values to a derived type.
report_spectrum_nml	Output default and user-defined values.
report_run_nml	Output default and user-defined values.
report_timesteps_nml	Output default and user-defined values.

report_grid_nml	Output default and user-defined values.
report_rect_nml	Output default and user-defined values.
report_curv_nml	Output default and user-defined values.
report_smc_nml	Output default and user-defined values.
report_depth_nml	Output default and user-defined values.
report_mask_nml	Output default and user-defined values.
report_obst_nml	Output default and user-defined values.
report_slope_nml	Output default and user-defined values.
report_sed_nml	Output default and user-defined values.
report_inbound_nml	Output default and user-defined values.
report_excluded_nml	Output default and user-defined values.
report_outbound_nml	Output default and user-defined values.

NetCDF boundary conditions namelist module w3nmlbouncmd.ftn

w3nmlbounc	Read and report all the namelists.
read_bound_nml	Init and save values to a derived type.
report_bound_nml	Output default and user-defined values.

NetCDF input field preprocessor namelist module w3nmlprncmd.ftn

w3nmlprnc	Read and report all the namelists.
read_forcing_nml	Init and save values to a derived type.
read_file_nml	Init and save values to a derived type.
report_forcing_nml	Output default and user-defined values.
report_file_nml	Output default and user-defined values.

Generic shell namelist module w3nmlshelmd.ftn

w3nmlshel	Read and report all the namelists.
read_domain_nml	Init and save values to a derived type.
read_input_nml	Init and save values to a derived type.
read_output_type_nml	Init and save values to a derived type.
read_output_date_nml	Init and save values to a derived type.
read_homogeneous_nml	Init and save values to a derived type.
report_domain_nml	Output default and user-defined values.

report_input_nml	Output default and user-defined values.
report_output_type_nml	Output default and user-defined values.
report_output_date_nml	Output default and user-defined values.
report_homogeneous_nml	Output default and user-defined values.

Multi-grid shell namelist module w3nmlmultimd.ftn

w3nmlmultidef	Set the number of model and forcing grids.
w3nmlmulticonf	Read and report all the namelists.
read_domain_nml	Init and save values to a derived type.
read_input_grid_nml	Init and save values to a derived type.
read_model_grid_nml	Init and save values to a derived type.
read_output_type_nml	Init and save values to a derived type.
read_output_date_nml	Init and save values to a derived type.
read_homogeneous_nml	Init and save values to a derived type.
report_domain_nml	Output default and user-defined values.
report_input_grid_nml	Output default and user-defined values.
report_model_grid_nml	Output default and user-defined values.
report_output_type_nml	Output default and user-defined values.
report_output_date_nml	Output default and user-defined values.
report_homogeneous_nml	Output default and user-defined values.

Gridded output NetCDF post-processor namelist module w3nmlounfmd.ftn

w3nmlounf	Read and report all the namelists.
read_field_nml	Init and save values to a derived type.
read_file_nml	Init and save values to a derived type.
read_smc_nml	Init and save values to a derived type.
report_field_nml	Output default and user-defined values.
report_file_nml	Output default and user-defined values.
report_smc_nml	Output default and user-defined values.

Point output NetCDF post-processor namelist module w3nmlounpmd.ftn

w3nmlounp	Read and report all the namelists.
read_point_nml	Init and save values to a derived type.

read_file_nml	Init and save values to a derived type.
read_spectra_nml	Init and save values to a derived type.
read_param_nml	Init and save values to a derived type.
read_source_nml	Init and save values to a derived type.
report_point_nml	Output default and user-defined values.
report_file_nml	Output default and user-defined values.
report_spectra_nml	Output default and user-defined values.
report_param_nml	Output default and user-defined values.
report_source_nml	Output default and user-defined values.

Track output NetCDF post-processor namelist module w3nmltrncmd.ftn

w3nmltrnc	Read and report all the namelists.
read_track_nml	Init and save values to a derived type.
read_file_nml	Init and save values to a derived type.
report_track_nml	Output default and user-defined values.
report_file_nml	Output default and user-defined values.

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

Input update module w3updtmd.ftn

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

There are seven types of WAVEWATCH III data files (other than the pre-processed input fields, which are part of the program shall rather than the actual wave model). The corresponding routines are gathered in six modules.

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

w3iogr	Reading and writing of mod_def.ww3.

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

w3outg	Calculation of gridded output parameters.
w3iogo	Reading and writing of out_grd.ww3.

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

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

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

w3iotr	Generate track output in track_o.ww3.

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

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

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

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

I/O module (partition.ww3) w3iofsmd.ftn

w3iofs	Writing of partition.ww3.

There are presently several propagation schemes and GSE alleviation techniques available for rectangular and curvilinear grids, as well as a 'slot' for a user supplied propagation routine, and there are four schemes for triangle-based grids. The propagation schemes are packaged in the following modules.

Propagation module (first order, no GSE alleviation) w3pro1md.ftn

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

Propagation module (higher order scheme with GSE diffusion) w3pro2md.ftn

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

Propagation module (higher order scheme with GSE averaging) w3pro3md.ftn

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

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

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

Propagation module (generic UQ) w3uqckmd.ftn

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

Propagation module (generic UNO) w3uqckmd.ftn

w3uno, w3unor w3unos	Like UQ schemes above.

SMC grid routines w3psmcmd.ftn

W3PSMC	Spatial propagation on SMC grid.
W3KSMC	Spectral modification by GCT and refraction.

SMCxUNO2/3	Irregular grid mid-flux on U-faces by UNO2/3.
SMCyUNO2/3	Irregular grid mid-flux on V-faces by UNO2/3.
SMCxUNO2r/3r	Regular grid mid-flux on U-faces by UNO2/3.
SMCyUNO2r/3r	Regular grid mid-flux on V-faces by UNO2/3.
SMCkUNO2	Shift in k-space due to refraction by UNO2.
SMCGradn	Evaluate field gradient at cell centre.
SMCAverg	1-2-1 weighted average for centre field.
SMCGtCrfr	Refraction and GCT rotation in theta.
SMCDHXY	Evaluate depth gradient and refraction limiter.
SMCDCXY	Evaluate current velocity gradient.
W3GATHSMC W3SCATSMC	Gather and scatter spectral components.

Triangle-based propagation schemes w3profsmd.ftn

w3xypug	Interface to the unstructured propagation schemes.
w3cflug	Computes the maximum CFL number for spatial propagation.
w3xypfsn2	N-scheme.
w3xypfspsi2	PSI-scheme.
w3xypfsnimp	Implicit version of the N-scheme.
w3xypfsfct2	FCT-scheme.
bcgstab	Part of the iterative SPARSKIT solver, used for the implicit scheme.

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

Source term integration module w3srcemd.ftn

w3srce	Integration of source terms.

Flux (stress) module (Wu, 1980)		w3flx1md.ftn
w3flx1	Calculation of stresses.	
Flux (stress) module (Tolman and Chalikov)		w3flx2md.ftn
w3flx2	Calculation of stresses.	
Flux (stress) module (Tolman and Chalikov, capped)		w3flx3md.ftn
w3flx3	Calculation of stresses.	
Flux (stress) module (slot for user supplied routines)		w3flxxmd.ftn
w3flxx	Calculation of stresses.	
inflxx	Initialization routine.	
Linear input (Cavaleri and Malanotte Rizzoli)		w3sln1md.ftn
w3sln1	Calculation S_{lin}.	

Linear input (slot for user supplied routines) w3slnxmd.ftn

w3slnx	Calculation S_{lin}.
inslnx	Corresponding initialization routine.

Input and dissipation module (dummy version) w3src0md.ftn

w3spr0	Calculation of mean wave parameters (single grid point).

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

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

Input and dissipation module Tolman and Chalikov 1996 w3src2md.ftn

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

Input and dissipation module WAM-4 and ECWAM. w3src3md.ftn

w3spr3	Calculation of mean wave parameters (single grid point).
w3sin3	Calculation of S_{in}.
w3sds3	Calculation of S_{ds}.
tabu_stress	Tabulation of wind stress as a function of U_{10} and τ_w
tabu_tauhf	Tabulation of the short waves-supported stress
calc_ustar	Computes friction velocity using stress table

Input and dissipation module Ardhuin et al. 2010 w3src4md.ftn

w3spr4	Calculation of mean wave parameters (single grid point).
w3sin4	Calculation of S_{in}.
w3sds4	Calculation of S_{ds}.
tabu_stress	Tabulation of wind stress as a function of U_{10} and τ_w
tabu_tauhf	Tabulation of the short waves-supported stress
tabu_tauhf2	Tabulation of the short waves-supported stress with sheltering
tabu_swellft	Tabulation of oscillatory friction factor for negative part of S_{in}.
calc_ustar	Computes friction velocity using stress table

Input and dissipation module BYDRZ w3src6md.ftn

w3spr6	Integral parameter calculation following ST1.
w3sin6	Observation-based wind input.
w3sds6	Observation-based dissipation.
irange	Generate a sequence of integer values.
lfactor	Calculate reduction factor for Sin.
tauwinds	Normal stress calculation for Sin.
polyfit2	Quadratic fit using least-squares.

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

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

Swell dissipation module w3swldmd.ftn

w3swl4	Ardhuin et al (2010+) swell dissipation.
w3swl6	Babanin (2011) swell dissipation.
irange	Generate a sequence of integer values.

Nonlinear interaction module (DIA) w3snl1md.ftn

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

Nonlinear interaction module (WRT) w3snl2md.ftn

w3snl2	Interface routine for S_{nl}.
insnl2	Initialization for S_{nl}.

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

Nonlinear interaction module (GMD) w3snl3md.ftn

w3snl3	Calculation of S_{nl}.
expand	Expand spectral space.
expan2	Map form expanded to original spectral space.
insnl3	Initialization for S_{nl}.

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

w3snlx	Calculation of S_{nl}.
insnlx	Initialization for S_{nl}.

Nonlinear high-frequency filter w3snlsmd.ftn

w3snls	Calculation of filter.
expand	Expand spectral space.
insnls	Initialization for filter.

Bottom friction module (JONSWAP) w3sbt1md.ftn

w3bt1	Calculation of S_{bot}.

Bottom friction module (SHOWEX) w3sbt4md.ftn

insbt4	Initialization of S_{bot})

tabu_erf	Table or error function.
w3sbt4	Calculation of S_{bot}, and energy and momentum fluxes to the bottom boundary layer.

Fluid mud dissipation (Dalrymple and Liu, 1978) w3sbt8md.ftn

w3sbt8	Source term.

Fluid mud dissipation (Ng, 2000) w3sbt9md.ftn

w3sbt9	Source term.

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

w3sbtx	Calculation of S_{bot}.
insbtx	Initialization of S_{bot}.

Depth induced breaking module (Battjes-Janssen) w3sdb1md.ftn

w3sdb1	Calculation of S_{db}.

Depth induced breaking module (slot for user supplied routines) w3sdbxmd.ftn

w3sdbx	Calculation of S_{db}.
insdbx	Initialization of S_{db}.

Triad interactions module (LTA) w3str1md.ftn

w3str1	Calculation of S_{tr}.

Triad interactions module (slot for user supplied routines) w3strxmd.ftn

w3strx	Calculation of S_{tr}.
instrx	Initialization of S_{tr}.

Bottom scattering module w3sbs1md.ftn

w3sbs1	Calculation of S_{bs} and associated momentum flux to the bottom.
insbs1	Initialization of S_{bs}.

Bottom scattering module (slot for user supplied routines) w3sbsxmd.ftn

w3sbsx	Calculation of S_{bs}.
insbsx	Initialization of S_{bs}.

Wave-ice interactions (simple) w3sic1md.ftn

w3sic1	Calculation of S_{id}.

Wave-ice interactions (Liu et al.) w3sic2md.ftn

w3sic2	Calculation of S_{id}.
	Interpolation tables.

Wave-ice interactions Wang and Shen (2010) w3sic3md.ftn

w3sic3	Calculation of S_{id}.
bsdet	Calculate the determinant for the dispersion relation.
wn_complex	Calculate complex wavenumber in ice.
cmplx_root_muller	Find root for complex numbers.
fun_zhao	Wrapper for functions below.
func0_zhao, finc1_zhao	

w3sis2	Calculation of S_{is}.

Waves scattering in ice and ice break-up w3sis2md.ftn
Shoreline reflection w3ref1md.ftn

w3ref1	Calculation of S_{ref}.

Module for unclassified source term (slot for user supplied routines) w3sxxxmd.ftn

w3sxxx	Calculation of S_{xx}.
insxxx	Initialization of S_{xx}.

To complete the basic wave model, several additional modules are needed. For the actual contents of the service modules see the documentation in the source code files.

constants.ftn	Physical and mathematical constants and Kelvin functions.
w3arrymd.ftn	Array manipulation routines including 'print plot' routines.

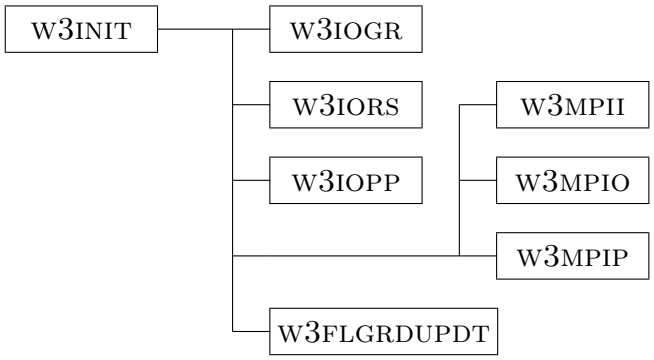

Figure 6.2: Subroutine structure for wave model initialization routine without service routines, data base management routines and MPI calls. Note that W3IOGR on reading data in calls all necessary initialization routines for interpolation tables and physics parameterizations.

---

w3bullmd.ftn	Perform bulletin style output for output points.
w3cspcmd.ftn	Conversion of spectral discretization.
w3dispmd.ftn	Routines to solve the Laplace dispersion relation (linear waves, flat bottom, no ice), including interpolation tables. Includes also ice corrections in liu_foreward_dispersion and liu_inverse_dispersion.
w3gsrumd.ftn	Regridding utilities.
w3partmd.ftn	Perform spectral partitioning for a single spectrum.
w3servmd.ftn	General service routines.
w3timemd.ftn	Time management routines.
w3triamd.ftn	Basic routines for triangle-based grids: reading, interpolation, definition of miscellaneous arrays, determination of boundary points.

This completes the description of the basic wave model routines. The relation between the initialization routine and other routines is illustrated in Fig. 6.2. A similar relational diagram for the wave model routine is presented in Fig. 6.3.

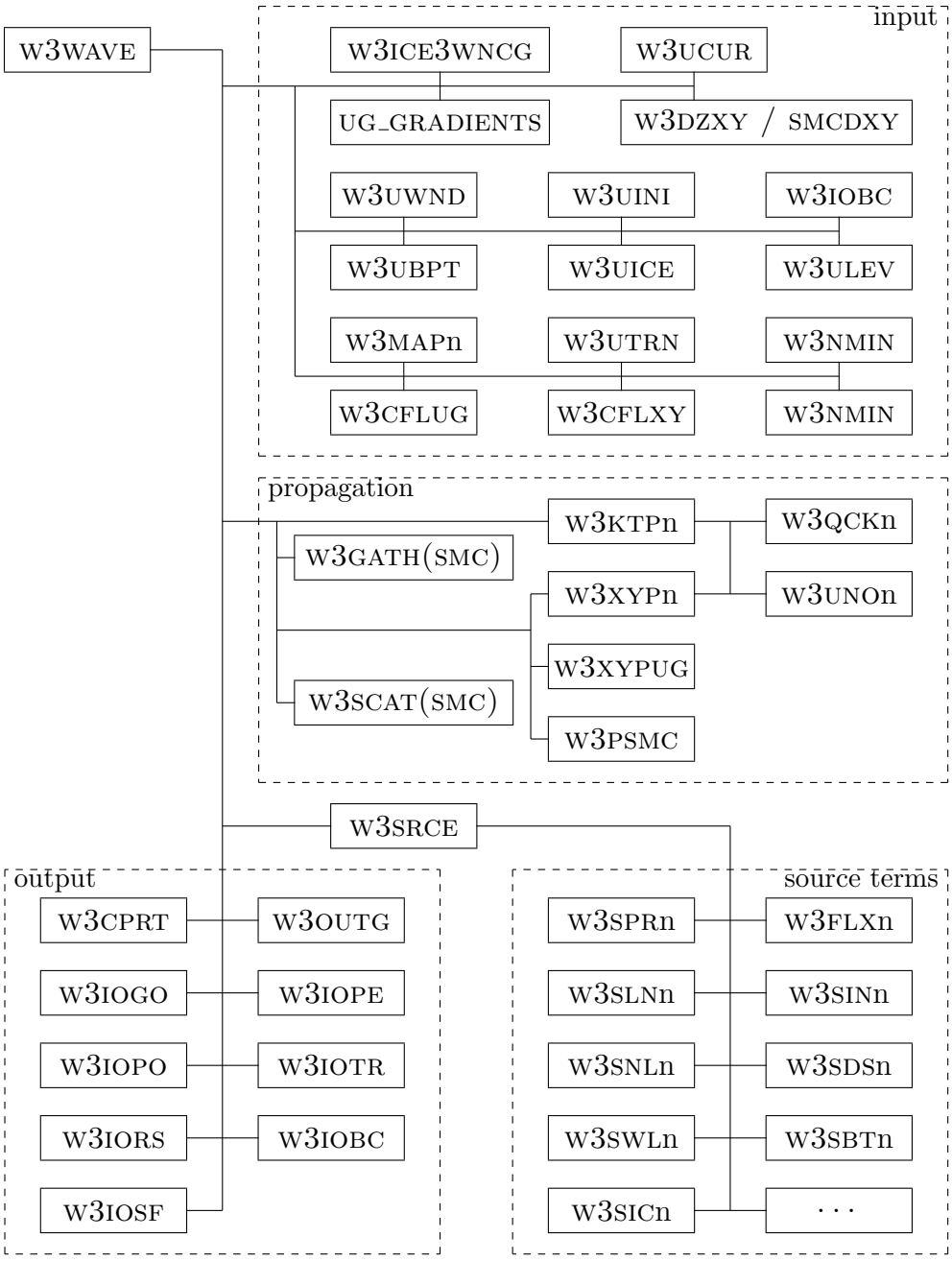

Figure 6.3: Subroutine structure for wave model routine without service routines, routines managing the data structures, and MPI routines. '. . . ' identifies additional source term routines.

6.3.2 Multi-grid modules

The multi-grid wave model shel ww3_multi provides a shell around the basic wave model as described in the previous section. This shell manages the side-by-side running of multiple wave model grids, and all communication between the grids. To achieve this various additional modules have been developed. At the core are the initialization, multi-grid model and finalization routines.

Initialization of multi-grid model wminitmd.ftn

wminit	Multi-grid model initialization.

Running of multi-grid model wmwavemd.ftn

wmwave	Multi-grid model execution.
wmprnt	Printing to log file.
wmbcst	Non-blocking MPI broadcast.
wmwout	Idem.

Finalizing of multi-grid model wmfinlmd.ftn

wmfinl	Multi-grid model finalization.

These routines are designed to become part of a coupled model. For the structure of the actual wave model routine, reference is made to Tolman (2007). The resulting wave model driver ww3_multi consequently becomes extremely simple; it initializes the MPI environment, and then calls the above three modules consecutively.

The main multi-grid wave model routines require an expansion of the data structure used by WAVEWATCH III. Furthermore, main activities are gathered in subroutines in various modules.

Data storage wmmdatmd.ftn

wmndat	Set number of grids to be considered.
wmdimd, wmdimm	Set dimensions and allocate storage.
wmsetm	Set pointers to selected grid.

Determine grid relations — wmgridmd.ftn

wmglow	Relations to lower ranked grids.
wmghgh	Relations to higher ranked grids.
wmgeql	Relations between equal ranked grids.
wmrspc	Determine need for spectral conversion between grids.

Update model input — wmupdtmd.ftn

wmupdt	General input update routine.
wmupd1	Update input from native files using w3fldsmd.ftn from Section 6.3.4.
wmupd2	Update input from pore-defined input grids.
wmupdv	Update vector fields.
wmupds	Update scalar fields.

Perform internal communications — wminiomd.ftn

wmiobs	Stage internal boundary data.
wmiobg	Gather internal boundary data.
wmiobf	Finalize WMIOBS (MPI only).
wmiohs	Stage internal high to low rank data.
wmiohg	Gather internal high to low rank data.
wmiohf	Finalize WMIOHS (MPI only).
wmioes	Stage internal data between equal ranked grids.
wmioeg	Gather internal data between equal ranked grids.
wmioef	Finalize WMIOES (MPI only).

Unify point output to single file — wmiopomd.ftn

wmiopp	Initialization routine.
wmiopo	Data gather and write routine (using W3IOPO in w3iopomd.ftn).

To complete the multi-grid wave model, one additional service module is needed. For the actual contents of the service module see the documentation in the source code files.

wmunitmd.ftn	Dynamic unit number assignment
wmscrpmd.ftn	SCRIP utilities.

6.3.3 Data assimilation module

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

Data assimilation module w3wdasmd.ftn

w3wdas	Data assimilation interface.

6.3.4 Auxiliary programs

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

Input data file management module w3fldsmd.ftn

w3fldo	Opening and checking of data files for W3SHEL.
w3fldg	Reading and writing of data files for W3SHEL (model input).

w3fldd	Reading and writing of data files for W3SHEL (data assimilation).
w3fldp	Prepare interpolation of input fields from arbitrary grids.
w3fldh	Management of homogeneous input fields in W3SHEL.
w3fldm	Process moving grid data in W3SHEL.

Grid pre-processing program ww3_grid.ftn

w3grid	The grid preprocessor.
readnl	Reading NAMELIST input (internal).

Initial conditions program ww3_strt.ftn

w3strt	The initial conditions program.

Boundary conditions program ww3_bound.ftn
Boundary conditions program (NetCDF) ww3_bounc.ftn

w3bound	The boundary conditions program (NetCDF).

Input field pre-processing program ww3_prep.ftn
Input field pre-processing program from NetCDF files ww3_prnc.ftn

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

Tide pre-processing program ww3_prtide.ftn

w3prtide	Pre-processor for tides.

Restart file pre-processing program ww3_uprstr.ftn

w3uprstr	The restart file preprocessor.
readnl	Reading NAMELIST input (internal).

Generic wave model program ww3_shel.ftn

w3shel	The generic program shell.

Grid splitting for ww3_multi — ww3_gspl.ftn

w3gspl	The grid splitting program.
grinfo, grtrim grfill, grlost, grsqrg, grsngl, grsepa, grfsml, grfrlg, gr1grd	Routines to incrementally adjust individual grids.

Generic wave model program — ww3_multi.ftn

w3mlti	The multi-grid program shell.

Grid output integration for ww3_multi — ww3_gint.ftn

w3gint	The post-processing program for integrating gridded fields of mean wave parameters.
w3exgi	Actual output routine (internal).

Gridded data post-processing program — ww3_outf.ftn

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

Gridded data post-processing program (NetCDF) ww3_ounf.ftn

w3ounf	The post-processing program for gridded fields of mean wave parameters, using NetCDF3 or NetCDF4 libraries for Fortran90.
w3crnc	Creation of NetCDF files, definition of dimensions and header data.
w3exnc	Actual output routine (internal).

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

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

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

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

Point post-processing program ww3_outp.ftn

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

Point post-processing program ww3_ounp.ftn

w3ounp	The post-processing program output at selected locations using NetCDF.
w3crnc	Creation of NetCDF files, definition of dimensions and header data.
w3exnc	Actual output routine (internal).

Point post-processing program (GrADS) gx_outp.ftn

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

Track output post-processing program ww3_trck.ftn

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

Track output post-processing program ww3_trnc.ftn

w3trck	Converting unformatted direct access track output file to NetCDF file.
w3crnc	Creation of NetCDF files, definition of dimensions and header data.
w3exnc	Actual output routine (internal).

Wave field tracking post-processing program ww3_systrk.ftn

w3systrk	Tracking wave fields in space and time.

6.4 Optimization

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

Optimization for vector computers has been performed by structuring the code in long vector loops where possible. Optimization was originally performed for the Cray YMP and C90. Note that some compiler directives for vectorization have been used. Note also that the vector optimization has not been updated since about 1997, and therefore needs to be revisited if the model is implemented on a vector machine. Vectorization directives are activated by the corresponding preprocessor switch (C90).

Parallelization for shared memory machines using threading has been implemented using standard OpenMP directives. Such parallelization takes

place mainly in the loop calling the source term routine W3SRCE and the different propagation routines. OpenMP directives are activated by the corresponding preprocessor switches (OMPn).

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

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

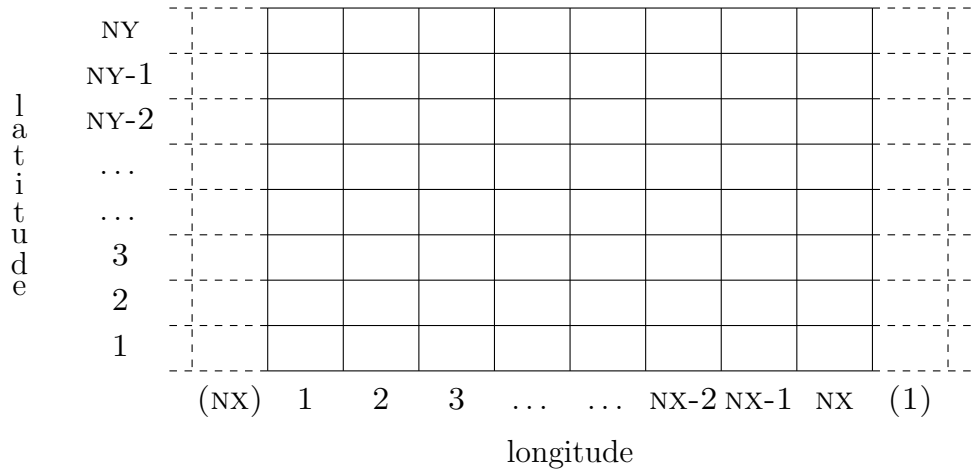

Figure 6.4: Layout of the spatial grid. Grid points are denoted as boxes, dotted boxes denoted repeated columns for global model applications.

6.5 Internal data storage

The remainder of this chapter will deal with the internal data storage used by WAVEWATCH III. In Section 6.5.1 the layout of a single wave model grid as used in ww3_shel is discussed. In Section 6.5.2 the parallelization approaches for a single grid are discussed. In Section 6.5.3 the simultaneous storage of multiple wave grids is discussed. Finally, the actual wave model variables are described in Section 6.6. Note that the code is fully documented, including the variables defining the 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. 6.4. The grid can either be a Cartesian '(x, y)' grid, a spherical grid (with regular steps on latitude and longitude), a curvilinear grid, or a triangle-based grid. In a spherical grid, the longitudes

are denoted throughout the program by the counter IX, and latitudes by the counter IY, and the corresponding grid dimensions (NX,NY). All spatial field arrays are dynamically allocated within the code, corresponding work arrays are usually automatic, to allow for thread-safe code. The closure of the grid in case of a global applications is handled within the model, and does not require user intervention. To simplify the calculation of derivatives of in particular the current, the outer grid points (IX=1,NX, unless the grid is global) and (IY=1,NY) will be considered as land points, inactive points or active boundary points. The minimum grid size therefore is NX=3, NY=3, except for triangle-based grids. In that latter case, all the nodes are listed as a long vector of dimension NX, while NY=1, allowing to keep the same code structure. Input arrays are typically assumed to be of the form

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

and are read row by row (see also chapter 4). Within the program, however, they are typically stored with rotated indices

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

This makes it easier to provide global closure, which typically requires extension of the x axis. Furthermore, such two-dimensional array are usually treated as one-dimensional arrays, to increase vector lengths. The array ARRAY, its one-dimensional equivalent VARRAY and IXY are defined as

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

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

The spectral grid for a given spatial grid point (IX,IY) is defined similarly, using a directional counter ITH and a wavenumber counter IK (Fig. 6.5). The size of the spectral grid is set using dynamic allocation. As with the spatial grid, the internal description of the spectrum A is defined as

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

and equivalent one-dimensional arrays are used throughout the program. In-

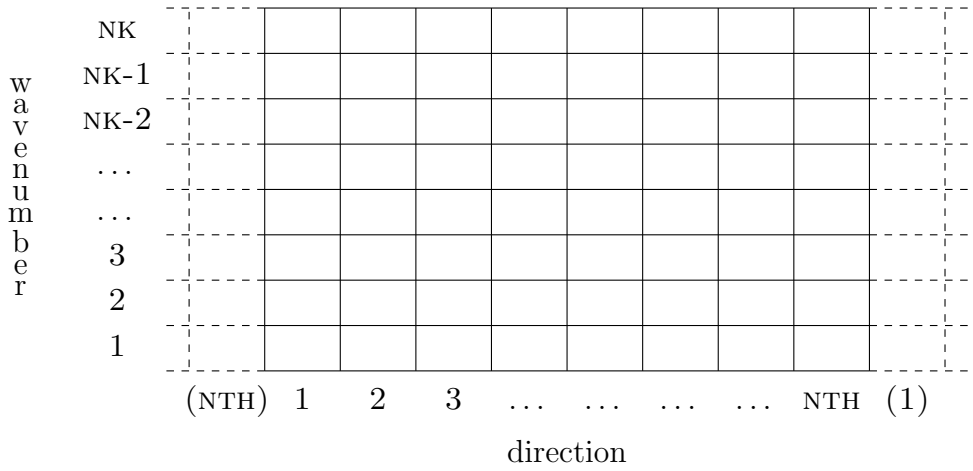

Figure 6.5: Layout of the spectral grid. Dotted boxes denoted repeated columns for directional closure.

---

side the model, directions are always Cartesian, $\theta = 0°$ corresponds to propagation from west to east (positive x or IX direction), and $\theta = 90°$ corresponds to propagation from south to north (positive y or IY direction). Output directions use other conventions, as is discussed in Chapter 4.

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

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

An example of the layout of this storage grid in relation to the full grid of Fig. 6.4 is given in Fig. 6.6. 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.

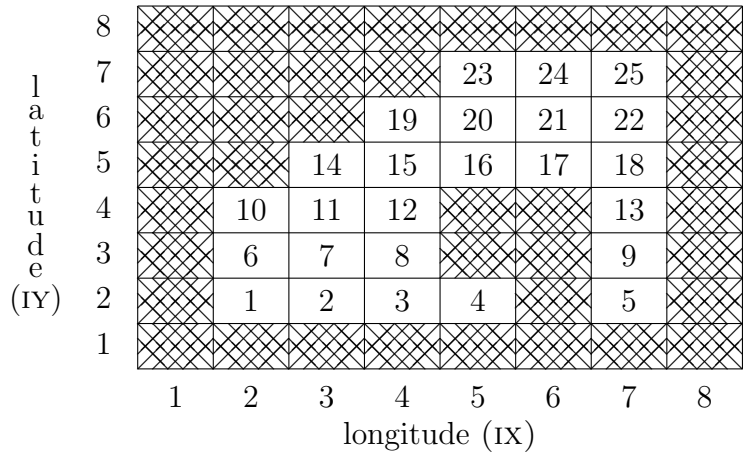

Figure 6.6: An example of the one-dimensional storage grid for spectra. Hatched grid boxes denote land points. Numbers within the grid boxes show the grid counter ISEA of the storage grid.

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

where MAPFS(IY,IX) = 0 for land points. Finally, status maps MAPSTA(IY,IX) and MAPST2(IY,IX) are maintained to identify sea, land, active boundary and ice points. MAPSTA represents the main status map for the grid;

MAPSTA(IY,IX) = 0	for excluded 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). MAPST2 contains secondary information. For excluded points MAPSTA(IY,IX) = 0, this map distinguished between land

points MAPST2(IY,IX) = 0 and otherwise excluded points MAPST2(IY,IX) = 1. For sea points that are disabled MAPSTA(IY,IX) < 0, consecutive bits in MAPST2 identify the reason for deactivation (bit value 1 indicating deactivation).

bit	identifies
1	Ice coverage
2	Point dried out
3	Land in moving grid or inferred in nesting
4	Masked in two-way nesting

Two additional considerations have been made. First, the two status maps can be collapsed into a single map for storage. To assure that the storage is backward compatible with the previous mode version, the two maps are combined into a single map MAPTMP

$$\text{MAPTMP} = \text{MAPSTA} + 8 * \text{MAPST2}$$

considering that only the first few bits of MAPSTA contain data. It is this map MAPTMP that is saved in NetCDF files. The original maps can be recovered as

$$\text{MAPSTA} = \text{MOD} (\text{MAPTMP} + 2 , 8) - 2$$
$$\text{MAPST2} = \text{MAPTMP} - \text{MAPSTA}$$

Second, a single map is used in the graphics output program, to simplify the plotting of the status of grid points. In the graphics files, the map is defined as

map	implies
2	Active boundary point
1	Active sea point
0	Land point (including as identified in MAPST2
-1	Point covered by ice, but wet
-2	Dry point, not covered by ice
-3	Dry point covered by ice
-4	Point masked in the two-way nesting scheme
-5	Other disabled point

Similarly, a single map can be used to simplify processing in the grid preparation program ww3_grid. In this map a distinction is made between points as follows:

map	implies
3	Excluded points
2	Active boundary point
1	Active sea point
0	Land point

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 used in the computation is NAPROC, and if IAPROC is the processor number ranging form 1 to NAPROC, these parameters are related in the following way

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

In model version 3.10, a further refinement was introduced. The actual number of processors NAPROC can be smaller than the total number of processors used by the program (NTPROC). Processors where NAPROC $<$ IAPROC $\leq$ NTPROC are reserved for output processing only.

With this data distribution, source terms and intra-spectral propagation can be calculated at the each given processor without the need for communication between processors. For spatial propagation, however, a data transpose is required where the spectral components (ITH,IK) for all spatial grid points have to be gathered at a single processor. After propagation has

been performed, the modified data have to be scattered back to their 'home' processor. Individual spectral components are assigned to specific processors in such a way that the number of partial propagation steps to be performed by each processor is roughly identical. This makes a good load balance possible. The actual algorithm can be found in section 4.d of the subroutine W3INIT (w3initmd.ftn).

The data transpose for the gather operation is implemented in two steps using the Message Passing Interface (MPI) standard (e.g. Gropp et al., 1997). First, values for each spatial grid point for a given spectral bin (ITH,IK) are gathered in a single target processor in a one-dimensional array STORE(ISEA), which then is converted to the full two-dimensional field of spectral components. After propagation has been performed, the transpose for the scatter operation reverses this process, using the same one-dimensional array STORE. Whereas the algorithm for distributing spatial propagation over individual processors assures a global (per time step) load balance, it does not assure that communication is synchronized, because not each calculation at each processor will take the same effort. To avoid that this results in a load imbalance, non-blocking communication has been used. Furthermore, the one-dimensional array STORE(ISEA) is replaced by STORE(ISEA,IBUF), where the added dimension of the array supplies an actively managed buffer space (see W3GATH and W3SCAT in w3wavemd.ftn). These buffers allow that spare clock cycles as may occur during communication can be used for calculation, and that hiding of communication behind calculation will occur if the hardware is capable of doing this. To avoid problems with incompatibilities between FORTRAN and MPI, separate gather and scatter data arrays are used. The buffered data transposes are graphically depicted in Fig. 6.7. More details can be found in Tolman (2002b).

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

Distributed memory also requires modifications to the I/O. Input files are read completely by each separate processor. The type of file output is determined by the I/O type indicator IOSTYP.

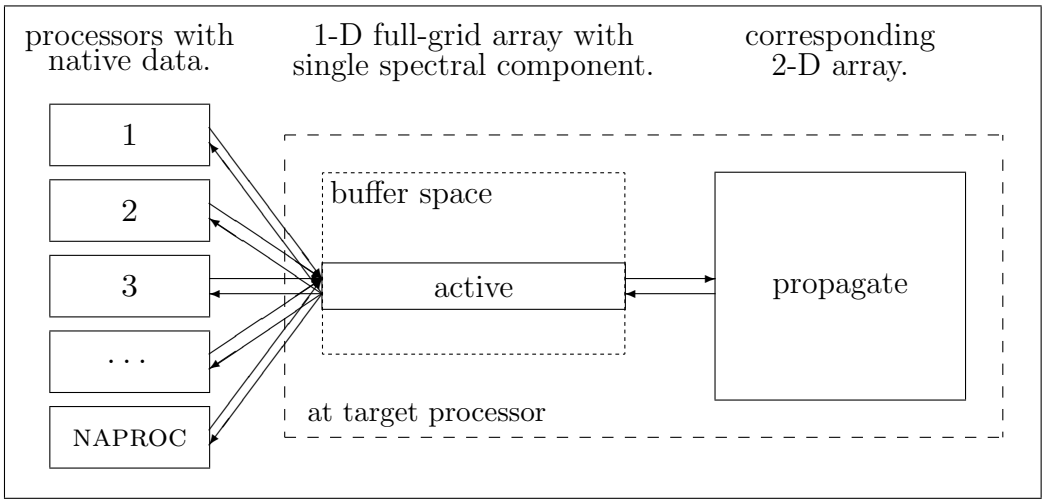

Figure 6.7: Data transpose in distributed memory model version. First, the data is moved from left to right in the figure during the gather operation. After the calculation is performed, the data is moved from right to left in the scatter operation.

IOSTYP	implies
0	Restart file written from each individual process.
1	Each file written from assigned process.
2	Each file written from a single dedicated output process.
3	Dedicated output processes for each output type.

Note that the restart file is a direct access file, so that each processor can efficiently gather only the locally stored spectra, without the need of reading through the entire file. The restart file is either written by each individual process directly, or all data is funneled through a dedicated processor. The first method requires a parallel file system, the second method is generally applicable.

The present algorithm for data distribution has been chosen for several reasons. First, it results in an automatic and efficient load balancing with respect to the (dynamic) integration of source terms, the exclusion of ice covered grid points, and of intra-spectral propagation. Secondly, the communication by definition becomes independent of the numerical propagation scheme, unlike for the more conventional domain decomposition. In the latter case, only a so-called 'halo' of boundary data needs to be converted to

neighboring 'blocks' of grid points. The size of the halo depends on the propagation scheme selected. The main disadvantage of the present data distribution scheme is that the amount of data to be communicated each time step is much larger than for a more conventional domain decomposition, particularly when relatively small numbers of processors are used. On an IBM RS6000 SP, on which the distributed memory version of WAVEWATCH III was tested, the relatively large amount of communication did not constitute a significant part of the overall time of computation, and the model shows excellent scaling behavior for up to O(100) processors (Tolman, 2002b).

More recently, hybrid parallelization techniques have been developed using a combination of a course scale domain decomposition and a local data transpose, using approaches already available in ww3_multi. To accommodate this, the file ww3_gspl(.sh) tools were introduced in model version 4.10. Although this approach still needs some work with respect to the model memory footprint in the initialization in ww3_multi, initial scaling results obtained with this approach are encouraging (see Tolman, 2013b).

6.5.3 Multiple grids

So far, only a single wave model grid has been considered. To make it possible to run several model grids in a single program, a data structure needs to be devised in which all different model grids and internal work arrays for all models are retained simultaneously, with a simple mechanism to choose the actual wave model grid to work on. In order to achieve this, some FORTRAN 90 features (e.g., Metcalf and Reid, 1999) are used in the following way:

1) Define one or more data structures in the model code that contain the model setup and relevant work arrays, using a TYPE declaration.

2) Construct arrays of these data structures, with each element of the array defining a separate model grid.

3) Redefine the basic parameters describing the model such as the number of grid points NX and NY as pointers, and point these to the proper element of the proper data structures to generate instantaneous aliases.

```
!/
!/ Data structures
!/
      TYPE GRID
        INTEGER                :: NX, NY, NSEA
        REAL, POINTER          :: ZB(:)
      END TYPE GRID
!/
!/ Data storage
!/
      TYPE(GRID), TARGET, ALLOCATABLE :: GRIDS(:)
!/
!/ Data aliasses
!/
      INTEGER, POINTER         :: NX, NY, NSEA
      REAL, POINTER            :: ZB(:):
!/
```

Figure 6.8: Example of the data structure declarations used in w3gdatmd.ftn to define multiple spatial grids in the wave model. For simplicity, the example considers only the grid dimensions NX, NY and NSEA, and the bottom depth array ZB.

In this way it is possible to define a multi-model data structure, while keeping the layout of all original variables describing the model unchanged inside the model subroutines. Such a structure and its usage are illustrated in Figs. 6.8 and 6.9 with an example from the actual source code. Note that the pointer arrays like ZB inside the structures are assigned memory as

ALLOCATE GRIDS(IMOD)%ZB(NSEA)

After this statement, the alias pointer ZB again needs to be pointed to the proper element of the structure for this alias to properly point to the newly allocated space. For this reason, the subroutine W3DIMX, which allocates the arrays in this structure, includes at the end a call to the subroutine W3SETX,

```
!
      NX      => GRIDS(IMOD)%NX
      NY      => GRIDS(IMOD)%NY
      NSEA    => GRIDS(IMOD)%NSEA
!
      ZB      => GRIDS(IMOD)%ZB
!
```

Figure 6.9: Example of the source code used to activate the pointer aliases in Fig. 6.8 for the model number IMOD.

---

which in turn sets all pointer aliases for the selected grid. The same is true for other subroutines setting array sizes in other structures.

6.6 Variables in modules

In the documentation of model versions up to version 3.14, all PUBLIC and PRIVATE variables in modules were described in the present and following sections. All these parameters are also documented in the source code of the model. Keeping two separate unlinked copies of the documentations is becoming a daunting task with little benefit to the model user and developer. Hence, from model version 6.07 on, the main documentation of the variables in the code is kept up to date in the source code itself, and second full documentation in the manual is no longer maintained. In this manual, we now only describe PARAMETER definitions, as they may influence model behavior, and identify critical versions of I/O elements of the code. The file name of the module is given at the right margin of the start of each list. The second column of each list identifies the type of the variable. I, R, L and C represent integer, real, logical and character, A identifies an array, and P identifies a PARAMETER declaration. All variables are public, unless marked with *. The following sections account for parameter settings in modules (and programs), and give a top level description of what is stored in the data structures, and where these data structures are located in the code.

6.6.1 Parameter settings in modules

Several modules have internally used parameter settings. Here only parameter settings that are generally usable or impact model behavior are presented.

Physical and mathematical constants : constants.ftn

GRAV	RP	Acceleration of gravity g.	(m s^{-2})
DWAT	RP	Density of water.	(kg m^{-3})
DAIR	RP	Density of air.	(kg m^{-3})
NU_AIR	RP	Kinematic viscosity of air	(m^2 s^{-1})
NU_WATER	RP	Kinematic viscosity of water	(m^2 s^{-1})
SED_SD	RP	Specific gravity of sediment	(–)
KAPPA	RP	Von Karman's constants	(–)
PI	RP	π.	
TPI	RP	2π.	
HPI	RP	0.5π.	
TPIINV	RP	$(2\pi)^{-1}$.	
HPIINV	RP	$(0.5\pi)^{-1}$.	
RADE	RP	Conversion factor from radians to degrees.	
DERA	RP	Conversion factor from degrees to radians.	
RADIUS	RP	Radius of the earth.	(m)
G2PI3I	RP	$g^{-2}(2\pi)^{-3}$.	
G1PI1I	RP	$g^{-1}(2\pi)^{-1}$.	

Wave model initialization module : w3initmd.ftn

CRITOS	RP	Critical fraction of resources used for output only (triggers warning output).
WWVER	CP	Version number of the main program.
SWITCHES	CP	Switches taken from bin/switch.

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

VERGRD	CP*	Version number of file mod_def.ww3.
IDSTR	CP*	ID string for file.

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

VEROGR	CP*	Version number of file out_grd.ww3.

IDSTR	CP*	ID string for file.

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

VEROPT	CP*	Version number of file out_pnt.ww3.
IDSTR	CP*	ID string for file.
ACC	CP	Relative offset below which output point is moved to grid point.

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

VERTRK	CP*	Version number of file track_o.ww3.
IDSTRI	CP*	ID string for file track_i.ww3.
OTYPE	CP	Array dimension.

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

VERINI	CP*	Version number of file restart.ww3.
IDSTR	CP*	ID string for file.

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

VERBPT	CP*	Version number of file nest.ww3.
IDSTR	CP*	ID string for file.

I/O module (partition.ww3) : w3iosfmd.ftn

VERTRT	CP*	Version number of file partition.ww3.
IDSTR	CP*	ID string for file.

Multi-grid model input update : wmupdtmd.ftn

SWPMAX	IP	Maximum number of extrapolation sweeps allowed to make maps match in conversion from input from input grid to wave model grid.

Several routines contain interpolation tables that are set up with parameter statements, including

Solving the dispersion relation : w3dispmd.ftn

NAR1D	IP	Dimension of interpolation tables.

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

Shallow water quadruplet lookup table for GMD : w3snl3md.ftn

NKD	IP	Number of nondimensional depths in storage array.
KDMIN	RP	Minimum relative depth in table.
KDMAX	RP	Maximum relative depth in table.
LAMMAX	RP	Maximum value for λ or μ.
DELTHM	RP	Maximum angle gap θ_{12} ($^\circ$).

Shallow water lookup table for nonlinear filter : w3snlsmd.ftn

NKD	IP	Number of nondimensional depths in storage array.
KDMIN	RP	Minimum relative depth in table.
KDMAX	RP	Maximum relative depth in table.
ABMAX	RP	Maximum value for a_{34}.

Lookup table for β in Tolman and Chalikov 1996 : w3src2md.ftn

NRSIGA	IP	Array dimension (σ_a).
NRDRAG	IP	Array dimension (C_d).
SIGAMX	RP	Maximum nondimensional frequency $\tilde{\sigma}_a$.
DRAGMX	RP	Maximum drag coefficient C_d

Lookup table for ... in WAM-4 / ECWAM : w3src3md.ftn

KAPPA	RP	von Kármán's constant.
NU_AIR	RP	air viscosity.
ITAUMAX	IP	size of stress dimension.
JUMAX	IP	size of wind dimension.
IUSTAR	IP	size of ustar dimension.
IALPHA	IP	size of Charnock dimension.
ILEVTAIL	IP	size of tail level dimension.
UMAX	RP	Maximum wind speed in table.
TAUWMAX	RP	Maximum ustar in table.
EPS1	RP	Small number for stress convergence.

EPS2	RP	Small number for stress convergence.
NITER	IP	Number of iterations in stress table.
XM	IP	power of TAUW/TAU in roughness parameterization.
JTOT	IP	Number of points in discretization of tail.

Lookup tables Ardhuin et al. 2010 : w3src3md.ftn

Combination of previous two sets of parameters.

Table of error functions in bottom friction : w3sbt4md.ftn

SIZEERFTABLE	IP	Size of table for erf function.
XERFMAX	RP	Maximum value of x in table of erf(x).
WSUB	RPA	Weights for 3-point Gauss-Hermitte quadrature.
XSUB	RPA	x values for 3-point Gauss-Hermitte quadrature.

Some model parameters are set using parameter statements.

Source term computation and integration : w3srcemd.ftn

OFFSET	RP*	Offset ϵ in Eq. (3.61).

Auxiliary data storage : w3adatmd.ftn

MPIBUF	IP	Number of buffers used in MPI data transpose.

Some service routines contain parameters that can be used to influence, for instance, the model output.

Array I/O including text outputs : w3arrymd.ftn

ICOL	IP*	Set maximum columns on output (now set to 80).
NFRMAX	IP*	Set maximum number of frequency in spectral print plots (now set to 50).

Automatic unit number assignment : wmunitmd.ftn

UNITLW	IP	Lowest unit number to be considered.
UNITHG	IP	Highest unit number to be considered.
INPLOW, INPHGH	IP	Range of input file unit numbers.
OUTLOW, OUTHGH		

IP Range of output file unit numbers.

SCRLOW, SCRHGH
IP Range of scratch file unit numbers.

Creating spectral bulletins : w3bullmd.ftn

NPTAB, NFLD, NPMAX, BHSMIN, BHSDROP, DHSMAX, DPTMX, DDMMAX, DDWMAX, AGEMIN
I/RP Setting of size of bulletin as well as various filter values.

6.6.2 Data structures

As outlined in Section 6.5.3, the core of the wave model consists of a set of data structures allowing for the consecutive storage of data for multiple grids. The individual storage structures are contained in the following modules:

w3gdatmd.ftn	In formation for spatial and spectral grids, and all physical and numerical model parameters.
w3wdatmd.ftn	The actual wave data, consisting of spectra and the fields like u_* that are needed to hot-start the model.
w3adatmd.ftn	Auxiliary fields and parameters.
w3odatmd.ftn	Output data.
w3idatmd.ftn	Input data.
wmmdatmd.ftn	Data specific to the multi-grid model.

The data structures are fully documented in the above files, and the documentation is no linger reproduced here in the manual.

This page is intentionally left blank.

References

Abdalla, S., and J. R. Bidlot (2002), Wind gustiness and air density effects and other key changes to wave model in CY25R1, *Tech. Rep. Memomrandum R60.9/SA/0273*, Research Department, ECMWF, Reading, U. K.

Abdolali, A., A. Roland, A. van der Westhuysen, Z. Ma, and A. Chawla (2018), Wind wave modeling of storm-induced hurricanes using implicit scheme in WAVEWATCH III model over large scale resolved bathymetries, modeling of storm-induced hurricanes inundation through a comprehensive flexible coupling framework, *98th AMS Annual Meeting - American Meteorological Society*, Austin, TX, USA.

Abdolali, A., A. Roland, A. van der Westhuysen, A. Chawla, S. Moghimi, and S. Vinogradov (2019), On the comparison of implicit with explicit schemes in WAVEWATCH III, *99th AMS Annual Meeting - American Meteorological Society*, Phoenix, AZ, USA.

Aboulali, A., A. Roland, A. van der Westhuysen, J. Meixner, A. Chawla, T. J. Hesser, and J. M. Smith (2019), Large-scale hurricane modeling using domain decomposition parallelization and implicit scheme implemented in WAVEWATCH III wave model, *Ocean Modeling*, **under review**.

Alves, J.-H. G., and M. L. Banner (2003), Performance of a saturation-based dissipation-rate source term in modeling the fetch-limited evolution of wind waves, *J. Phys. Oceanogr.*, **33**(6), 1274–1298, doi:10.1175/1520-0485(2003)033⟨1274:POASDS⟩2.0.CO;2.

Alves, J. H. G., M. L. Banner, and I. R. Young (2003), Revisiting the pierson–moskowitz asymptotic limits for fully developed wind waves, *Journal of physical oceanography*, **33**(7), 1301–1323.

Alves, J.-H. G. M., A. Chawla, H. L. Tolman, D. Schwab, G. Lang, and G. Mann (2014), The operational implementation of a great lakes wave forecasting system at noaa/ncep, *Weather and Forecasting*, **29**(6), 1473–1497, doi:10.1175/WAF-D-12-00049.1.

Ardhuin, F., and A. Boyer (2006), Numerical modelling of sea states: validation of spectral shapes (in French), *Navigation*, **54**, 55–71.

Ardhuin, F., and T. H. C. Herbers (2002), Bragg scattering of random surface gravity waves by irregular seabed topography, *J. Fluid Mech.*, **451**, 1–33.

Ardhuin, F., and A. D. Jenkins (2006), On the interaction of surface waves and upper ocean turbulence, *J. Phys. Oceanogr.*, **36**(3), 551–557.

Ardhuin, F., and R. Magne (2007), Scattering of surface gravity waves by bottom topography with a current, *J. Fluid Mech.*, **576**, 235–264.

Ardhuin, F., and A. Roland (2012), Coastal wave reflection, directional spread, and seismoacoustic noise sources, *J. Geophys. Res.*, **117**, C00J20.

Ardhuin, F., G. Boutin, J. Stopa, F. Girard-Ardhuin, C. Melsheimer, J. Thomson, A. Kohout, M. Doble, and P. Wadhams (), Wave attenuation through an arctic marginal ice zone on 12 october 2015: 2. numerical modeling of waves and associated ice breakup, *J. Geophys. Res.*, **123**(8), 5652–5668, doi:10.1002/2018JC013784.

Ardhuin, F., W. C. O'Reilly, T. H. C. Herbers, and P. F. Jessen (2003), Swell transformation across the continental shelf. Part I: Attenuation and directional broadening, *J. Phys. Oceanogr.*, **33**, 1,921–1,939.

Ardhuin, F., B. Chapron, and F. Collard (2009a), Ocean swell evolution from distant storms, *Geophys. Res. Lett.*, **36**, doi:10.1029/2008GL037030.

Ardhuin, F., L. Marié, N. Rascle, P. Forget, and A. Roland (2009b), Observation and estimation of Lagrangian, Stokes and Eulerian currents induced by wind and waves at the sea surface, *J. Phys. Oceanogr.*, **39**(11), 2,820–2,838.

Ardhuin, F., W. E. Rogers, A. V. Babanin, J. Filipot, R. Magne, A. Roland, A. van der Westhuysen, P. Queffeulou, J. Lefevre, L. Aouf, and F. Collard (2010), Semiempirical dissipation source functions for ocean waves. Part I: Definition, calibration, and validation, *J. Phys. Oceanogr.*, **40**, 1,917–1,941.

Ardhuin, F., J. Hanafin, Y. Quilfen, B. Chapron, P. Queffeulou, M. Obrebski, J. Sienkiewicz, and D. Vandermark (2011a), Calibration of the IOWAGA global wave hindcast (1991-2011) using ECMWF and CFSR winds, in *Proc. 12th Int. Workshop on Wave Forecasting and Hindcasting*, pp. 1–13.

Ardhuin, F., J. Tournadre, P. Queffelou, and F. Girard-Ardhuin (2011b), Observation and parameterization of small icebergs: Drifting breakwaters in the Southern Ocean, *Ocean Mod.*, **39**, 405–410.

Ardhuin, F., A. Rawat, and J. Aucan (2014), A numerical model for free infragravity waves: definition and validation at regional and global scales, *Ocean Mod.*, **77**, 20–32.

Ardhuin, F., F. Collard, B. Chapron, F. Girard-Ardhuin, G. Guitton, A. Mouche, and J. Stopa (2015), Estimates of ocean wave heights and attenuation in sea ice using the SAR wave mode on Sentinel-1A, *Geophys. Res. Lett.*, **42**, 2,317–2,325, doi:10.1002/2014GL062940.

Ardhuin, F., P. Sutherland, M. Doble, and P. Wadhams (2016), Ocean waves across the Arctic: Attenuation due to dissipation dominates over scattering for periods longer than 19 s, *Geophys. Res. Lett.*, **43**(11), 5775–5783, doi:

10.1002/2016GL068204.

Babanin, A., T.-W. Hsu, A. Roland, S.-H. Ou, D.-J. Doong, and C. Kao (2011), Spectral wave modelling of typhoon krosa, *Natural Hazards and Earth System Sciences*, **11**(2), 501–511.

Babanin, A. V. (2009), Breaking of ocean surface waves, *Acta Phys. Slovaca*, **59**(4), 305–535.

Babanin, A. V. (2011), *Breaking and dissipation of ocean surface waves*, Cambridge University Press, 480 pp.

Babanin, A. V., and Y. P. Soloviev (1987), Parameterization of width of directional energy distributions of wind-generated waves at limited fetches, *Izvestiya, Atmos. and Oceanic Phys.*, **23**, 645–651.

Babanin, A. V., and I. R. Young (2005), Two-phase behaviour of the spectral dissipation of wind waves, in *Proc. Fifth Int. Symp. Ocean Waves Measurement and Analysis*.

Babanin, A. V., I. R. Young, and M. L. Banner (2001), Breaking probabilities for dominant surface waves on water of finite constant depth, *J. Geophys. Res.*, **106**, 11,659–11,676, doi:10.1029/2000JC000215.

Babanin, A. V., M. L. Banner, I. R. Young, and M. A. Donelan (2007), Wave-follower field measurements of the wind-input spectral function. Part iii: Parameterization of the wind-input enhancement due to wave breaking, *J. Phys. Oceanogr.*, **37**, 2,764–2,775.

Babanin, A. V., K. N. Tsagareli, I. R. Young, and D. J. Walker (2010), Numerical investigation of spectral evolution of wind waves. Part II: Dissipation term and evolution tests, *J. Phys. Oceanogr.*, **40**, 667–683.

Banner, M. L., and W. L. Peirson (1998), Tangential stress beneath wind-driven air–water interfaces, *J. Fluid Mech.*, **364**, 115–145.

Banner, M. L., A. V. Babanin, and I. R. Young (2000), Breaking probability for dominant waves on the sea surface, *J. Phys. Oceanogr.*, **30**, 3,145–3,160.

Barbariol, F., A. Benetazzo, S. Carniel, and M. Sclavo (2015), Space-time wave extremes: The role of metocean forcings, *J. Phys. Oceanogr.*, **45**(7), 1,897–1,916, doi:10.1175/JPO-D-14-0232.1.

Barbariol, F., J.-H. G. M. Alves, A. Benetazzo, F. Bergamasco, L. Bertotti, S. Carniel, L. Cavaleri, Y. Y. Chao, A. Chawla, A. Ricchi, M. Sclavo, and H. L. Tolman (2016), Numerical modeling of space-time wave extremes using WAVEWATCH III, *Ocean Dynamics*, **Under rev.**

Battjes, J. A., and J. P. F. M. Janssen (1978), Energy loss and set-up due to breaking of random waves, in *Proc. 16th Int. Conf. Coastal Eng.*, pp. 569–587, ASCE.

Benetazzo, A., F. Barbariol, F. Bergamasco, A. Torsello, S. Carniel, and M. Sclavo (2015), Observation of extreme sea waves in a space-time ensemble, *J. Phys. Oceanogr.*, **45**(9), 2,261–2,275, doi:10.1175/JPO-D-15-0017.1.

Bennetts, L. G., and V. A. Squire (2012), On the calculation of an attenuation coefficient for transects of ice-covered ocean, *Proc. R. Soc. A*, **468**(2137), 136–162.

Bertin, X., K. Li, A. Roland, Y. J. Zhang, J. F. Breilh, and E. Chaumillon (2014), A modeling-based analysis of the flooding associated with xynthia, central bay of biscay, *Coastal Engineering*, **94**, 80–89.

Bertin, X., K. Li, A. Roland, and J.-R. Bidlot (2015), The contribution of short-waves in storm surges: Two case studies in the bay of biscay, *Continental Shelf Research*, **96**, 1–15.

Bidlot, J.-R. (2001), Ecmwf wave-model products, *ECMWF Newsletter*, (91).

Bidlot, J.-R. (2012), Present status of wave forecasting at E.C.M.W.F., in *Proceedings of ECMWF workshop on ocean wave forecasting, June*.

Bidlot, J. R., S. Abdalla, and P. A. E. M. Janssen (2005), A revised formulation for ocean wave dissipation in CY25R1, *Tech. Rep. Memorandum R60.9/JB/0516*, Research Department, ECMWF, Reading, U. K.

Bidlot, J. R., P. A. E. M. Janssen, and S. Abdalla (2007), A revised formulation of ocean wave dissipation and its model impact, *Tech. Rep. Memorandum 509*, ECMWF, Reading, U. K.

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

Booij, N., R. C. Ris, and L. H. Holthuijsen (1999), A third-generation wave model for coastal regions, 1. Model description and validation, *J. Geophys. Res.*, **104**, 7,649–7,666.

Boutin, G., F. Ardhuin, D. Dumont, C. Svigny, F. Girard-Ardhuin, and M. Accensi (2018), Floe size effect on wave-ice interactions: Possible effects, implementation in wave model, and evaluation, *Journal of Geophysical Research: Oceans*, **123**(7), 4779–4805, doi:10.1029/2017JC013622.

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**, 1,653–1,658.

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

Bunney, C. C., A. Saulter, and T. Palmer (2013), Reconstruction of complex 2D wave spectra for rapid deployment of nearshore wave models, in *Marine Structures and Breakwaters 2013*, Institute of Civil Engineers.

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

Cavaleri, L., and P. Malanotte-Rizzoli (1981), Wind-wave prediction in shallow water: Theory and applications., *J. Geophys. Res.*, **86**, 10,961–10,973.

Chalikov, D. V. (1995), The parameterization of the wave boundary layer, *J. Phys. Oceanogr.*, **25**, 1,333–1,349.

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

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

Chawla, A., and H. L. Tolman (2007), Automated grid generation for WAVEWATCH III, *Tech. Note 254*, NOAA/NWS/NCEP/MMAB, 71 pp.

Chawla, A., and H. L. Tolman (2008), Obstruction grids for spectral wave models, *Ocean Mod.*, **22**, 12–25.

Chawla, A., H. L. Tolman, V. Gerald, D. Spindler, T. Spindler, J.-H. G. Alves, D. Cao, J. L. Hanson, and E.-M. Devaliere (2013), A multigrid wave forecasting model: A new paradigm in operational wave forecasting, *Weather and Forecasting*, **28**(4), 1057–1078.

Cheng, S., W. E. Rogers, J. Thomson, M. Smith, M. J. Doble, P. Wadhams, A. L. Kohout, B. Lund, O. P. Persson, C. O. Collins, S. F. Ackley, F. Montiel, and H. H. Shen (2017), Calibrating a viscoelastic sea ice model for wave propagation in the Arctic fall marginal ice zone, *J. Geophys. Res.*, **122**, 8770–8793, doi:https://doi.org/10.1002/2017JC013275.

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

Cole, D., R. Johnson, and G. Durell (1998), Cyclic loading and creep response of aligned first-year sea ice, *J. Geophys. Res.*, **103**(C10), 21,751–21,758.

Collins, C. O., and W. E. Rogers (2017), A source term for wave attenuation by sea ice in WAVEWATCH III® : IC4, *NRL Memorandum Report NRL/MR/7320–17-9726*, Naval Research Laboratory, Stennis Space Center, MS, 25 pp., www7320.nrlssc.navy.mil/pubs.php.

Dalrymple, R. A., and P. L. F. Liu (1978), Waves over soft muds: A two-layer fluid model, *J. Phys. Oceanogr.*, **986**, 1,121–1,131.

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

Devaliere, E. M., J. L. Hanson, and R. Luettich (2009), Spatial tracking of

numerical wave model output using a spiral search algorithm, in *2009 WRI World Congress on Computer Science and Information Engineering, Los Angeles, CA*, vol. 2, pp. 404–408, doi:10.1109/CSIE.2009.1021.

Doble, M. J., and J.-R. Bidlot (2013), Wave buoy measurements at the Antarctic sea ice edge compared with an enhanced ECMWF WAM: Progress towards global waves-in-ice modelling, *Ocean Mod.*, **70**, 166–173.

Doble, M. J., G. D. Carolis, M. H. Meylan, J.-R. Bidlot, and P. Wadhams (2015), Relating wave attenuation to pancake ice thickness, using field measurements and model results, *Geophys. Res. Lett.*, **42**, 4473–4481, doi: 10.1002/2015GL063628.

Dodet, G., X. Bertin, N. Bruneau, A. B. Fortunato, A. Nahon, and A. Roland (2013), Wave-current interactions in a wave-dominated tidal inlet, *Journal of Geophysical Research: Oceans*, **118**(3), 1587–1605.

Donelan, M. A. (1999), Wind-induced growth and attenuation of laboratory waves, in *Wind-over-wave couplings: Perspectives and prospects*, pp. 183–194, Clarendon Press.

Donelan, M. A. (2001), A nonlinear dissipation function due to wave breaking, in *Proc. ECMWF Workshop on Ocean Wave Forecasting*, pp. 87–94.

Donelan, M. A., A. V. Babanin, I. R. Young, and M. L. Banner (2006), Wave follower measurements of the wind-input spectral function. Part II. Parameterization of the wind input, *J. Phys. Oceanogr.*, **36**, 1,672–1,689.

Donelan, M. A., M. Curcic, S. S. Chen, and A. K. Magnusson (2012), Modeling waves and wind stress., *J. Geophys. Res.*, **117**, 1–26.

Dore, B. D. (1978), Some effects of the air-water interface on gravity waves, *Geophys. Astrophys. Fluid. Dyn.*, **10**, 215–230.

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

Dumont, D., A. Kohout, and L. Bertino (2011), A wave-based model for the marginal ice zone including a floe breaking parameterization, *J. Geophys. Res.*, **116**(C4), doi:10.1029/2010JC006682, c04001.

Eldeberky, Y. (1995), Personal communication with R. Ris, *Tech. rep.*

Eldeberky, Y. (1996), Nonlinear transformationations of wave spectra in the nearshore zone, Ph.D. thesis, Delft University of Technology, Delft, The Netherlands, 203 pp.

Eldeberky, Y., and J. A. Battjes (1996), Spectral modelling of wave breaking: Application to boussinesq equations, *J. Geophys. Res.*, **101**, 1,253–1,264.

Elgar, S., T. H. C. Herbers, and R. T. Guza (1994), Reflection of ocean surface gravity waves from a natural beach, *J. Phys. Oceanogr.*, **24**, 1,503–

1,511.

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

Fedele, F. (2012), Space–time extremes in short-crested storm seas, *J. Phys. Oceanogr.*, **42**(9), 1,601–1,615.

Fedele, F., G. Gallego, A. Yezzi, A. Benetazzo, L. Cavaleri, M. Sclavo, and M. Bastianini (2012), Euler characteristics of oceanic sea states, *Mathematics and Computers in Simulation*, **82**(6), 1,102–1,111.

Fedele, F., A. Benetazzo, G. Gallego, P.-C. Shih, A. Yezzi, F. Barbariol, and F. Ardhuin (2013), Space-time measurements of oceanic sea states, *Ocean Mod.*, **70**, 103–115.

Ferrarin, C., G. Umgiesser, A. Cucco, T.-W. Hsu, A. Roland, and C. L. Amos (2008), Development and validation of a finite element morphological model for shallow water basins, *Coastal Engineering*, **55**(9), 716–731.

Ferrarin, C., A. Roland, M. Bajo, G. Umgiesser, A. Cucco, S. Davolio, A. Buzzi, P. Malguzzi, and O. Drofa (2013), Tide-surge-wave modelling and forecasting in the mediterranean sea with focus on the italian coast, *Ocean Mod.*, **61**, 38–48.

Filipot, J.-F., and F. Ardhuin (2012a), A unified spectral parameterization for wave breaking: from the deep ocean to the surf zone, *J. Geophys. Res.*, **117**, C00J08, doi:10.1029/2011JC007784.

Filipot, J.-F., and F. Ardhuin (2012b), A unified spectral parameterization for wave breaking: From the deep ocean to the surf zone, *J. Geophys. Res.*, **117**(C11).

Filipot, J.-F., F. Ardhuin, and A. Babanin (2010), A unified deep-to-shallow-water wave-breaking probability parameterization, *J. Geophys. Res.*, **115**(C4).

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

Foreman, M. G. G., J. Y. Cherniawsky, and V. A. Ballantyne (2009), Versatile harmonic tidal analysis: Improvements and applications, *J. Atmos. Oceanic Techn.*, **26**, 806–817.

Forristall, G. Z. (1981), Measurements of a saturated range in ocean wave spectra, *J. Geophys. Res.*, **86**(C9), 8075, doi:10.1029/JC086iC09p08075.

Fox, C., and V. A. Squire (1994), On the oblique reflexion and transmission of ocean waves at shore fast sea ice, *Philosophical Transactions of the Royal*

Society A, **347**(1682), 185–218, doi:10.1098/rsta.1994.0044.

Goda, T. (1970), Numerical experiments on wave statistics with spectral simulation, *Rep. Port Harbour Res. Inst. Jpn.*, **9**, 3–57.

Grant, W. D., and O. S. Madsen (1979), Combined wave and current interaction with a rough bottom, *J. Geophys. Res.*, **84**, 1,797–1,808.

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

Hanson, J. L., and R. E. Jensen (2004), Wave system diagnostics for numerical wave models, in *8 th international workshop on wave hindcasting and forecasting, JCOMM Tech. Rep. 29, WMO/TD-No. 1319*.

Hanson, J. L., and O. M. Phillips (2001), Automated analysis of ocean surface directional wave spectra, *J. Atmos. Oceanic Techn.*, **18**, 177–293.

Hanson, J. L., B. A. Tracy, H. L. Tolman, and D. Scott (2006), Pacific hindcast performance evaluation of three numerical wave models., in *9 th international workshop on wave hindcasting and forecasting, JCOMM Tech. Rep. 34*, Paper A2.

Hanson, J. L., B. A. Tracy, H. L. Tolman, and R. D. Scott (2009), Pacific hindcast performance of three numerical wave models, *Journal of Atmospheric and Oceanic Technology*, **26**(8), 1614–1633.

Hara, T., and S. Belcher (2004), Wind profile and drag coefficient over mature ocean surface wave spectra, *J. Phys. Oceanogr.*, **34**, 2345–2358.

Hardy, T. A., and I. R. Young (1996), Field study of wave attenuation on an offshore coral reef, *J. Geophys. Res.*, **101**, 14,311–14,326.

Hardy, T. A., L. B. Mason, and J. D. McConochie (2000), A wave model for the Great Barrier Reef, *Ocean Eng.*, **28**, 45–70.

Hargreaves, J. C., and J. D. Annan (1998), Integration of source terms in WAM, in *Proceedings of the 5th International Workshop on Wave Forecasting and Hindcasting*, pp. 128–133.

Hargreaves, J. C., and J. D. Annan (2001), Comments on "Improvement of the short fetch behavior in the ocean wave model (WAM)", *J. Atmos. Oceanic Techn.*, **18**, 711–715.

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

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

Hasselmann, K. (1963b), On the non-linear transfer in a gravity wave spectrum, Part 3. Evaluation of energy flux and sea-swell interactions for a

Neuman spectrum, *J. Fluid Mech.*, **15**, 385–398.

Hasselmann, K., T. P. Barnett, E. Bouws, H. Carlson, D. E. Cartwright, K. Enke, J. A. Ewing, H. Gienapp, D. E. Hasselmann, P. Kruseman, A. Meerburg, P. 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)*, **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**, 1,369–1,377.

Hasselmann, S., K. Hasselmann, J. H. Allender, and T. P. Barnett (1985), Computations and parameterizations of the nonlinear energy transfer in a gravity-wave spectrum, Part II: Parameterizations of the nonlinear energy transfer for application in wave models, *J. Phys. Oceanogr.*, **15**, 1,378–1,391.

Herbers, T. H. C., and M. C. Burton (1997), Nonlinear shoaling of directionally spread waves on a beach, *J. Geophys. Res.*, **102**(C9), 21,101–21,114.

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

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

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

Hino, M. (1968), Equilibrium-range spectra of sand waves formed by flowing water, *J. Fluid Mech.*, **34**(3), 565–573.

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

Horvat, C., and E. Tziperman (2015), A prognostic model of sea-ice floe size and thickness distribution, *The Cryosphere*, **9**, 2,119–2,134.

Hwang, P. A. (2011), A note on the ocean surface roughness spectrum, *J. Atmos. Oceanic Techn.*, **28**, 436–443.

J. Larson, E. O., R. Jacob (2005), The model coupling toolkit: A new for-

tran90 toolkit for building multiphysics parallel coupled models, *Int. J. High Perf. Comp. App.*, **19**(3), 277–292.

Janekovic, I., Y. Hetzel, C. Pattiaratchi, E. Wijeratne, and A. Roland (), Unstructured high resolution 2-way coupled storm surge–wave model for western australia.

Janssen, P. A., P. Lionello, M. Reistad, and A. Hollingsworth (1989), Hindcasts and data assimilation studies with the wam model during the seasat period, *Journal of Geophysical Research: Oceans*, **94**(C1), 973–993.

Janssen, P. A. E. M. (1982), Quasilinear approximation for the spectrum of wind-generated water waves, *J. Fluid Mech.*, **117**, 493–506.

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

Janssen, P. A. E. M. (1991), Quasi-linear theory of of wind wave generation applied to wave forecasting, *J. Phys. Oceanogr.*, **21**, 1,631–1,642.

Janssen, P. A. E. M. (2004), *The interaction of ocean waves and wind*, 300 pp., Cambridge University Press.

Janssen, P. A. E. M. (2009), On some consequences of the canonical transformation in the Hamiltonian theory of water waves, *J. Fluid Mech.*, **637**, 1–44.

Jones, P. W. (1998), *A User's Guide for SCRIP: A Spherical Coordinate Remapping and Interpolation Package, Version 1.5*, 29 pp.

Jones, P. W. (1999), First- and second-order conservative remapping schemes for grids in spherical coordinates, *Mon. Wea. Rev.*, **127**, 2,204–2,210.

Kahma, K. K., and C. J. Calkoen (1992), Reconciling discrepancies in the observed growth rates of wind waves, *J. Phys. Oceanogr.*, **22**, 1,389–1,405.

Kahma, K. K., and C. J. Calkoen (1994), *Dynamics and modelling of ocean waves*, chap. II.8 Growth curve observations, pp. 174–182, Cambridge Univ. Press.

Karypis, G. (2011), Metis and parmetis, pp. 1117–1124.

Kerr, P. C., A. S. Donahue, J. J. Westerink, R. A. Luettich, L. Zheng, R. H. Weisberg, Y. Huang, H. Wang, Y. Teng, D. Forrest, et al. (2013), Us ioos coastal and ocean modeling testbed: Inter-model evaluation of tides, waves, and hurricane surge in the gulf of mexico, *Journal of Geophysical Research: Oceans*, **118**(10), 5129–5172.

Kohout, A., M. Williams, S. Dean, and M. Meylan (2014), Storm-induced sea-ice breakup and the implications for ice extent, *Nature*, **509**, 604–607.

Kohout, A. L., and M. H. Meylan (2008), An elastic plate model for wave attenuation and ice floe breaking in the marginal ice zone, *J. Geophys.*

Res., **113**(C9), doi:10.1029/2007JC004434.

Komen, G. J., S. Hasselmann, and K. Hasselmann (1984), On the existence of a fully developed wind-sea spectrum, *J. Phys. Oceanogr.*, **14**, 1,271–1,285.

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

Krasitskii, V. P. (1994), On reduced equations in the Hamiltonian theory of weakly nonlinear surface waves, *J. Fluid Mech.*, **272**, 1–20.

Kreisel, G. (1949), Surface waves, *Quart. Journ. Appl. Math.*, pp. 21–44.

Kuik, A. J., G. P. Van Vledder, and L. Holthuijsen (1988), A method for the routine analysis of pitch-and-roll buoy wave data, *J. Phys. Oceanogr.*, **18**, 1,020–1,034.

Leckler, F. (2013), Observation and modelling of wave breaking, Ph.D. thesis, Université de Bretagne Occidentale, Brest, France, 150 pp.

Leckler, F., F. Ardhuin, C. Peureux, A. Benetazzo, F. Bergamasco, and V. Dulov (2015), Analysis and interpretation of frequency-wavenumber spectra of young wind waves, *J. Phys. Oceanogr.*, **45**, 2484–2496, doi:10.1175/JPO-D-14-0237.1.

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

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

Leonard, B. P., A. P. Lock, and M. K. MacVean (1996), Conservative explicit unrestricted-time-step multidimensional constancy-prexerving advection schemes, *Mon. Wea. Rev.*, **124**, 2,588–2,606.

Li, J., A. L. Kohout, and H. H. Shen (2015), Comparison of wave propagation through ice covers in calm and storm conditions, *Geophys. Res. Lett.*, **42**, 5935–5941, doi:doi:10.1002/2015GL064715.

Li, J., A. L. Kohout, M. J. Doble, P. Wadhams, C. Guan, and H. H. Shen (2017), Rollover of apparent wave attenuation in ice covered seas, *J. Geophys. Res.*, **122**, 8557–8566, doi:https://doi.org/10.1002/2017JC012978.

Li, J. G. (2003), A multiple-cell flat-level model for atmospheric tracer dispersion over complex terrain, *Bound. Layer Meteor.*, **107**, 289–322.

Li, J. G. (2008), Upstream nonoscillatory advection schemes, *Mon. Wea. Rev.*, **136**, 4,709–4,729.

Li, J. G. (2011), Global transport on a spherical multiple-cell grid, *Mon. Wea.*

Rev., **139**, 1,536–1,555.

Li, J. G. (2012), Propagation of ocean surface waves on a spherical multiple-cell grid, *J. Comput. Physics*, **231**(24), 8,262–8,277.

Li, J. G. (2016), Ocean surface waves in an ice-free Arctic Ocean, *Ocean Dynamics*, **66**(8), 989–1,004.

Li, J.-G., and A. Saulter (2014), Unified global and regional wave model on a multi-resolution grid, *Ocean Dynamics*, **64**(11), 1657–1670.

Li, J. G., and A. Saulter (2017), Applications of the spherical multiple-cell grid in ocean surface wave models, in *First Int. workshop on waves, storm surges and coastal hazards*, p. 20, Paper I1.

Liau, J.-M., A. Roland, T.-W. Hsu, S.-H. Ou, and Y.-T. Li (2011), Wave refraction–diffraction effect in the wind wave model wwm, *Coastal Engineering*, **58**(5), 429–443.

Liu, A. K., and E. Mollo-Christensen (1988), Wave propagation in a solid ice pack, *J. Phys. Oceanogr.*, **18**, 1,702–1,712.

Liu, A. K., B. Holt, and P. W. Vachon (1991), Wave propagation in the marginal ice zone: model predictions and comparisons with buoy and synthetic aperture radar data, *J. Geophys. Res.*, **96**(C3), 4,605–4,621.

Liu, Q., A. V. Babanin, Y. Fan, S. Zieger, C. Guan, and I.-J. Moon (2017), Numerical simulations of ocean surface waves under hurricane conditions: Assessment of existing model performance, *Ocean Mod.*, **118**, 73–93.

Liu, Q., W. E. Rogers, A. V. Babanin, I. R. Young, L. Romero, S. Zieger, F. Qiao, and C. Guan (2019), Observation-based source terms in the third-generation wave model WAVEWATCH III: updates and verification, *J. Phys. Oceanogr.*, **49**(2), 489–517, doi:10.1175/JPO-D-18-0137.1.

Liu, Q., W. E. Rogers, A. Babanin, J. Li, and C. Guan (submitted), Spectral modelling of ice-induced wave decay with viscoelastic sea ice models: comparisons and limitations, *Journal of Geophysical Research: Oceans*.

Longuet-Higgins, M. S. (1984), Statistical properties of wave groups in a random sea state, *Phil. Trans. R. Soc. A.*, **312**(1521), 219–250.

Longuet-Higgins, M. S., and R. W. Stewart (1960), Changes in the form of short gravity waves on long waves and tidal currents, *J. Fluid Mech.*, **8**, 565–583.

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

Longuet-Higgins, M. S., and R. W. Stewart (1962), Radiation stress and mass transport in gravity waves, with application to 'surf-beats', *J. Fluid*

Mech., **10**, 529–549.

Madsen, P., and O. Sorensen (1993), Bound waves and triad interactions in shallow water, *J. Ocean Engng.*, **20(4)**, 359–388.

Magne, R., K. Belibassakis, T. H. C. Herbers, F. Ardhuin, W. C. O'Reilly, and V. Rey (2007), Evolution of surface gravity waves over a submarine canyon, *J. Geophys. Res.*, **112**(C1), doi:10.1029/2005JC003035.

Magne, R., F. Ardhuin, and A. Roland (2010), Waves forecast and hincast from global ocean to the beach, *Eur. J. of Environ. and Civil Eng.*, **14**(2), 149–162.

McCowan, J. (1894), On the highest wave of permanent type, *Philosophical Magazine*, **38**, 351–358.

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

Mellor, M. (1986), Mechanical behavior of sea ice, in *The Geophysics of Sea Ice*, edited by N. Untersteiner, NATO ASI Series, pp. 165–281, Springer US, dOI: 10.1007/978-1-4899-5352-0_3.

Mentaschi, L., G. Besio, F. Cassola, and A. Mazzino (2015a), Performance evaluation of wavewatch iii in the mediterranean sea, *Ocean Modelling*, **90**, 82–94.

Mentaschi, L., J. Pérez, G. Besio, F. Mendez, and M. Menendez (2015b), Parameterization of unresolved obstacles in wave modelling: A source term approach, *Ocean Mod.*, **96**, 93–102, doi:10.1016/j.ocemod.2015.05.004.

Mentaschi, L., G. Kakoulaki, M. Vousdoukas, E. Voukouvalas, L. Feyen, and G. Besio (2018a), Parameterizing unresolved obstacles with source terms in wave modeling: A real-world application, *Ocean Mod.*, **126**, 77–84, doi: 10.1016/j.ocemod.2018.04.003.

Mentaschi, L., G. Kakoulaki, M. Vousdoukas, E. Voukouvalas, P. Pereira, and L. Feyen (2018b), On spatial resolution and subscale effects in global wave models, *Ocean Mod.*, p. submitted.

Mentaschi, L., M. Vousdoukas, , L. Feyen, and G. Besio (2018c), alphabeta-lab: automatic estimation of subscale transparencies for the unresolved obstacles source term in ocean wave modelling, *SoftwareX*, p. submitted.

Metcalf, M., and J. Reid (1999), *FORTRAN 90/95 explained, second edition*, Oxford University Press, 341 pp.

Metzger, E. J., O. M. Smedstad, P. G. Thoppil, H. E. Hurlburt, J. A. Cummings, A. J. Wallcraft, L. Zamudio, D. S. Franklin, P. G. Posey, M. W. Phelps, P. J. Hogan, F. L. Bub, and C. J. Dehaan (2014), US Navy operational global ocean and Arctic ice prediction systems, *Oceanogr.*, **27**(3),

32–43.

Meylan, M., L. Bennets, and A. Kohout (2014), In situ measurements and analysis of ocean waves in the Antarctic marginal ice zone, *Geophys. Res. Lett.*, **41**(C14), 5,046–5,051.

Meylan, M. H., and D. Masson (2006), Physics-based parameterization of air-sea momentum flux at high wind speeds and its impact on hurricane intensity predictions, *Ocean Mod.*, **11**, 417–427.

Miche, A. (1944), Mouvements ondulatoire de la mer en profondeur croissante ou décroissante. Forme limite de la houle lors de son déferlement. Application aux digues maritimes. Deuxième partie. Mouvements ondulatoires périodiques en profondeur régulièrement décroissante, *Annales des Ponts et Chaussées*, **114**, 25–78,131–164, 270–292,369–406.

Miles, J. W. (1957), On the generation of surface waves by shear flows, *J. Fluid Mech.*, **3**.

Moon, I. J., I. Ginnis, T. Hara, and B. Thomas (2006), Physics-based parameterization of air-sea momentum flux at high wind speeds and its impact on hurricane intensity predictions, *Mon. Wea. Rev.*, **135**, 2,869–2,878.

Mosig, J. E. M., F. Montiel, and V. A. Squire (2015), Comparison of viscoelastic-type models for ocean wave attenuation in ice-covered seas, *Journal of Geophysical Research: Oceans*, pp. 1–17, doi:10.1002/2015JC010700.Received.

Murray, R. J. (1996), Explicit generation of orthogonal grids for ocean models, *J. Comput. Physics*, **126**(2), 251–273.

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

Nelson, R. C. (1994), Depth limited wave heights in very flat regions, *Coastal Eng.*, **23**, 43–59.

Nelson, R. C. (1997), Height limits in top down and bottom up wave environments., *Coastal Eng.*, **32**, 247–254.

Ng, C. (2000), Water waves over a muddy bed: a two-layer Stokes' boundary layer model, *Coastal Eng.*, **40**(3), 221–242.

O'Reilly, W. C., R. T. Guza, and R. J. Seymour (1999), Wave prediction in the Santa Barbara Channel, in *5th California Islands Symposium, March 29-31*, Mineral Management Service, Santa Barbara CA.

Perrie, W., and D. T. Resio (2009), A two-scale approximation for efficient representation of nonlinear energy transfers in a wind wave spectrum. Part II: Application to observed wave spectra, *J. Phys. Oceanogr.*, **39**(10), 2,451–2,476, doi:10.1175/2009JPO3947.1.

Perrie, W., B. Toulany, D. T. Resio, A. Roland, and J.-P. Auclair (2013a), A two-scale approximation for wave-wave interactions in an operational wave model, *Ocean Mod.*, **70**, 38 – 51, doi:http://dx.doi.org/10.1016/j.ocemod.2013.06.008.

Perrie, W., B. Toulany, D. T. Resio, A. Roland, and J.-P. Auclair (2013b), A two-scale approximation for wave–wave interactions in an operational wave model, *Ocean Mod.*, **70**, 38–51.

Perrie, W., B. Toulany, A. Roland, M. Dutour-Sikiric, C. Chen, R. C. Beardsley, J. Qi, Y. Hu, M. P. Casey, and H. Shen (2018), Modeling north atlantic nor'easters with modern wave forecast models, *Journal of Geophysical Research: Oceans*, **123**(1), 533–557.

Peureux, C., F. Ardhuin, and P. V. Guimaraes (2019), On the modulation and steepening of short gravity waves by longer waves, *J. Phys. Oceanogr.*, **submitted**.

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

Phillips, O. M. (1984), On the response of short ocean wave components at a fixed wavenumber to ocean current variations, *J. Phys. Oceanogr.*, **14**, 1,425–1,433.

Phillips, O. M. (1985), Spectral and statistical properties of the equilibrium range in wind-generated gravity waves, *J. Fluid Mech.*, **156**, 505–531.

Pierson, W. J., and L. Moskowitz (1964), A proposed spectral form for fully developed wind seas based on the similarity theory of S. A. Kitaigorodskii., *J. Geophys. Res.*, **69**, 5,181–5,190.

Polnikov, V. G., and I. V. Lavrenov (2007), Calculation of the nonlinear energy transfer through the wave spectrum at the sea surface covered with broken ice, *Oceanol.*, **47**, 334–343.

R. Jacob, E. O., J. Larson (2005), M×N communication and parallel interpolation in CCSM3 using the model coupling toolkit., *Int. J. High Perf. Comp. App.*, **19**(3), 293–307.

Rasch, P. J. (1994), Conservative shape-preserving two-dimensional transport on a spherical reduced grid, *Mon. Wea. Rev.*, **122**, 1,337–1,350.

Rascle, N., and F. Ardhuin (2013), A global wave parameter database for geophysical applications. Part 2: Model validation with improved source term parameterization, *Ocean Mod.*, **70**, 174–188.

Reichl, B. G., T. Hara, and I. Ginis (2014), Sea state dependence of the wind stress over the ocean under hurricane winds, *J. Geophys. Res.*, **119**(1), 30–51.

Resio, D. T., and W. Perrie (1991), A numerical study of nonlinear energy fluxes due to wave–wave interactions. Part 1: Methodology and basic results, *J. Fluid Mech.*, **223**, 603–629.

Resio, D. T., and W. Perrie (2008), A two-scale approximation for efficient representation of nonlinear energy transfers in a wind wave spectrum. Part I: Theoretical development, *J. Phys. Oceanogr.*, **38**(12), 2,801–2,816, doi: 10.1175/2008JPO3713.1.

Resio, D. T., C. E. Long, and W. Perrie (2011), The role of nonlinear momentum fluxes on the evolution of directional wind-wave spectra, *J. Phys. Oceanogr.*, **41**(4), 781–801, doi:10.1175/2010JPO4545.1.

Ricchiuto, M., Á. Csík, and H. Deconinck (2005), Residual distribution for general time-dependent conservation laws, *Journal of Computational Physics*, **209**(1), 249–289.

Roe, P. L. (1986), Characteristic-based schemes for the Euler equations, *Ann. Rev. Fluid Mech*, **18**, 337–365.

Rogers, W. E. (2017), Mean square slope in SWAN and WAVEWATCH III®; buoy response functions; and limitations of ST1 physics in SWAN, in *Waves in Shallow Water Environments Meeting (WISE-24)*, Victoria, Canada.

Rogers, W. E., and T. J. Campbell (2009), Implementation of curvilinear coordinate system in the WAVEWATCH III model, *NRL Memorandum Report NRL/MR/7320–09-9193*, Naval Research Laboratory, Stennis Space Center, MS, 42 pp., www7320.nrlssc.navy.mil/pubs.php.

Rogers, W. E., and K. Holland (2009), A study of dissipation of wind-waves by mud at Cassino Beach, Brazil: Prediction and inversion, *Cont. Shelf Res.*, **29**, 676–690.

Rogers, W. E., and M. D. Orzech (2013), Implementation and testing of ice and mud source functions in WAVEWATCH III model, *NRL Memorandum Report NRL/MR/7320–13-9462*, Naval Research Laboratory, Stennis Space Center, MS, 31 pp., www7320.nrlssc.navy.mil/pubs.php.

Rogers, W. E., and S. Zieger (2014), New wave-ice interaction physics in WAVEWATCH III, in *22nd IAHR International Symposium on Ice, Singapore, August 11 to 15, 2014*, IAHR, 8 pp., www7320.nrlssc.navy.mil/pubs.php.

Rogers, W. E., A. V. Babanin, and D. W. Wang (2012), Observation-consistent input and whitecapping dissipation in a model for wind-generated surface waves: Description and simple calculations, *J. Atmos. Oceanic Techn.*, **29**, 1,329–1,346.

Rogers, W. E., J. D. Dykes, and P. A. Wittmann (2014), Us navy global and regional wave modeling, *Oceanography*, **27**(3), 56–67.

Rogers, W. E., J. Thomson, H. H. Shen, M. J. Doble, P. Wadhams, and S. Cheng (2016), Dissipation of wind waves by pancake and frazil ice in the autumn Beaufort Sea, *J. Geophys. Res.*, **121**, 7991–8007, doi:doi:10.1002/2016JC012251.

Rogers, W. E., P. Posey, L. Li, and R. A. Allard (2018a), Forecasting and hindcasting waves in and near the marginal ice zone: Wave modeling and the ONR "Sea State" field experiment, *NRL Memorandum Report NRL/MR/7320–18-9786*, Naval Research Laboratory, Stennis Space Center, MS, 179 pp., www7320.nrlssc.navy.mil/pubs.php.

Rogers, W. E., M. H. Meylan, and A. L. Kohout (2018b), Frequency distribution of dissipation of energy of ocean waves by sea ice using data from Wave Array 3 of the ONR "Sea State" field experiment, *NRL Memorandum Report NRL/MR/7320–18-9801*, Naval Research Laboratory, Stennis Space Center, MS, 25 pp., www7320.nrlssc.navy.mil/pubs.php.

Roland, A. (2009), Development of WWM II: Spectral wave modelling on unstructured meshes, Ph.D. thesis, Technische Universität Darmstadt, Institute of Hydraulic and Water Resources Engineering, 212 pp.

Roland, A. (2012), Application of residual distribution (RD) schemes to the geographical part of the wave action equation, in *Proceedings of ECMWF workshop on ocean wave forecasting, June*.

Roland, A., and F. Ardhuin (2014a), On the developments of spectral wave models: numerics and parameterizations for the coastal ocean, *Ocean Dynamics*, **64**(6), 833–846.

Roland, A., and F. Ardhuin (2014b), On the developments of spectral wave models: numerics and parameterizations for the coastal ocean, *Ocean Dynamics*, **64**(6), 833–846.

Roland, A., U. Zanke, T.-W. Hsu, S.-H. Ou, and J.-M. Liau (2006a), Spectral wave modelling on unstructured grids with the wwm (wind wave model) i: The deep water case, in *Third Chinese–German Joint Symposium on Coastal and Ocean Engineering, Tainan, Taiwan*.

Roland, A., U. Zanke, T.-W. Hsu, S.-H. Ou, J.-M. Liau, and S.-K. Wang (2006b), Verification of a 3rd generation fem spectral wave model for shallow and deep water applications, in *25th International Conference on Offshore Mechanics and Arctic Engineering*, pp. 487–499, American Society of Mechanical Engineers.

Roland, A., A. Cucco, C. Ferrarin, T.-W. Hsu, J.-M. Liau, S.-H. Ou,

G. Umgiesser, and U. Zanke (2009), On the development and verification of a 2-d coupled wave-current model on unstructured meshes, *Journal of Marine Systems*, **78**, S244–S254.

Roland, A., Y. J. Zhang, H. V. Wang, Y. Meng, Y.-C. Teng, V. Maderich, I. Brovchenko, M. Dutour-Sikiric, and U. Zanke (2012), A fully coupled 3d wave-current interaction model on unstructured grids, *Journal of Geophysical Research: Oceans*, **117**(C11).

Roland, A., M. Dutour-Sikiric, A. Aboulali, J. M. Smith, T. Hesser, H. Alvez, F. Ardhuin, A. Chawla, J.-F. Fillipot, H. Michaud, and F. Leckler (2019), Improving downscaling efficiency of WAVEWATCH III on unstructured grids., *Geoscientific Model Development*, **submitted**.

Saha, S., S. Moorthi, H. Pan, X. Wu, J. Wang, S. Nadiga, P. Tripp, R. Kistler, J. Woollen, D. Behringer, H. Liu, D. Stokes, R. Grumbine, G. Gayno, J. Wang, Y. Hou, H. Chuang, H. Juang, J. Sela, M. Iredell, R. Treadon, D. Kleist, P. V. Delst, D. Keyser, J. Derber, M. Ek, J. Meng, H. Wei, R. Yang, S. Lord, H. van den Dool, A. Kumar, W. Wang, C. Long, M. Chelliah, Y. Xue, B. Huang, J. Schemm, W. Ebisuzaki, R. Lin, P. Xie, M. Chen, S. Zhou, W. Higgins, C. Zou, Q. Liu, Y. Chen, Y. Han, L. Cucurull, R. Reynolds, G. Rutledge, and M. Goldberg (2010), The NCEP climate forecast system reanalysis, *Bull. Am. Meteor. Soc.*, **91**, 1,015–1,057.

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

Sikirić, M. D., A. Roland, I. Tomaz, I. Janeković, et al. (2012), Hindcasting the adriatic sea near-surface motions with a coupled wave-current model, *Journal of Geophysical Research: Oceans*, **117**(C12).

Sikirić, M. D., A. Roland, I. Janeković, I. Tomazić, and M. Kuzmić (2013), Coupling of the regional ocean modeling system (roms) and wind wave model, *Ocean Mod.*, **72**, 59–73.

Sikirić, M. D., D. Ivanković, A. Roland, S. Ivatek-Šahdan, and M. Tudor (2018), Operational wave modelling in the adriatic sea with the wind wave model, *Pure and Applied Geophysics*, pp. 1–15.

Smith, J. M., T. Hesser, M. A. Bryant, A. Roland, A. Chawla, A. Abdolali, H. Alves, and A. van der Westhuysen (2018), Validation of unstructured WAVEWATCH III for nearshore waves, *Proceeding of 36th International Conference on Coastal Engineering (ICCE2018), ASCE*, Baltimore, MD, USA.

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.

Soulsby, R. (1997), *Dynamics of marine sands, a manual for practical applications*, 256 pp., Thomas Telford Publications, London.

Stopa, J. E. (2018), Wind forcing calibration and wave hindcast comparison using multiple reanalysis and merged satellite wind datasets, *Oceanogr. Meteorol.*, **127**, 55–69, doi:10.1016/j.ocemod.2018.04.008.

Stopa, J. E., F. Ardhuin, and F. Girard-Ardhuin (2016), Wave climate in the Arctic 19922014: seasonality and trends, *The Cryosphere*, **10**, 1605–1629.

Teixeira, M. A. C., and S. E. Belcher (2002), On the distortion of turbulence by a progressive surface wave, *J. Fluid Mech.*, **458**, 229–267.

The WAVEWATCH III® Development Group (WW3DG) (2019), User manual and system documentation of WAVEWATCH III ® version 6.07, *Tech. Note 333*, NOAA/NWS/NCEP/MMAB, College Park, MD, USA, 465 pp. + Appendices.

Thornton, E. B., and R. T. Guza (1983), Transformation of wave height distribution, *J. Geophys. Res.*, **88**, 5,925–5,938.

Tolman, H. L. (1990), The influence of unsteady depths and currents of tides on wind wave propagation in shelf seas, *J. Phys. Oceanogr.*, **20**, 1,166–1,174.

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

Tolman, H. L. (1992), Effects of numerics on the physics in a third-generation wind-wave model, *J. Phys. Oceanogr.*, **22**, 1,095–1,111.

Tolman, H. L. (1994), Wind-waves and moveable-bed bottom-friction, *J. Phys. Oceanogr.*, **24**, 994–1,009.

Tolman, H. L. (1995a), Sub-grid modeling of moveable-bed bottom-friction in wind-wave models, *Coastal Eng.*, **26**, 57–75.

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

Tolman, H. L. (2001), Improving propagation in ocean wave models, in *4th International Symposium on Ocean Wave Measurement and Analysis*, edited by B. L. Edge and J. M. Hemsley, pp. 507–516, ASCE.

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

Tolman, H. L. (2002b), Distributed memory concepts in the wave model

WAVEWATCH III, *Parallel Computing*, **28**, 35–52.

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

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

Tolman, H. L. (2002e), User manual and system documentation of WAVEWATCH III version 2.22, *Tech. Note 222*, NOAA/NWS/NCEP/MMAB, 133 pp.

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

Tolman, H. L. (2003a), Optimum Discrete Interaction Approximations for wind waves. Part 1: Mapping using inverse modeling, *Tech. Note 227*, NOAA/NWS/NCEP/MMAB, 57 pp. + Appendices.

Tolman, H. L. (2003b), Treatment of unresolved islands and ice in wind wave models, *Ocean Mod.*, **5**, 219–231.

Tolman, H. L. (2004), Inverse modeling of Discrete Interaction Approximations for nonlinear interactions in wind waves, *Ocean Mod.*, **6**, 405–422.

Tolman, H. L. (2005), Optimum Discrete Interaction Approximations for wind waves. Part 2: Convergence of model integration, *Tech. Note 247*, NOAA/NWS/NCEP/MMAB, 74 pp. + Appendices.

Tolman, H. L. (2006), Toward a third release of WAVEWATCH III; a multi-grid model version, in *9th international workshop on wave hindcasting and forecasting, JCOMM Tech. Rep. 34*, Paper L1.

Tolman, H. L. (2007), Development of a multi-grid version of WAVEWATCH III, *Tech. Note 256*, NOAA/NWS/NCEP/MMAB, 194 pp. + Appendices.

Tolman, H. L. (2008a), Optimum Discrete Interaction Approximations for wind waves. Part 3: Generalized multiple DIAs, *Tech. Note 269*, NOAA/NWS/NCEP/MMAB, 117 pp.

Tolman, H. L. (2008b), A mosaic approach to wind wave modeling, *Ocean Mod.*, **25**, 35–47.

Tolman, H. L. (2009a), Practical nonlinear interaction algorithms, in *11th international workshop on wave hindcasting and forecasting & coastal hazards symposium, JCOMM Tech. Rep. 52, WMO/TD-No. 1533*, Paper J2.

Tolman, H. L. (2009b), User manual and system documentation of WAVEWATCH III ™ version 3.14, *Tech. Note 276*, NOAA/NWS/NCEP/MMAB, 194 pp. + Appendices.

Tolman, H. L. (2010a), WAVEWATCH III ® development best practices,

Tech. Note 286, Ver. 0.1, NOAA/NWS/NCEP/MMAB, 19 pp.

Tolman, H. L. (2010b), Optimum Discrete Interaction Approximations for wind waves. Part 4: Parameter optimization, *Tech. Note 288*, NOAA/NWS/NCEP/MMAB, 175 pp.

Tolman, H. L. (2010c), A genetic optimization package for the Generalized Multiple DIA in WAVEWATCH III ®, *Tech. Note 289, Ver. 1.0*, NOAA/NWS/NCEP/MMAB, 21 pp.

Tolman, H. L. (2011a), On the impact of nonlinear interaction parameterizations in practical wave models, in *12th international workshop on wave hindcasting and forecasting & 3rd coastal hazards symposium*, paper I15, 10 pp.

Tolman, H. L. (2011b), A conservative nonlinear filter for the high-frequency range of wind wave spectra, *Ocean Mod.*, **39**, 291–300.

Tolman, H. L. (2013a), A Generalized Multiple Discrete Interaction Approximation for resonant four-wave nonlinear interactions in wind wave models with arbitrary depth, *Ocean Mod.*, **70**, 11–24.

Tolman, H. L. (2013b), Scaling of WAVEWATCH III ® on massively-parallel computer architectures. Part I: hybrid parallelization., *Tech. note*, NOAA/NWS/NCEP/MMAB, in Preparation.

Tolman, H. L. (2014a), WAVEWATCH III ® development best practices, *Tech. Note 286, Ver. 1.1*, NOAA/NWS/NCEP/MMAB, 23 pp.

Tolman, H. L. (2014b), A genetic optimization package for the Generalized Multiple DIA in WAVEWATCH III ®, *Tech. Note 289, Ver. 1.4*, NOAA/NWS/NCEP/MMAB, 21 pp. + Appendix.

Tolman, H. L. (2014c), WAVEWATCH III ® development best practices, *Tech. Note 286, Ver. 1.1*, NOAA/NWS/NCEP/MMAB, 23 pp.

Tolman, H. L., and J. H. G. M. Alves (2005), Numerical modeling of wind waves generated by tropical cyclones using moving grids, *Ocean Mod.*, **9**, 305–323.

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

Tolman, H. L., and D. V. Chalikov (1996), Source terms in a third-generation wind-wave model, *J. Phys. Oceanogr.*, **26**, 2,497–2,518.

Tolman, H. L., and R. W. Grumbine (2013), Holistic genetic optimization of a Generalized Multiple Discrete Interaction Approximation for wind waves, *Ocean Mod.*, **70**, 25–37.

Tolman, H. L., and V. M. Krasnopolsky (2004), Nonlinear interactions in

practical wind wave models, in *8th international workshop on wave hindcasting and forecasting, JCOMM Tech. Rep. 29, WMO/TD-No. 1319*, Paper E1.

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

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

Tracy, B., E.-M. Devaliere, T. Nicolini, H. L. Tolman, and J. L. Hanson (2007), Wind sea and swell delineation for numerical wave modeling, in *10th international workshop on wave hindcasting and forecasting & coastal hazards symposium, JCOMM Tech. Rep. 41, WMO/TD-No. 1442*, Paper P12.

Tracy, F. T., B. Tracy, and D. T. Resio (2006), ERDC MSRC Resource, *Tech. Rep. Fall 2006*, US Army Corps of Engineers.

Tsagareli, K. N., A. V. Babanin, D. J. Walker, and I. R. Young (2010), Numerical investigation of spectral evolution of wind waves. Part I: Wind-input source function, *J. Phys. Oceanogr.*, **40**, 656–666.

Tuomi, L., H. Pettersson, C. Fortelius, K. Tikka, J.-V. Björkqvist, and K. K. Kahma (2014), Wave modelling in archipelagos, *Coastal Engineering*, **83**, 205–220.

Valcke, S. (2013), The OASIS3 coupler: a European climate modelling community software, *Geosci. Model Dev.*, **6**, 373–388.

Van der Westhuysen, A. J., J. L. Hanson, and E. M. Devaliere (2016), Spatial and temporal tracking of wave systems, *J. Atmos. Oceanic Techn.*, 'In prep.

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

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

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

Veras Guimarães, P., F. Ardhuin, P. Sutherland, M. Accensi, M. Hamon,

Y. Pérignon, J. Thomson, A. Benetazzo, and P. Ferrant (2018), A surface kinematics buoy (skib) for wave–current interaction studies, *Ocean Science*, **14**(6), 1449–1460.

Vincent, L., and P. Soille (1991), Watersheds in digital spaces: An efficient algorithm based on immersion simulations, *IEEE Transactions of Pattern Analysis and Machine Intelligence*, **13**, 583–598.

Voorrips, A. C., V. K. Makin, and S. Hasselmann (1997), Assimilation of wave spectra from pitch-and-roll buoys in a North Sea wave model, *J. Geophys. Res.*, **102**, 5,829–5,849.

Wadhams, P. (1973), Attenuation of swell by sea ice, *J. Geophys. Res.*, **78**(18), 3,552–3,563.

Wadhams, P. (1975), Airborne laser profiling of swell in an open ice field, *J. Geophys. Res.*, **80**(33), 4,520–4,528.

Wadhams, P., V. Squire, D. Goodman, A. Cowan, and S. Moore (1988), The attenuation rates of ocean waves in the marginal ice zone, *J. Geophys. Res.*, **93**, 6,799–6,818.

WAMDIG (1988), The WAM model – A third generation ocean wave prediction model, *J. Phys. Oceanogr.*, **18**, 1,775–1,810.

Wang, R., and H. H. Shen (2010), Gravity waves propagating into an ice-covered ocean: A viscoelastic model, *J. Geophys. Res.*, **115**(C6).

Wang, Y., B. Holt, W. E. Rogers, J. Thomson, and H. H. Shen (2016), Wind and wave influences on sea ice floe size and leads in the Beaufort and Chukchi Seas during the summer-fall transition, *J. Geophys. Res.*, **121**, 1502–1525, doi:doi:10.1002/2015JC011349.

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

Wessel, P., and W. Smith (1996), A global self-consistent hierarchical high resolution shoreline database, *J. Geophys. Res.*, **101**, 8,741–8,743.

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

Williams, T. D., L. G. Bennetts, V. A. Squire, D. Dumont, and L. Bertino (2013), Wave-ice interactions in the marginal ice zone. Part 1: Theoretical foundations, *Ocean Mod.*, **70**(1), 81–91, doi:10.1016/j.ocemod.2013.05.010.

WISE Group (2007), Wave modelling - the state of the art, *Progress in Oceanography*, **75**, 603–674.

WMO (2001), WMO MANUAL ON CODES VOLUME I - International codes; part B - binary codes, *Publication Number 306*, WMO.

Wu, J. (1982), Wind-stress coefficients over sea surface from breeze to hurri-

cane, *J. Geophys. Res.*, **87**, 9,704–9,706.

Young, I. R., and A. V. Babanin (2006), Spectral distribution of energy dissipation of wind-generated waves due to dominant wave breaking, *J. Phys. Oceanogr.*, **36**, 376–394.

Zanke, U., A. Roland, T.-W. Hsu, S.-H. Ou, and J.-M. Liau (2006), Spectral wave modelling on unstructured grids with the wwm (wind wave model)ii: The shallow water cases, in *Third Chinese–German Joint Symposium on Coastal and Ocean Engineering (JOINT2006), Tainan, Taiwan*.

Zieger, S., A. V. Babanin, W. E. Rogers, and I. R. Young (2015), Observation-based source terms in the third-generation wave model WAVEWATCH, *Ocean Mod.*, **96**, 2–25.

Zieger, S., D. Greenslade, and J. D. Kepert (2018), Wave ensemble forecast system for tropical cyclones in the australian region, *Ocean Dynamics*, pp. 1–23.

Zijlema, M. (2010), Computation of wind-wave spectra in coastal waters with SWAN on unstructured grids, *Coastal Eng.*, **57**(3), 267 – 277.

APPENDICES

This page is intentionally left blank.

A Setting model time steps

Model time steps are set on a grid-by-grid basis and are considered as a part of the model setup in the model definition file mod_def.ww3. This implies that in a multi-grid model set-up (using the model driver ww3_multi) each grid is associated with its own time step setting. In this section some guidance is given for setting time steps for individual grids, and for grids in a mosaic approach. Examples of practical time step setting for practical grids can be found in the individual grids used in the test cases mww3_case_01 through mww3_case_03.

A.1 Individual grids

A basic wave model grid requires the definition of four time steps as is described in Section 3.2 on page 121 of this manual. Typically, the first step to consider is the CFL time step for spatial propagation, that is, the second of the four time steps defined in ww3_grid.inp for the grid considered. The critical CFL number C_c that identifies stability of the numerical scheme is defined as [compare Eq. (3.16)]

$$C_c = \frac{c_{g,\max}\Delta t}{\min(\Delta x, \Delta y)} \quad , \tag{A.1}$$

where $c_{g,\max}$ is the maximum group velocity, and Δt, Δx, and Δy are time and space increments. The maximum group velocity is the group velocity for the lowest discrete model frequency. Noting that for a given frequency the largest group velocity occurs in intermediate water depth, this maximum velocity is approximately 1.15 times the deep water group velocity for the lowest discrete spectral frequency. Note that the CFL number formally includes affects of currents [Eq. (2.9)] and grid movement [Eq. (3.45)]. The latter two effects are accounted for internally in the model by adjusting the corresponding minimum time step dynamically depending on the current velocity and the grid movement speed. Hence, the user can define this minimum propagation time step ignoring currents and grid movement. For the schemes used here the critical CFL number is 1.

The second time step to consider is the overall time step (the first time step identified in ww3_grid.inp). For maximum numerical accuracy, this time step should be set smaller than or equal to the above CFL time step. However,

particularly in spherical grids, the critical CFL condition occurs only in a few grid points. In most grid points, CFL numbers will be much smaller. In such grids, accuracy does not suffer significantly if the overall time step is take as 2 to 4 times the critical CFL time steps. Such a setting generally has a major positive impact on model economy. The key to numerical accuracy is the interpretation of the CFL number. This number represents the normalized distance over which information propagates in a single time step. Inaccuracy occurs if information propagates over several grid boxes before source terms are applied. With CFL ≈ 1 and the overall time step four times the CFL time step, information will propagate over four grid boxes before source terms are applied. This may lead to model inaccuracies. If, however, the maximum CFL number is 1, but the average CFL number is only 0.25, as is the case even for the lowest frequency in many spherical grids, information only propagates over one grid box in a single overall time step, and no issues with accuracy develop.

An effective overall time step also considers requested time intervals at which model forcing is available, and at which model output is requested. If input and output time steps are multiple integer times the overall time step, a balanced and consistent numerical integration scheme exists, although the model does not require this. Most important in this consideration is reproducibility of results. If input or output time steps are modified so that they are no longer an integer multiple of the overall model time step, then the actual discrete time stepping in the model will be modified by these input and output time steps, and hence an impact on actual model results may be expected. Such an impact may be notable, but is generally very minor.

The third time step to consider is the maximum refraction (and wavenumber shift) time step. For maximum model economy, this time step should be set equal to (or larger than) the overall time step. However, this will alternate the order of spatial and refraction computations for consecutive model time steps, which in cases of strong refraction may lead to a minor oscillation of wave parameter with a period of $2\Delta t$. Such oscillations can be avoided altogether by setting the maximum refraction time step to half the overall time step. Considering the minor cost of the refraction term in the model, this generally has a negligible impact on model economy. The preferred refraction time step is therefore half the overall model time step.

One note of caution is appropriate with setting this time step. To assure numerical stability, the characteristic refraction velocities are filter as in Eq. (3.51). This filtering suppresses refraction in cases with rapidly changing

bottom topography. The impact of this filtering is reduced when the refraction time step is reduced. It is therefore prudent to test a model grid with much smaller intra-spectral model time steps to assess the impact of this filtering.

The final time step to set is the minimum time step for the dynamical source term integration in Section 3.6. This is a safety valve to avoid prohibitively small time steps in the source term integration. Depending on the grid increment size this is typically set to 5 to 15s. Note that increasing this time step does not necessarily improve model economy; a larger minimum source term integration time step will increase the spectral noise in the integration, which in turn may *reduce* the average source term integration time step!

A.2 Mosaics of grids

Considerations for time step settings for individual grids making up a mosaic model using ww3_multi are in principle identical to those for individual grids as discussed in the previous section. Additional considerations are:

- Overall time steps for individual grids do not need to 'match' in any way for the management algorithm for the mosaic approach to work properly. However, if identically ranked grids share overall time steps, and if integer ratios between time steps of grids with different ranks are employed, then it will be much easier to follow and predict the working of the management algorithm,

- If two grids with identical rank overlap, then the required width of the overlap area will be defined by the stencil width of the numerical scheme, and the number of times this scheme is called for the longest wave component (ratio of overall time step to maximum CFL time step). Thus, model economy for individual grids will improve with increased overall model time step, but the required overlap of equally ranked grids will then increase, reducing the economy of the mosaic approach.

This page is intentionally left blank.

B Setting up nested runs

B.1 Using ww3_shel

The mechanics of running nested models using the single-grid wave model program ww3_shel in principle is simple. A large scale model produces a file with boundary data, for instance nest1.ww3. This file is then renamed to nest.ww3 and put in the directory in which the nested (small scale) model is run. The small scale model then will automatically process the file and update the boundary conditions as required and available. Setting up the nesting consistently is more involved. A simple step-by-step method is presented here. Another possibility, described in the next subsection is to assemble the nest.ww3 file from spectral output using `ww3_bound`.

1) The first step is to set up the large scale model completely, but without generating boundary data for the nested model(s). Include the proper wind fields, graphical outputs etc. Test this model until you are satisfied that it works properly.

2) Set up the small scale model, for the moment ignoring the boundary conditions. Take into consideration that the boundary conditions ideally should coincide with grid lines in the large scale model to minimize the file size of the boundary data files. Set up this model in the same way as the large scale model, and test it thoroughly.

3) When the small scale model is set up satisfactorily in the above way, the boundary conditions need to be defined. Go into the file ww3_grid.inp for the small scale model, and mark all the intended input boundaries as outlined in the documentation in section 4.4.3. Make sure that the model switch !/O1 is selected in the switch file, and recompile if necessary. Run `ww3_grid` and save the screen output. The output of this program now includes a list of all points that are marked as input boundary points. Also make sure that stored copies of mod_def.ww3 for the small scale model (if any) are properly updated.

4) The next step is to include all the input boundary points in the above list as output boundary points in the large scale model. Keep

the list handy, and go to the file ww3_grid.inp for the large scale model. Add all points of the above list as output boundary points as indicated in the documentation in section 4.4.3. Make sure that all data (an no other data) is sent to a single file, and run `ww3_grid` with the proper input file. This should now give a list of output boundary points that should be consistent with the above list of input boundary points. Note that the order in which the points occur in the list is inconsequential. Again make sure that stored copies of mod_def.ww3 for the large scale model (if any) are properly updated.

5) If there are discrepancies between the two lists of points, iterate between the two previous steps until the list are consistent.

6) The next step is to start generating the boundary data from the large scale model. This requires the nesting output to be activated in the large scale model. The output is already set up and included in the model definition file (mod_def.ww3) of the large scale model in the above steps. It now needs to be activated by setting the beginning time, time increment and ending time in the input file ww3_shel.inp for the actual model run of the large scale model. This step does not need to be performed if a second or consecutive nest is added. The large scale model will now produce the file with boundary data. If this is the first nest included the output file will be nest1.ww3. This file needs to be saved for use in the small scale model.

7) To include the nesting data in the small scale model, the above boundary data file needs to be renamed to nest.ww3 and needs to be put in the directory from which ww3_shel for the small scale model is run. If the small scale model has properly defined the input boundary points in its definition file mod_def.ww3, it will automatically process the file nest.ww3 and update the boundary data as available. At this point, two additional tests are recommended.

   - When first running the small scale model with the file nest.ww3 present, pay close attention to the output of `ww3_shel` to assure that (i) the program reports that the file nest.ww3 has been processed and has been found OK, and (ii) that no additional warnings are present regarding incompatible or missing

boundary data. Also check the log file log.ww3 to assure that the boundary data are updated at the expected times.

- When all data apparently are processed, it is illustrative and prudent to make a model run of the small scale model where the wind fields are switched off in ww3_shel.inp, and where no restart file restart.ww3 is made available. In such a model run, wave energy can only enter the domain from the boundaries. This is a good test to assure that the boundary data is passed from the large scale model to the small scale model as expected.

Additional nested models can be added in the same way. Adding a second level nest from the small scale model is also done in the same way. The model is presently set up for producing up to 9 files with boundary data per model run. There are no limitations on the number of consecutive ('telescoping') nests.

B.2 Using ww3_bound and/or unstructured grids

In some circumstances it is difficult or impossible to know in advance the position of the forcing points for small scale model when running the large scale model. This is the case if one wants to run a coastal zoom using boundary condition from an on-line or third-party database.

In this case, it is possible to generate nest.ww3 file from spectral output using `ww3_bound`. This is particularly handy also for unstructured grids due to the irregular spacing of points on the boundary. `ww3_bound` takes a list of spectra files, which should have the same spectral grid, and generates a nest.ww3. The interpolation coefficients are determined from the positions of the nearest available spectra and the positions of the active boundary points in the small scale model.

B.3 Using ww3_multi

Performing two-way nesting in the wave model driver ww3_multi is greatly simplified compared to using the wave model driver ww3_shel, because all data transfer needed is performed internally in the multi-grid wave model routines. A mosaic model system is set up by iteratively going through the following steps.

1) Set up a grid using the ww3_grid utility. Define the grid, its active boundary points and all other model information such as time steps, but *do not* attempt to generate output nesting data for other grids. This will be assessed automatically by the multi-grid wave model routines in ww3_multi. Note that the lowest ranked grid can optionally use active boundary data, either as read from file or to be kept constant during computation. Higher ranked grids will require active boundary point in order to be valid in the mosaic approach,

2) Add this grid as an extra grid to the input file ww3_multi.inp with the appropriate rank number. Running ww3_multi will identify discrepancies between grids and requested boundary data points that can be resolved iteratively, and other discrepancies between grids. It can be tedious to remove such discrepancies by hand. The grid generation package of Chawla and Tolman (2007, 2008) checks for such discrepancies automatically, and is therefore recommended for grid generation for this version of WAVEWATCH III.

Note that grid on which input data fields are defined can be added in a similar way. Note that the use of land-sea masks in oceanic input fields (current, water level and ice) is recommended to assure realistic input values at coastal points.

Generally, lower ranked grids are developed first, although grid of any rank could be added at any time.

C Setting up for distributed machines (MPI)

C.1 Model setup

In order to run WAVEWATCH III on a distributed memory machine using MPI, two requirements need to be met. First, all executables need to be compiled properly. This implies that the codes are compiled with the proper WAVEWATCH III options (switches), and with the proper compiler options. Second, the parallel version of the model needs to be run in a proper parallel environment. This implies that the parallel codes are run on a multi-processor machine, invoking the proper parallel environment on that machine. These two issues are discussed in some detail below.

Of all the WAVEWATCH III programs described in section 4, only three benefit from a parallel implementation with MPI: the actual models ww3_shel and ww3_multi, and the initial conditions program ww3_strt. ww3_strt is typically not used in operational environments, and can generally be run in single processor mode. The main reason for running ww3_strt in multi-processor mode is to reduce its memory requirements. These three codes are the only codes that manipulate all spectra for all grid points simultaneously, and hence require much more memory than all other WAVEWATCH III programs. An added benefit (other than reduced run times) of running these programs in parallel is that the parallel versions of these programs require less memory per processor if the number of processors is increased.

Considering the above, it is sufficient for most implementations on parallel machines to compile only the main programs ww3_shel and ww3_multi with the MPI options. All other WAVEWATCH III programs with the exception of ww3_strt are designed for single-processor use. The latter programs should not be run in a parallel environment, because this will lead to I/O errors in output files. Furthermore, there is no possible gain in run time for these codes in a parallel environment due to their design. Because all programs share subroutines, it is important to assure that this compilation is done correctly, that is, that the subroutines and main programs are compiled with compatible compiler settings. This implies that subroutines that are shared between parallel and non-parallel programs should be compiled individually

for each application.

The first step for compiling the MPI version of programs is to assure that the proper compiler and compiler options are used. Examples of this for an IBM system using the xlf compiler, and a Linux system using the Portland compiler can be found in the example comp and link scripts provided with the distribution of WAVEWATCH III.

The second step is to invoke the proper compile options (switches) in compiling all parts of WAVEWATCH III. Most programs will be compiled for single-processor use. To assure that all subroutines are consistent with the main programs to which they are linked, the compile procedure should be divided into two parts. A simple script that will properly compile all WAVEWATCH III programs is given in Fig. C.1. An expanded version of this example is now available as

```
make_MPI
```

or in

```
w3_automake
```

. Alternatively, the commands in the script can be run interactively, while directly editing the switch file when appropriate.

An alternative way of consistently compiling the code is to first extract all necessary subroutines per code using w3_source, then put the sources and the makefile in individual directories, and compile using the make command. In this case the code for ww3_shel and ww3_multi are extracted using the appropriate MPI switches, whereas all other codes are extracted using the switches for the shared memory architecture.

After all codes have been compiled properly, the actual wave models ww3_shell and `ww3_multi` needs to be run in the proper parallel environment. The actual parallel environment depends largely on the computer system used. For instance, on NCEP's IBM systems, the number of processors and the proper environment is set in 'job cards' at the beginning of the script. The code is then directed to the parallel environment by invoking it as

```
poe ww3_shel
```

Conversely, on many Linux types systems, the MPI implementation includes the mpirun command which is typically used in the form

```
#!/bin/sh

# Generate appropriate switch file for shared and
# distributed computational environments

  cp switch switch.hold
  sed -e 's/DIST/SHRD/g' \
      -e 's/MPI //g'      switch.hold > switch.shrd
  sed 's/SHRD/DIST MPI/g' switch.hold > switch.MPI

# Make all single processor codes

  cp switch.shrd switch
  w3_make ww3_grid ww3_strt ww3_prep ww3_outf ww3_outp \
          ww3_trck ww3_grib gx_outf gx_outp

# Make all parallel codes

  cp switch.MPI switch
  w3_make ww3_shel ww3_multi

# Go back to a selected switch file

  cp switch.shrd switch
# cp switch.hold switch

# Clean up

  rm -f switch.hold switch.shrd switch.MPI
  w3_clean

# end of script
```

Figure C.1: Simple script to assure proper compilation of all WAVEWATCH III codes in a distributed (MPI) environment. This script assumes that the SHRD switch is selected in the switch file before the script is run.

```
mpirun -np $NP ww3_shel
```

where the `-np $NP` option typically requests a number of processes from a resource file (`$NP` is a shell script variable with a numerical value). For details of running parallel codes on your system, please refer to the manual or user support (if available).

Note that the as a part of the parallel model setup, I/O options are available to select between parallel and non-parallel file systems (see also Tolman, 2003a).

C.2 Common errors

Some of the most common errors made in attempting to run ww3_shel and ww3_multi under MPI are:

- Running in a parallel environment with a serial code (no MPI in compilation).

  This will result in corrupted data files, because all processes are attempting to write to the same file. This can be identified by the standard output of ww3_shel. The proper parallel version of the code will produce each output line only once. The non-parallel version will produce one copy of each output line for each individual process started.

- You are running in a parallel environment with a serial code (programs other than intended MPI codes).

  This will result in corrupted data files, because all processes are attempting to write to the same file. This can be identified by the standard output of the programs, which will produce multiple copies of each output line.

- ww3_shel or ww3_multi are compiled properly, but not run in a parallel environment.

  On some systems, this will result in automatic failure of the execution of `ww3_shel`. If this does not occur, this can only be traced by using system tools for tracking when and where the code is running.

- During compilation serial and parallel compiled subroutines are mixed.

  This is the most common source of compiling, linking and run time errors of the code. Follow the steps outlined in the previous section to avoid this.

C.3 MPI point-to-point communication errors

Running ww3_multi in parallel with several large overlapping grids involves a large number of concurrently active MPI point-to-point communications (MPI send/recv pairs). For correct execution, each active MPI message must have a unique envelope (send id, recv id, tag, communicator) with an allowed tag value. In this context two types of MPI point-to-point communication errors may occur: (1) the MPI message tag value exceeds an upper-bound or (2) two or more MPI messages have the same envelope. The first error may result in ww3_multi crashing with a MPI "invalid tag" error or an internal tag upper-bound exceeded error. The second error may result in spectra sent from one MPI task to another being delivered to the wrong location. The second error is more difficult to detect in that it is not trapped by MPI and may only be manifested as strange results in model output.

To address these possible errors the allowed ranges of MPI tags for the different sets of point-to-point communication in ww3_multi are controlled by the $MTAGB$, $MTAG0$, $MTAG1$, $MTAG2$, and $MTAG\_UB$ parameters defined in WMMDATMD. These parameters must satisfy $MTAGB \geq 0$ and $7 * NRGRD - 1 \leq MTAG0 < MTAG1 < MTAG2 < MTAG\_UB \leq MPI\_TAG\_UB$, where MPI_TAG_UB is the tag upper-bound for the MPI implementation.

The value of MPI_TAG_UB for a specific MPI implementation can be obtained at run-time using the $MPI\_COMM\_GET\_ATTR$ routine. An MPI implementation is free to set the value of MPI_TAG_UB larger than the minimum set by the MPI standard ($32767 = 2^{15} - 1$). In the current release version of OpenMPI, the value of MPI_TAG_UB is 2147483647 ($2^{31} - 1$). On the Cray XC40 with Cray MPICH, the value of MPI_TAG_UB is much smaller, that is, 2097151 ($2^{21} - 1$). As the currently known lowest value of MPI_TAG_UB amongst available parallel platforms, the Cray XC40 value is used to set $MTAG\_UB$ in WMMDATMD.

If an MPI tag value exceeds the upper-bound (MPI_TAG_UB) imposed by the MPI implementation, ww3_multi may crash with a MPI "invalid tag" error. If an MPI tag value exceeds one of the internal tag upper-bounds, ww3_multi will crash with error code 1001 and a report of which tag upper-bound was exceeded. What follows is a description of the allowed MPI tag ranges and how they are set.

The $MTAGB$ parameter is only used as the tag lower-bound for blocking communication that does not overlap with other point-to-point communication. Hence it is sufficent to set $MTAGB$ to the lowest allowed MPI tag value of 0.

In addition to being the tag lower-bound for communication of internal boundary data in WMINIOMD, the $MTAG0$ parameter is used in WMIOPOMD as the tag upper-bound for the unified point output communication. To ensure that the unified point output communication tag values are ≥ 0, $MTAG0$ must be at least $7 * NRGRD - 1$. A generous setting of $MTAG0 = 1000$ is used in WMMDATMD.

The allowed tag range for point-to-point communication of internal boundary data (WMINIOMD:WMIOBS) is $(MTAG0, MTAG1]$. Given that this communication involves only the boundary points of the model grids a generous setting of $MTAG1 = 10000$ is used in WMMDATMD.

The allowed tag range for point-to-point communication from high rank to low rank grids (WMINIOMD:WMIOHS) is $(MTAG1, MTAG2]$. The allowed tag range for point-to-point communication between equal rank model grids (WMINIOMD:WMIOES) is $(MTAG2, MTAG\_UB]$. The high-rank-to-low-rank and equal-rank communications involve both boundary and interior points of model grids. Hence, the allowed tag ranges for these two communication sets should be larger than the allowed range for the communication of internal boundary data. In nested grid applications (e.g., a single global grid with several regional nests) the required tag range for high-rank-to-low-rank communications will be larger than the required tag range for the equal-rank communications. The setting of $MTAG2 = 1500000$ is used in WMMDATMD to give a larger portion of the total allowed range of tags for the high-rank-to-low-rank communications. Other multiple grid applications may require adjusting $MTAG2$.

D Mosaic approach with non-regular grids

D.1 Introduction

WAVEWATCH III version 3.14 (Tolman, 2009b) introduced multi-grid capability. This capability is described above (Section 3.14.2). With model version 4, there is an option to use irregular grids or unstructured grids, as described in Section 3.4.3 and Section 3.4.4, respectively. Unfortunately, the methods described in Section 3.14.2 are not general, as they are intended for regular grids only. Some new capabilities are implemented in 6.07 to accommodate irregular and unstructured grids within the multi-grid approach.

The core component for communication from lower rank grids to higher rank grids of Tolman (2008b) is an interpolation in space to provide boundary data at the higher spatial resolution. For version 6.07 the technique was generalized by making calls to the grid-search-utility (GSU) implemented in WAVEWATCH III version 4 by T. Campbell. Other generalizations were made to ancillary components of this routine.

The core component for communication from higher rank grids to lower rank grids of Tolman (2008b) is a conservative remapping operation: the spectral density of a larger (low rank) grid cell is updated based on the spectral densities of the overlapping smaller (high rank) grid cells, weighted according to the fraction of the larger cell that is covered by each smaller cell, keeping in mind that a smaller cell may be overlapping with more than one larger cell. For version 6.07 the technique was generalized by making calls to an external software package, SCRIP-WW3, which is described below. The remapping weights are stored in a FORTRAN "derived data type" array. Generalizations were also made to ancillary components of the remapping routine, for example to the logic used to calculate distances to the boundaries, to deal with masked points and land points, etc.

D.2 SCRIP-WW3

The SCRIP-WW3 software package is adapted from the SCRIP (Spherical Coordinate Remapping and Interpolation Package) software package of

Jones (1998), which we refer to here as SCRIP-LANL. SCRIP-WW3 is based on SCRIP-LANL v1.5. The primary difference between SCRIP-LANL and SCRIP-WW3 is that the former is a standalone code using NetCDF files for user interface, and the latter is modified to run within WAVEWATCH III with communication via system memory. Further, SCRIP-WW3 only utilizes the conservative remapping feature, whereas SCRIP-LANL has a number of other optional uses, such as bi-linear remapping.

The conservative remapping used in SCRIP is based on Jones (1999). In this method, for each source/destination grid pair, line integrals are computed around all cells in each grid while keeping track of intersections with the other grid, resulting in area of overlap between grids. The method is designed for use with a spherical coordinate system (as opposed to treating latitudes and longitudes as if they are x- any y-axes in a Cartesian system) and includes special logic for handling longitude wrapping (the so-called "branch cut") and cells that include a pole. It also allows for unstructured grids, with arbitrary number of cell corners. The grid corner coordinates must be given in an order which traces the outside of a grid cell in a counterclockwise direction. The software allows either first- or second-order remapping; weights for both are calculated in SCRIP-WW3. At present, only the first-order remapping is implemented in WAVEWATCH III : Jones (1999) points out that there is virtually no advantage to using the second-order method when mapping from a fine grid to a coarse grid.

D.3 SCRIP Operation

SCRIP-WW3 is activated by including SCRIP in the file switch. If the user attempts to use irregular or unstructured grids within ww3_multi without this switch, this will result in an error message and program termination. SCRIP-WW3 is not required for ww3_shel (traditional one-way nesting), and is not required for ww3_multi with only regular grids, since original methods for remapping are retained in the code for this purpose. SCRIP-WW3 source files are kept in a separate directory /ftn/SCRIP/, since it is modified 3rd party software. With the SCRIP switch, the build system (ww3_make) will automatically compile files from this directory and link them into ww3_multi.

A user may also optionally include the switch SCRIPNC along with SCRIP. This feature requires NetCDF. Instructions for using NetCDF in

WAVEWATCH III are found in Section 5.7 and in the file w3_make. With SCRIPNC activated, for each source/destination grid pair, a NetCDF file will be created, e.g. rmp_src_to_dst_conserv_002_001.nc, with 002 and 001 referring to the source and destination grid respectively; the numbering of grids is assigned by ww3_multi and is indicated in screen output of that program. This .nc file contains all information required by WAVEWATCH III for remapping. Additional diagnostic information about the remapping can be included in the .nc file by adding the switch T38. Note: switch should include either 'SCRIP SCRIPNC' or 'SCRIP'; using SCRIPNC without SCRIP will result in a compile error.

Though it is not required, SCRIP-WW3 may be utilized for remapping between regular grids. In the case of spherical (lat/lon) grids, there may be slight differences using SCRIP-WW3, since SCRIP-WW3 calculates areas based on real distances, and the non-SCRIP approach uses degrees lat/lon.

D.4 Optimization and common problems

SCRIP-WW3 routines are not parallelized. Therefore, if ww3_multi is run with many processes, each process will perform identical calculations of all weights. For remapping between grids with large numbers of points, this can make the preparations for ww3_multi time-consuming, e.g. 3 to 10 minutes, which can be prohibitively expensive for routine, operational use. To deal with this problem, SCRIP-WW3 has been adapted to allow use of remapping weights that were computed in a prior application of ww3_multi. If the appropriate .nc files are found by ww3_multi, it will simply read the remapping data from these files, and SCRIP will not be called. Of course, if any grids have been changed since the prior run, or if moving grids are used, pre-computed weights should not be used.

An additional feature is provided for user convenience: if a file named SCRIP_STOP is found in the run directory, ww3_multi will terminate after the .nc files are created. The content of SCRIP_STOP is unimportant; it may be an empty file. When this feature is used, remapping operations will be distributed among processes: rmp_src_to_dst_conserv_002_001.nc is created by process 1, rmp_src_to_dst_conserv_003_001.nc is created by process 2, etc., which will dramatically improve performance in cases where a significant number of grids are used. To clarify, there are two modes of operation

that are targetted with this feature: Mode A) Precalculate weights, where SCRIP_STOP exists and .nc files do not exist. Mode B) Use precalculated weights, where SCRIP_STOP does not exist and .nc files do exist. If both files types exist (through accident) in the work directory, ww3_multi will fail with an error. In a hypothetical operational context, Mode A is used for the first run and Mode B is used for all subsequent runs with the same grid set. The scalability is limited by the most expensive remapping pair, i.e. load balancing is an issue. For a case where 12 remapping pairs are calculated and each pair requires 1/12th of the computation time, speed-up will be by factor twelve. For another case with 12 remapping pairs, where one remapping pair takes 50% of computation time, speed-up will be by factor two only. Note that resources are maximized by using a number of processes equal to the number of remapping pairs: extra processes will not be used.

To further explain the options available to users, take an example of a multi-grid system with 9 grids and 12 remapping pairs, with many sea points, run twice a day for several months, for a total of 1000 forecasts. The user may handle this in different ways:

1) Using SCRIP, SCRIPNC, and MPI, and using the SCRIP_STOP feature, the calculation of weights will be done in parallel. The first time the model is applied, this may take 5 to 10 minutes to calculate remapping weights (Mode A above) and 20 minutes to perform the model forecast (Mode B above). For forecasts 2 to 1000, only the 20 minutes to perform the model forecast (Mode B) is needed.
2) Using SCRIP, SCRIPNC and the SCRIP_STOP feature, but creating the .nc files running in serial mode and running the forecast with MPI, the first time the model is applied, this may take 20 minutes to calculate remapping weights and 20 minutes to perform the model forecast. For forecasts 2 to 1000, only the 20 minutes to perform the model forecast is needed.
3) Using SCRIP, SCRIPNC, and MPI, without using the SCRIP_STOP feature, the calculation of weights will not be done in parallel, and will even be slower than if run in serial, because of communications. The first time the model is applied, this may take 40 minutes to calculate remapping weights and 20 minutes to perform the model forecast. For forecasts 2 to 1000, only the 20 minutes to perform the model forecast is needed.

4) Using SCRIP and MPI, running on 12 processors, the calculation of weights will not be done in parallel, will be slow, and will need to be computed each time. For all forecasts 1 to 1000, this may take 40 minutes to calculate remapping weights and 20 minutes to perform the model forecast.

In some cases, SCRIP-WW3 will return suspicious values for some points, which will result in warning message(s) in the screen output. When this occurs for a small fraction of grid points, our experience (from analysis of the diagnostic output in the .nc files) is that the remapping weights are valid, since the problem points are at edges where WAVEWATCH III does not use the weights. However, when this occurs for a large fraction of grid points, it is likely that SCRIP-WW3 has truly failed. In this case, WAVEWATCH III stops with an error message. Our experience is that this occurs most often for overlapping regular grids with a large number of coincident line segments. A workaround exists: it can be remedied by adding an artificial offset to one of the grids. It was already possible to specify an offset in ww3_grid.inp in the grid description, but since that offset is intended as a real quantity, this other, artificial offset is implemented separately as a namelist option. It is `GSHIFT` under namelist group MISC. An example namelist would be: `MISC GSHIFT = 1.0D-6`. A smaller number will result in less regridding error, though the number must be sufficient large to actually have the intended beneficial effect. We recommend to determine this by trial-and-error, varying by factor 10 each time.

D.5 Limitations

Two features are not yet addressed, and will be addressed in a later version:

1) Communication between equal rank grids is still limited to regular grids. If one of the grids is irregular or unstructured, ww3_multi will terminate with an error message. It is possible to have non-regular grids as part of a multi-grid system which includes equal rank grids, as long as the overlapping equal-ranked grids are all regular.
2) The "input grid" (or "F modid") option for defining input fields (e.g. winds) is not implemented yet for irregular or unstructured

grids. If this is attempted, ww3_multi will terminate with an error message. The "native" input grid option should be used instead.

Attribution statement: This section was written by E. Rogers. The coding and testing for this effort was performed by E. Rogers, M. Dutour, A. Roland, F. Ardhuin, and K. Lind. Technical advice was given by H. Tolman and T. Campbell.

E Ocean-Ice-Waves-Atmosphere coupling with OASIS

E.1 Introduction

WAVEWATCH III has been interfaced with OASIS3-MCT to allow coupling simulations with atmosphere and/or ocean models. OASIS (Ocean Atmosphere Sea Ice Soil)[1] – note that waves are missing in that acronym – is a coupling software developed by the CERFACS and CNRS (Valcke, 2013). The current OASIS3-MCT version is interfaced with MCT, the Model Coupling Toolkit (J. Larson, 2005; R. Jacob, 2005). developed by the Argonne National Laboratory. The OASIS coupler is also interfaced with the SCRIP library developed by Los Alamos National Laboratory. All the information on how to use the OASIS coupler is present in the oasis user guide. Here we will just add the information about the use of OASIS in WAVEWATCH III.

In a nutshell, OASIS3-MCT ...

- ... is totally parallelized
- ... doesn't have an executable (we don't need to give it processes when we launch a coupling simulation)
- ... is able to exchange 2D and 3D fields
- ... is able to exchange fields in parallel
- ... support unstructured grids
- ... uses an input file called *namcouple* that allows changes to the coupling characteristics (exchange time, interpolation type, number of exchange fields) without recompiling the code...

[1] https://verc.enes.org/oasis/

E.2 Interfacing with OASIS3-MCT

To communicate with another model, a component model (ocean, wave, sea ice or atmosphere) needs to include a few specific calls to the OASIS3-MCT coupling library. To use these OASIS's functions in WAVEWATCH III we created 4 new modules:

- w3oacpmd.ftn, module containing common functions for atmosphere and ocean coupling. These functions are called before the temporal loop, in the `ww3_shel` program.
- w3ogcmmd.ftn, w3igcmmd.ftn and w3agcmmd.ftn, modules containing specific functions for, respectively, waves-ocean coupling, waves-sea ice-coupling and waves-atmosphere coupling. These functions are called in the temporal loop.

E.3 Compiling with OASIS3-MCT

To use or not these coupling functions 5 switches were created:

- switch `COU`, to perform the coupling : reading the ww3_shel.inp input file and define the number of variable exchanged and the time exchange.
- switch `OASIS`, to initialize the coupler OASIS w3oacpmd.ftn
- switch `OASACM` and/or `OASOCM` and/or `OASICM`, to send/receive the coupling fields to/from the atmospheric model w3agcmmd.ftn and/or the oceanic model w3ogcmmd.ftn and/or the sea ice model w3igcmmd.ftn

To allow the use of OASIS, WAVEWATCH III ® should be compiled with the following switches. For a coupling with an atmospheric circulation model

```
COU OASIS OASACM
```

and with an ocean circulation model

```
COU OASIS OASOCM
```

and with a sea ice model

```
COU OASIS OASICM
```

and both ocean and atmosphere circulation models

```
COU OASIS OASACM OASOCM
```

Only the program `ww3_shel` is compiled with the OASIS library, all the other programs can be used as usual.

The switches for interpolation in time of the wind and current forcing fields must not be used regarding the fact that coupling mechanisms cannot provided the future value of the forcing field. Depending on the type of coupling, the switch `WNT0` must be set for atmospheric coupling and the switch `CRT0` must be set for oceanic coupling.

E.4 Launch a coupling simulation

To launch a coupling simulation, for example with Intel Mpi, we need the:

- input files for WW3 : ww3_shel.inp, and the usual *.ww3 files.
- input file for OASIS : namcouple
- input files for ocean/sea ice/atmosphere models : files depends on the model

To launch a coupling simulation, the mpirun command should be used as follows

```
mpirun -np $nbre_cores_WW3 exe_WW3 :  -np $nbre_cores_OA
exe_OA
```

E.5 Limitations

A few limitations are not yet addressed, and will be addressed in a later version:

- the coupling with OASIS is only coding for ww3_shel program, not yet for the `ww3_multi`
- in the WW3 suite, there are 2 versions of SCRIP, one for OASIS and one for the `ww3_multi`
- ...

F Coupling with NUOPC

F.1 Introduction

WAVEWATCH III as of v6.02 has a component cap `wmesmf` that interfaces with the multi-grid routines via the National Unified Operational Prediction Capability (NUOPC)[1] Layer which specifies the use of the Earth System Modeling Framework (ESMF) [2] for coupling with other earth systems such as atmosphere, ocean, ice and storm surge models. This cap is meant to be flexible and is already in use in multiple different coupled models at NOAA and the US Navy. This section describes how to build, install, configure and run the NUOPC cap. It assumes a basic knowledge of WAVEWATCH III.

F.2 Building and Installing the NUOPC Cap

To make the library that will contain the WAVEWATCH III code and cap that can be included in a NUOPC coupled model, use `modelesmfMakefile`. For example, in the NOAA Environmental Modeling System (NEMS) the command is:

```
make ww3_nems
```

this makefile will subsequently call `w3_make`. As part of this process a nuopc.mk makefile fragment will also be created, which tells NUOPC/ESMF where the WAVEWATCH III library is located.

F.3 Import/Export Fields in the NUOPC Cap

The avaiable fields for import and export are listed below. Please see Section F.4 for information on how to activate coupling for an import field.
Import Fields:

[1] https://earthsystemcog.org/projects/nuopc/
[2] https://www.earthsystemcog.org/projects/esmf/

- sea_surface_height_above_sea_level
- surface_eastward_sea_water_velocity
- surface_northward_sea_water_velocity
- eastward_wind_at_10m_height
- northward_wind_at_10m_height
- sea_ice_concentration

Export:

- wave_induced_charnock_parameter
- wave_z0_roughness_length
- eastward_stokes_drift_current
- northward_stokes_drift_current
- eastward_wave_bottom_current
- northward_wave_bottom_current
- wave_bottom_current_period
- eastward_wave_radiation_stress
- eastward_northward_wave_radiation_stress
- northward_wave_radiation_stress

F.4 Configuration of Input Files for the NUOPC Cap

The required WAVEWATCH III input file is the ww3_multi.inp or ww3_multi.nml file. To spcify that a particular input field is to be obtained via coupling and not a file 'CPL:' is put infront of the input grid specification for the particular input field.

Note that current limitations of the NUOPC cap are that there can only be one input grid (whether it is one of the model grids or an input grid) and one export grid (the first computational grid if there are multiple computational grids). The grid can be unstructured or structured.

Note that while the start and end times for the run is determined by the NUOPC driver, the start time, end time and frequency of output are still determined by the ww3_multi input file.

F.5 Running the NUOPC Cap

While the cap is designed to be used in coupled systems outside of the scope of this documentation. A script to run a regression test of the standalone WAVEWATCH III cap is provided in the `regtestrun_esmf_test_suite` script.

G Configuration of Input Files

This section contains the input files in the traditional *.inp format or new namelist *.nml format for each program. These can also be found in the model/inp or model/nml folders respectively.

G.1 ww3_grid

G.1.1 ww3_grid.inp

start of example input file (traditional form)

```
$ ------------------------------------------------------------------- $
$ WAVEWATCH III Grid preprocessor input file                           $
$ ------------------------------------------------------------------- $
$ Grid name (C*30, in quotes)
$
  'TEST GRID (GULF OF NOWHERE)   '
$
$ Frequency increment factor and first frequency (Hz) ---------------- $
$ number of frequencies (wavenumbers) and directions, relative offset
$ of first direction in terms of the directional increment [-0.5,0.5].
$ In versions 1.18 and 2.22 of the model this value was by definiton 0,
$ it is added to mitigate the GSE for a first order scheme. Note that
$ this factor is IGNORED in the print plots in ww3_outp.
$
   1.1  0.04118  25  24  0.
$
$ Set model flags ---------------------------------------------------- $
$  - FLDRY         Dry run (input/output only, no calculation).
$  - FLCX, FLCY    Activate X and Y component of propagation.
$  - FLCTH, FLCK   Activate direction and wavenumber shifts.
$  - FLSOU         Activate source terms.
$
   F T T T F T
$
$ Set time steps ----------------------------------------------------- $
$ - Time step information (this information is always read)
$     maximum global time step, maximum CFL time step for x-y and
$     k-theta, minimum source term time step (all in seconds).
$
```

```
    900. 950. 900. 300.
$
$ Start of namelist input section ---------------------------------- $
$   Starting with WAVEWATCH III version 2.00, the tunable parameters
$   for source terms, propagation schemes, and numerics are read using
$   namelists. Any namelist found in the folowing sections up to the
$   end-of-section identifier string (see below) is temporarily written
$   to ww3_grid.scratch, and read from there if necessary. Namelists
$   not needed for the given switch settings will be skipped
$   automatically, and the order of the namelists is immaterial.
$   As an example, namelist input to change SWELLF and ZWND in the
$   Tolman and Chalikov input would be
$
$   &SIN2 SWELLF = 0.1, ZWND = 15. /
$
$ Define constants in source terms --------------------------------- $
$
$ Stresses - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
$   TC 1996 with cap    : Namelist FLX3
$                           CDMAX  : Maximum allowed CD (cap)
$                           CTYPE  : Cap type :
$                                     0: Discontinuous (default).
$                                     1: Hyperbolic tangent.
$   Hwang 2011          : Namelist FLX4
$                           CDFAC  : re-scaling of drag
$
$ Linear input - - - - - - - - - - - - - - - - - - - - - - - - - - -
$   Cavaleri and M-R    : Namelist SLN1
$                           CLIN   : Proportionality constant.
$                           RFPM   : Factor for fPM in filter.
$                           RFHF   : Factor for fh in filter.
$
$ Exponential input  - - - - - - - - - - - - - - - - - - - - - - - -
$   WAM-3               : Namelist SIN1
$                           CINP   : Proportionality constant.
$
$   Tolman and Chalikov : Namelist SIN2
$                           ZWND   : Height of wind (m).
$                           SWELLF : swell factor in (n.nn).
$                           STABSH, STABOF, CNEG, CPOS, FNEG :
$                                    c0, ST0, c1, c2 and f1 in . (n.nn)
$                                    through (2.65) for definition of
$                                    effective wind speed (!/STAB2).
$   WAM4 and variants   : Namelist SIN3
$                           ZWND    : Height of wind (m).
```

```
$                          ALPHA0  : minimum value of Charnock coeffici
$                          ZOMAX   : maximum value of air-side roughnes
$                          BETAMAX : maximum value of wind-wave couplin
$                          SINTHP  : power of cosine in wind input
$                          ZALP    : wave age shift to account for gust
$                       TAUWSHELTER : sheltering of short waves to reduc
$                         SWELLFPAR : choice of swell attenuation formul
$                                     (1: TC 1996, 3: ACC 2008)
$                           SWELLF : swell attenuation factor
$     Extra parameters for SWELLFPAR=3 only
$                   SWELLF2, SWELLF3 : swell attenuation factors
$                          SWELLF4 : Threshold Reynolds number for ACC2
$                          SWELLF5 : Relative viscous decay below thres
$                            ZORAT : roughness for oscil. flow / mean f
$   BYDRZ input          : Namelist SIN6
$                          SINA0    : factor for negative input
$                          SINWS    : wind speed scaling option
$                          SINFC    : high-frequency extent of the
$                                     prognostic frequency region
$
$ Nonlinear interactions - - - - - - - - - - - - - - - - - - - - - - -
$   Discrete I.A.        : Namelist SNL1
$                          LAMBDA : Lambda in source term.
$                          NLPROP : C in sourc term. NOTE : default
$                                   value depends on other source
$                                   terms selected.
$                          KDCONV : Factor before kd in Eq. (n.nn).
$                          KDMIN, SNLCS1, SNLCS2, SNLCS3 :
$                                   Minimum kd, and constants c1-3
$                                   in depth scaling function.
$   Exact interactions   : Namelist SNL2
$                          IQTYPE : Type of depth treatment
$                                    1 : Deep water
$                                    2 : Deep water / WAM scaling
$                                    3 : Shallow water
$                          TAILNL : Parametric tail power.
$                          NDEPTH : Number of depths in for which
$                                   integration space is established.
$                                   Used for IQTYPE = 3 only
$                        Namelist ANL2
$                          DEPTHS : Array with depths for NDEPTH = 3
$   Gen. Multiple DIA    : Namelist SNL3
$                          NQDEF  : Number of quadruplets.
$                          MSC    : Scaling constant 'm'.
$                          NSC    : Scaling constant 'N'.
```

```
$                           KDFD   : Deep water relative filter depth,
$                           KDFS   : Shallow water relative filter depth
$                         Namelist ANL3
$                           QPARMS : 5 x NQDEF paramaters describing the
$                                    quadruplets, repeating LAMBDA, MU,
$                                    Cdeep and Cshal. See examples below
$   Two Scale Approx.   : Namelist SNL4
$                           INDTSA : Index for TSA/FBI computations
$                                    (0 = FBI ; 1 = TSA)
$                           ALTLP  : Index for alternate looping
$                                    (1 = no ; 2 = yes)
$
$ Traditional DIA setup (default):
$
$ &SNL3 NQDEF =  1, MSC =  0.00,  NSC = -3.50 /
$ &ANL3 QPARMS = 0.250, 0.000,  -1.0, 0.1000E+08, 0.0000E+00 /
$
$ GMD3 from 2010 report (G13d in later paper) :
$
$ &SNL3 NQDEF =  3, MSC =  0.00,  NSC = -3.50 /
$ &ANL3 QPARMS = 0.126, 0.000,  -1.0, 0.4790E+08, 0.0000E+00 ,
$                0.237, 0.000,  -1.0, 0.2200E+08, 0.0000E+00 ,
$                0.319, 0.000,  -1.0, 0.1110E+08, 0.0000E+00 /
$
$ G35d from 2010 report:
$
$ &SNL3 NQDEF =  5, MSC =  0.00,  NSC = -3.50 /
$ &ANL3 QPARMS = 0.066, 0.018,  21.4, 0.170E+09, 0.000E+00 ,
$                0.127, 0.069,  19.6, 0.127E+09, 0.000E+00 ,
$                0.228, 0.065,   2.0, 0.443E+08, 0.000E+00 ,
$                0.295, 0.196,  40.5, 0.210E+08, 0.000E+00 ,
$                0.369, 0.226,  11.5, 0.118E+08, 0.000E+00 /
$
$ Nonlinear filter based on DIA  - - - - - - - - - - - - - - - - - - -
$                         Namelist SNLS
$                           A34    : Relative offset in quadruplet
$                           FHFC   : Proportionality constants.
$                           DMN    : Maximum relative change.
$                           FC1-3  : Constants in frequency filter.
$
$ Whitecapping dissipation   - - - - - - - - - - - - - - - - - - - - -
$   WAM-3               : Namelist SDS1
$                           CDIS, APM : As in source term.
$
$   Tolman and Chalikov : Namelist SDS2
```

```
$                           SDSA0, SDSA1, SDSA2, SDSB0, SDSB1, PHIMIN :
$                                    Constants a0, a1, a2, b0, b1 and
$                                    PHImin.
$
$   WAM4 and variants   : Namelist SDS3
$                           SDSC1    : WAM4 Cds coeffient
$                           MNMEANP, WNMEANPTAIL : power of wavenumber
$                                    for mean definitions in Sds and tai
$                           SDSDELTA1, SDSDELTA2 : relative weights
$                                    of k and k^2 parts of WAM4 dissipat
$                           SDSLF, SDSHF : coefficient for activation of
$                              WAM4 dissipation for unsaturated (SDSLF)
$                               saturated (SDSHF) parts of the spectrum
$                           SDSC2    : Saturation dissipation coefficien
$                           SDSC4    : Value of B0=B/Br for wich Sds is
$                           SDSBR    : Threshold Br for saturation
$                           SDSP     : power of (B/Br-B0) in Sds
$                           SDSBR2   : Threshold Br2 for the separation
$                             WAM4 dissipation in saturated and non-satu
$                           SDSC5 : coefficient for turbulence dissipati
$                           SDSC6 : Weight for the istropic part of Sds_
$                           SDSDTH: Angular half-width for integration o
$
$   BYDRZ               : Namelist SDS6
$                           SDSET    : Select threshold normalization spe
$                           SDSA1, SDSA2, SDSP1, SDSP2  :
$                                Coefficients for dissipation terms T1 an
$                       : Namelist SWL6
$                           SWLB1    : Coefficient for swell dissipation
$
$ Bottom friction  - - - - - - - - - - - - - - - - - - - - - - - - - - -
$   JONSWAP             : Namelist SBT1
$                           GAMMA   : Bottom friction emprical constant
$
$
$ Surf breaking  - - - - - - - - - - - - - - - - - - - - - - - - - - - -
$   Battjes and Janssen : Namelist SDB1
$                           BJALFA  : Dissipation constant (default = 1)
$                           BJGAM   : Breaking threshold (default = 0.73
$                           BJFLAG  : TRUE  - Use Hmax/d ratio only (def
$                                     FALSE - Use Hmax/d in Miche formul
$
$ Dissipation in the ice   - - - - - - - - - - - - - - - - - - - - - - -
$   Generalization of Liu et al. : Namelist SIC2
$                           IC2DISPER  : If true uses Liu formulation wi
```

```
$                                       If false, uses the generalizati
$                                       to laminar transition
$                           IC2TURB    : empirical factor for the turbul
$                           IC2ROUGH   : under-ice roughness length
$                           IC2REYNOLDS: Re number for laminar to turbul
$                           IC2SMOOTH  : smoothing of transition reprens
$                           IC2VISC    : empirical factor for viscous pa
$
$
$ Scattering in the ice  & creep dissipations- - - - - - - - - - - - - -
$   Generalization of Wiliams et al. : Namelist SIS2
$                           ISC1          : scattering coefficient (defa
$                           IS2BACKSCAT   : fraction of energy back-scat
$                           IS2BREAK      : TRUE  - changes floe max dia
$                                         : FALSE - does not change floe
$                           IS2C1         : scattering in pack ice
$                           IS2C2         : frequency dependance of scat
$                           IS2C3         : frequency dependance of scat
$                           ISBACKSCAT    : fraction of scattered energy
$                           IS2DISP       : use of ice-specific dispersi
$                           FRAGILITY     : parameter between 0 and 1 th
$                           IS2DMIN       : minimum floe diameter in met
$                           IS2DAMP       : multiplicative coefficient f
$                           IS2UPDATE     : TRUE  - updates the max floe
$                                         : FALSE - updates the max floe
$
$ Dissipation by sea ice
$   Empirical/parametric representations : Namelist SIC4
$                           IC4METHOD    : integer 1 to 7
$                                        : In most cases, additional inp
$                                        :   is required.
$                                        : See examples in /regtests/ww3
$                                        : See also: 1) description in m
$                                        :   and 2) inline documentation
$                                            w3sic4md.ftn
$
$ Triad nonlinear interactions - - - - - - - - - - - - - - - - - - - - -
$   Lumped Triad Interaction (LTA) : Namelist STR1 (To be implemented)
$                           PTRIAD1 : Proportionality coefficient (defau
$                           PTRIAD2 : Multiple of Tm01 up to which inter
$                                     is computed (2.5)
$                           PTRIAD3 : Ursell upper limit for computing
$                                     interactions (not used, default 10
$                           PTRIAD4 : Shape parameter for biphase
$                                     computation (0.2)
```

```
$                          PTRIAD5 : Ursell number treshold for computi
$                                    interactions (0.01)
$
$ Shoreline reflections - - - - - - - - - - - - - - - - - - - - - - - -
$   ref. parameters       : Namelist REF1
$                          REFCOAST  : Reflection coefficient at shorel
$                          REFFREQ   : Activation of freq-dependent ref
$                          REFMAP    : Scale factor for bottom slope ma
$                          REFRMAX   : maximum ref. coeffient (default
$                          REFFREQPOW: power of frequency
$                          REFICEBERG: Reflection coefficient for icebe
$                          REFSUBGRID: Reflection coefficient for islan
$                          REFCOSP_STRAIGHT: power of cosine used for
$                                      straight shoreline
$
$ Bound 2nd order spectrum and free IG - - - - - - - - - - - - - - - - -
$   IG1 parameters       : Namelist SIG1
$                          IGMETHOD  : 1: Hasselmann, 2: Krasitskii-Jan
$                          IGADDOUTP : activation of bound wave correct
$                                      in ww3_outp / ww3_ounp
$                          IGSOURCE  : 1: uses bound waves, 2: empirica
$                          IGSTERMS  :  > 0 : no source term in IG band
$                          IGMAXFREQ : maximum frequency of IG band
$                          IGEMPIRICAL: constant in empirical free IG s
$                          IGBCOVERWRITE: T: Replaces IG spectrum, does
$                          IGSWELLMAX: T: activates free IG sources for
$
$
$ Propagation schemes ---------------------------------------------- $
$   First order          : Namelist PRO1
$                          CFLTM  : Maximum CFL number for refraction.
$
$   UQ/UNO with diffusion : Namelist PRO2
$                          CFLTM  : Maximum CFL number for refraction.
$                          DTIME  : Swell age (s) in garden sprinkler
$                                   correction. If 0., all diffusion
$                                   switched off. If small non-zero
$                                   (DEFAULT !!!) only wave growth
$                                   diffusion.
$                          LATMIN : Maximum latitude used in calc. of
$                                   strength of diffusion for prop.
$
$   UQ/UNO with averaging : Namelist PRO3
$                          CFLTM  : Maximum CFL number for refraction.
$                          WDTHCG : Tuning factor propag. direction.
```

```
$                          WDTHTH : Tuning factor normal direction.
$
$   Note that UQ and UNO schemes have no tunable parameters.
$   All tuneable parameters are associated with the refraction
$   limitation and the GSE alleviation.
$
$ Unstructured grids ---------------------------------------------- $
$   UNST parameters       : Namelist UNST
$                           UGOBCAUTO : TRUE: OBC points are taken from
$                                       FALSE: OBC points must be listed
$                           UGOBCDEPTH: Threshold ( < 0) depth for OBC p
$                           EXPFSN    : Activation of N scheme
$                           EXPFSPSI  : Activation of PSI scheme
$                           EXPFSFCT  : Activation of FCT scheme
$                           IMPFSN    : Activation of N implicit scheme
$                           IMPTOTAL  : Activation of fully implicit sch
$                           EXPTOTAL  : Turn on implicit refraction (onl
$                           IMPREFRACTION  : Turn on implicit freq. shif
$                           IMPFREQSHIFT  : Turn on implicit freq. shift
$                           IMPSOURCE  : Turn on implicit source terms (
$                           JGS_TERMINATE_MAXITER  : max. Number of iter
$                           JGS_TERMINATE_DIFFERENCE  : terminate based
$                           JGS_TERMINATE_NORM  : terminate based on the
$                           JGS_USE_JACOBI  : Use Jacobi solver for impt
$                           JGS_BLOCK_GAUSS_SEIDEL  : Use Block Gauss Se
$                           JGS_MAXITER  : max. Number of solver iterati
$                           JGS_PMIN  : % of grid points that do not nee
$                           JGS_DIFF_THR  : implicit solver threshold fo
$                           JGS_NORM_THR  : terminate based on the norm
$                           SETUP_APPLY_WLV  : Compute wave setup (exper
$                           SOLVERTHR_SETUP  : Solver threshold for setu
$                           CRIT_DEP_SETUP  : Critical depths for setup
$
$ SMC grid propagation    : Namelist PSMC and default values
$                           CFLTM  : Maximum CFL no. for propagation, 0.
$                           DTIME  : Swell age for diffusion term (s), 0
$                           LATMIN : Maximum latitude (deg) for GCT,   8
$                           RFMAXD : Maximum refraction turning (deg), 8
$                           LvSMC  : No. of refinement level, default 1
$                           ISHFT  : Shift number of i-index, default 0
$                           JEQT   : Shift number of j-index, default 0
$                           NBISMC : No. of input boundary points,    0
$                           UNO3   : Use 3rd order advection scheme, .FA
$                           AVERG  : Add extra spatial averaging,    .FA
$                           SEAWND : Use sea-point only wind input.  .FA
```

```
$ &PSMC DTIME = 39600.0, LATMIN=85.0, RFMAXD = 36.0, LvSMC=3, JEQT=1344
$
$ Rotated pole ------------------------------------------------------ $
$   Pole  parameters    : Namelist ROTD
$                           PLAT  : Rotated pole latitude
$                           PLON  : Rotated pole longitude
$                           UNROT : Logical, un-rotate directions to tru
$
$    These will be used to derive rotation angle corrections in the
$    model. The corrections are used for rotation of boundary spectra
$    and for restoring conventional lat/lon orientation of the
$    output spectra, mean direction or any related variables.
$    The PLAT/LON example below is a standard setting for Met
$    Office UK regional models.
$
$ &ROTD PLAT = 37.5, PLON = 177.5, UNROT = .TRUE. /
$
$ Output of 3D arrays----------------------------------------------- $
$ In order to limit the use of memory, arrays for 3D output fiels (i.e.
$ variables that are a function of both space and frequency, are not
$ declared, and thus cannot be used, unless specified by namelists.
$ NB: Output of 'first 5' moments E, th1m, sth1m, th2, sth2m allows to e
$  directional spectrum using, e.g. MEM (Lygre&Krogstad 1986).
$
$ Parameters (integers)   : Namelist OUTS
$ For the frequency spectrum E(f)
$                           E3D     : <=0: not declared, > 0: declared
$                           I1E3D   : First frequency index of output (de
$                           I2E3D   : Last frequency index of output  (de
$ For the mean direction th1m(f), and spread sth1m(f)
$                    TH1MF, STH1MF  : <=0: not declared, > 0: declared
$                 I1TH1MF, I1STH1MF: First frequency index of output (de
$                 I2TH1MF, I2STH1MF: First frequency index of output (de
$ For the mean direction th2m(f), and spread sth2m(f)
$                    TH2MF, STH2MF  : <=0: not declared, > 0: declared
$                 I1TH2MF, I1STH2MF: First frequency index of output (de
$                 I2TH2MF, I2STH2MF: First frequency index of output (de
$ For 2nd order pressure at K=0 (source of microseisms & microbaroms)
$                           P2SF   : <=0: not declared, > 0: declared
$                           I1P2SF : First frequency index of output (de
$                           I2P2SF : Last frequency index of output  (de
$ For the surface Stokes drift partitions (USP)
$                           USSP : First index (default is 1, should alw
$                           IUSSP : Last index (must be <= than NK and s
$                                               between 3 and ~10 with t
```

```
$                                          between accuracy and res
$                          STK_WN : List of wavenumbers (size of IUSSP)
$                          e.g.: USSP = 1, IUSSP=3, STK_WN = 0.04, 0.11
$                                provides 3 partitions of both x & y co
$                                with a reasonable accuracy for using i
$                                a climate model.
$
$ Miscellaneous ------------------------------------------------------ $
$   Misc. parameters    : Namelist MISC
$                          CICE0  : Ice concentration cut-off.
$                          CICEN  : Ice concentration cut-off.
$                          PMOVE  : Power p in GSE aleviation for
$                                   moving grids in Eq. (D.4).
$                          XSEED  : Xseed in seeding alg. (!/SEED).
$                          FLAGTR : Indicating presence and type of
$                                   subgrid information :
$                                    0 : No subgrid information.
$                                    1 : Transparancies at cell boun-
$                                        daries between grid points.
$                                    2 : Transp. at cell centers.
$                                    3 : Like 1 with cont. ice.
$                                    4 : Like 2 with cont. ice.
$                          TRCKCMPR : Logical variable (T/F). Set to F
$                                     disable "compression" of track ou
$                                     This simplifies post-processing.
$                                     Default is T and will create trac
$                                     output in the traditional manner
$                                     (WW3 v3, v4, v5).
$                          XP, XR, XFILT
$                                   Xp, Xr and Xf for the dynamic
$                                   integration scheme.
$                          IHMAX  : Number of discrete levels in part.
$                          HSPMIN : Minimum Hs in partitioning.
$                          WSM    : Wind speed multiplier in part.
$                          WSC    : Cut of wind sea fraction for
$                                   identifying wind sea in part.
$                          FLC    : Flag for combining wind seas in
$                                   partitioning.
$                          NOSW   : Number of partitioned swell fields
$                                   in field output.
$                          PTM    : Partioning method:
$                                    1 : Default WW3
$                                    2 : Watershedding + wind cutoff
$                                    3 : Watershedding only
$                                    4 : Wind speed cutoff only
```

```
$                                      5 : High/Low band cutoff (see PTFC
$                           PTFC   : Cutouf frequency for High/Low band
$                                    partioning (PTM=5). Default = 0.1Hz
$                           FMICHE : Constant in Miche limiter.
$                           STDX   : Space-Time Extremes X-Length
$                           STDY   : Space-Time Extremes Y-Length
$                           STDT   : Space-Time Extremes Duration
$                           P2SF   : ......
$
$ Diagnostic Sea-state Dependent Stress- - - - - - - - - - - - - - - - -
$   Reichl et al. 2014  : Namelist FLD1
$                           TAILTYPE  : High Frequency Tail Method
$                                       0: Constant value (prescribed)
$                                       1: Wind speed dependent
$                                          (Based on GFDL Hurricane
$                                          Model Z0 relationship)
$                           TAILLEV   : Level of high frequency tail
$                                       (if TAILTYPE==0)
$                                       Valid choices:
$                                       Capped min: 0.001, max: 0.02
$                           TAILT1    : Tail transition ratio 1
$                                       TAILT1*peak input frequency
$                                       is the first transition point of
$                                       the saturation specturm
$                                       Default is 1.25
$                           TAILT1    : Tail transition ratio 2
$                                       TAILT2*peak input frequency
$                                       is the second transition point o
$                                       the saturation specturm
$                                       Default is 3.00
$   Donelan et al. 2012 : Namelist FLD2
$                           TAILTYPE : See above (FLD1)
$                           TAILLEV  : See above (FLD1)
$                           TAILT1   : See above (FLD1)
$                           TAILT2   : See above (FLD1)
$
$ In the 'Out of the box' test setup we run with sub-grid obstacles
$ and with continuous ice treatment.
$
  &MISC CICE0 = 0.25, CICEN = 0.75, FLAGTR = 4 /
  &FLX3 CDMAX = 3.5E-3 , CTYPE = 0 /
$ &SDB1 BJGAM = 1.26, BJFLAG = .FALSE. /
$
$ Mandatory string to identify end of namelist input section.
$
```

```
END OF NAMELISTS
$
$ Define grid ------------------------------------------------------- $
$
$ Five records containing :
$
$  1 Type of grid, coordinate system and type of closure: GSTRG, FLAGLL,
$    CSTRG.  Grid closure can only be applied in spherical coordinates.
$      GSTRG  : String indicating type of grid :
$               'RECT'  : rectilinear
$               'CURV'  : curvilinear
$               'UNST'  : unstructured (triangle-based)
$      FLAGLL : Flag to indicate coordinate system :
$               T  : Spherical (lon/lat in degrees)
$               F  : Cartesian (meters)
$      CSTRG  : String indicating the type of grid index space closure :
$               'NONE'  : No closure is applied
$               'SMPL'  : Simple grid closure : Grid is periodic in the
$                       : i-index and wraps at i=NX+1. In other words,
$                       : (NX+1,J) => (1,J). A grid with simple closure
$                       : may be rectilinear or curvilinear.
$               'TRPL'  : Tripole grid closure : Grid is periodic in the
$                       : i-index and wraps at i=NX+1 and has closure at
$                       : j=NY+1. In other words, (NX+1,J<=NY) => (1,J)
$                       : and (I,NY+1) => (NX-I+1,NY). Tripole
$                       : grid closure requires that NX be even. A grid
$                       : with tripole closure must be curvilinear.
$  2 NX, NY. As the outer grid lines are always defined as land
$    points, the minimum size is 3x3.
$
$ Branch here based on grid type
$
$ IF ( RECTILINEAR GRID ) THEN
$
$  3 Grid increments SX, SY (degr.or m) and scaling (division) factor.
$    If CSTRG='SMPL', then SX is set to 360/NX.
$  4 Coordinates of (1,1) (degr.) and scaling (division) factor.
$
$ ELSE IF ( CURVILINEAR GRID ) THEN
$
$  3 Unit number of file with x-coordinate.
$    Scale factor and add offset: x <= scale_fac * x_read + add_offset.
$    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.
$
$   If the above unit number equals 10, then the x-coord is read from t
$   file.  The x-coord must follow the above record.  No comment lines
$   allowed within the x-coord input.
$
$  4 Unit number of file with y-coordinate.
$   Scale factor and add offset: y <= scale_fac * y_read + add_offset.
$   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.
$
$   If the above unit number equals 10, then the y-coord is read from t
$   file.  The y-coord must follow the above record.  No comment lines
$   allowed within the y-coord input.
$
$ ELSE IF ( UNSTRUCTURED GRID ) THEN
$   Nothing to declare: all the data will be read from the GMESH file
$ END IF ( CURVILINEAR GRID )
$
$  5 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.
$
$    If the above unit number equals 10, then the bottom depths are read
$    this file.  The depths must follow the above record.  No comment li
$    allowed within the depth input. In the case of unstructured grids,
$    is expected to be a GMESH grid file containing node and element lis
$
$ ------------------------------------------------------------------------
$ Example for rectilinear grid with spherical (lon/lat) coordinate syste
$ Note that for Cartesian coordinates the unit is meters (NOT km).
$
     'RECT'  T  'NONE'
     12     12
      1.     1.     4.
     -1.    -1.     4.
     -0.1 2.50  10  -10. 3 1 '(....)' 'NAME' 'bottom.inp'
$
  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
$
$ ------------------------------------------------------------------------
$ Example for curvilinear grid with spherical (lon/lat) coordinate syste
$ Same spatial grid as preceding rectilinear example.
$ Note that for Cartesian coordinates the unit is meters (NOT km).
$
$     'CURV'  T  'NONE'
```

```
$     12     12
$
$     10 0.25 -0.5 3 1 '(....)' 'NAME' 'x.inp'
$
$   1  2  3  4  5  6  7  8  9 10 11 12
$   1  2  3  4  5  6  7  8  9 10 11 12
$   1  2  3  4  5  6  7  8  9 10 11 12
$   1  2  3  4  5  6  7  8  9 10 11 12
$   1  2  3  4  5  6  7  8  9 10 11 12
$   1  2  3  4  5  6  7  8  9 10 11 12
$   1  2  3  4  5  6  7  8  9 10 11 12
$   1  2  3  4  5  6  7  8  9 10 11 12
$   1  2  3  4  5  6  7  8  9 10 11 12
$   1  2  3  4  5  6  7  8  9 10 11 12
$   1  2  3  4  5  6  7  8  9 10 11 12
$   1  2  3  4  5  6  7  8  9 10 11 12
$
$     10 0.25 0.5 3 1 '(....)' 'NAME' 'y.inp'
$
$   1  1  1  1  1  1  1  1  1  1  1  1
$   2  2  2  2  2  2  2  2  2  2  2  2
$   3  3  3  3  3  3  3  3  3  3  3  3
$   4  4  4  4  4  4  4  4  4  4  4  4
$   5  5  5  5  5  5  5  5  5  5  5  5
$   6  6  6  6  6  6  6  6  6  6  6  6
$   7  7  7  7  7  7  7  7  7  7  7  7
$   8  8  8  8  8  8  8  8  8  8  8  8
$   9  9  9  9  9  9  9  9  9  9  9  9
$  10 10 10 10 10 10 10 10 10 10 10 10
$  11 11 11 11 11 11 11 11 11 11 11 11
$  12 12 12 12 12 12 12 12 12 12 12 12
$
$     -0.1 2.50  10  -10. 3 1 '(....)' 'NAME' 'bottom.inp'
$
$  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
$
$ -------------------------------------------------------------
$ SMC grid use the same spherical lat-lon grid parameters
$     'RECT'  T  'SMPL'
$   1024    704
$ SMC grid base level resolution dlon dlat and start lon lat
$ 0.35156250  0.23437500   1.
$ 0.17578125 -78.6328125   1.
$
$ Normal depth input line is used to passing the minimum depth
$ though the depth file is not read for SMC grid.
$  -0.1  10.0  30   -1. 1 1 '(....)' 'NAME' 'SMC25Depth.dat'
$ SMC cell and face arrays and obstruction ratio:
$  32  1  1  '(....)'  'S6125MCels.dat'
$  33  1  1  '(....)'  'S6125ISide.dat'
$  34  1  1  '(....)'  'S6125JSide.dat'
$  31  1.0  1  1 '(...)' 'NAME'  'SMC25Subtr.dat'
$ The input boundary cell file is only needed when NBISMC > 0.
$  35  1  1  '(....)'  'S6125Bundy.dat'
$ Extra cell and face arrays for Arctic part if ARC is selected.
$  36  1  1  '(....)'  'S6125MBArc.dat'
$  37  1  1  '(....)'  'S6125AISid.dat'
$  38  1  1  '(....)'  'S6125AJSid.dat'
$ Normal land-sea mask file input line is kept but file is not used.
$  39  1  1  '(....)'  'NAME'  'S6125Masks.dat'
$    Boundary cell id list file (unit 35) is only required if boundary
$    cell number entered above is non-zero.  The cell id number should b
$    the sequential number in the cell array (unit 32) S625MCels.dat.
$
$ If sub-grid information is available as indicated by FLAGTR above,
$ additional input to define this is needed below. In such cases a
$ field of fractional obstructions at or between grid points needs to
$ be supplied.  First the location and format of the data is defined
$ by (as above) :
$  - Unit number of file (can be 10, and/or identical to bottom depth
$    unit), scale factor for fractional obstruction, IDLA, IDFM,
$    format for formatted read, FROM and filename
$
   10 0.2  3 1 '(....)' 'NAME' 'obstr.inp'
$
$ *** NOTE if this unit number is the same as the previous bottom
$     depth unit number, it is assumed that this is the same file
$     without further checks.                                    ***
$
```

```
$ If the above unit number equals 10, the bottom data is read from
$ this file and follows below (no intermediate comment lines allowed,
$ except between the two fields).
$
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 5 0 0 0 0 0
  0 0 0 0 0 0 5 0 0 0 0 0
  0 0 0 0 0 0 4 0 0 0 0 0
  0 0 0 0 0 0 4 0 0 0 0 0
  0 0 0 0 0 0 5 0 0 0 0 0
  0 0 0 0 0 0 5 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
$
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 5 5 5 0
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
$
$ *** NOTE size of fields is always NX * NY                        ***
$
$ Input boundary points and excluded points --------------------------- $
$    The first line identifies where to get the map data, by unit number
$    IDLA and IDFM, format for formatted read, FROM and filename
$    if FROM = 'PART', then segmented data is read from below, else
$    the data is read from file as with the other inputs (as INTEGER)
$
   10 3 1 '(....)' 'PART' 'mapsta.inp'
$
$ Read the status map from file ( FROM != PART ) ---------------------- $
$
$ 3 3 3 3 3 3 3 3 3 3 3 3
$ 3 2 1 1 1 1 0 1 1 1 1 3
$ 3 2 1 1 1 1 0 1 1 1 1 3
```

```
$ 3 2 1 1 1 1 0 1 1 1 1 3
$ 3 2 1 1 1 1 0 0 1 1 1 3
$ 3 2 1 1 1 1 1 1 1 1 1 3
$ 3 2 1 1 1 1 1 1 1 1 1 3
$ 3 2 1 1 1 1 1 1 1 1 1 3
$ 3 2 1 1 1 1 1 1 1 1 1 3
$ 3 2 1 1 1 1 1 1 1 1 1 3
$ 3 2 1 1 1 1 1 1 1 1 1 3
$ 3 3 3 3 3 3 3 3 3 3 3 3
$
$ The legend for the input map is :
$
$    0 : Land point.
$    1 : Regular sea point.
$    2 : Active boundary point.
$    3 : Point excluded from grid.
$
$ Input boundary points from segment data ( FROM = PART ) ------------ $
$   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
$
$ Excluded grid points from segment data ( FROM != PART )
$   First defined as lines, identical to the definition of the input
$   boundary points, and closed the same way.
$
      0   0   F
$
$   Second, define a point in a closed body of sea points to remove
$   the entire body of sea points. Also close by point (0,0)
$
      0   0
$
```

```
$ Sedimentary bottom map if namelist &SBT4 SEDMAPD50 = T
$
$     22  1. 1 1 '(f10.6)' 'NAME' 'SED.txt'
$
$ Output boundary points ------------------------------------------- $
$ Output boundary points are defined as a number of straight lines,
$ defined by its starting point (X0,Y0), increments (DX,DY) and number
$ of points. A negative number of points starts a new output file.
$ Note that this data is only generated if requested by the actual
$ program. Example again for spherical grid in degrees. Note, these do
$ not need to be defined for data transfer between grids in the multi
$ grid driver.
$
     1.75  1.50  0.25 -0.10     3
     2.25  1.50 -0.10  0.00    -6
     0.10  0.10  0.10  0.00   -10
$
$  Close list by defining line with 0 points (mandatory)
$
     0.    0.    0.    0.       0
$
$ ------------------------------------------------------------------- $
$ End of input file                                                   $
$ ------------------------------------------------------------------- $
```

end of example input file (traditional form)

G.1.2 ww3_grid.nml

start of example input file (namelist form)

```
! ------------------------------------------------------------------- !
! WAVEWATCH III - ww3_grid.nml - Grid pre-processing                  !
! ------------------------------------------------------------------- !


! ------------------------------------------------------------------- !
! Define the spectrum parameterization via SPECTRUM_NML namelist
!
! * namelist must be terminated with /
! * definitions & defaults:
```

```
!     SPECTRUM%XFR         = 0.            ! frequency increment
!     SPECTRUM%FREQ1       = 0.            ! first frequency (Hz)
!     SPECTRUM%NK          = 0             ! number of frequencies (wave
!     SPECTRUM%NTH         = 0             ! number of direction bins
!     SPECTRUM%THOFF       = 0.            ! relative offset of first di
! ------------------------------------------------------------------- !
&SPECTRUM_NML
  SPECTRUM%XFR           =  1.1
  SPECTRUM%FREQ1         =  0.04118
  SPECTRUM%NK            =  32
  SPECTRUM%NTH           =  24
/


! ------------------------------------------------------------------- !
! Define the run parameterization via RUN_NML namelist
!
! * namelist must be terminated with /
! * definitions & defaults:
!     RUN%FLDRY            = F             ! dry run (I/O only, no calcu
!     RUN%FLCX             = F             ! x-component of propagation
!     RUN%FLCY             = F             ! y-component of propagation
!     RUN%FLCTH            = F             ! direction shift
!     RUN%FLCK             = F             ! wavenumber shift
!     RUN%FLSOU            = F             ! source terms
! ------------------------------------------------------------------- !
&RUN_NML
  RUN%FLCX            = T
  RUN%FLCY            = T
  RUN%FLCTH           = T
  RUN%FLSOU           = T
/


! ------------------------------------------------------------------- !
! Define the timesteps parameterization via TIMESTEPS_NML namelist
!
! * It is highly recommended to set up time steps which are multiple
!   between them.
!
! * The first time step to calculate is the maximum CFL time step
!   which depend on the lowest frequency FREQ1 previously set up and the
!   lowest spatial grid resolution in meters DXY.
!   reminder : 1 degree=60minutes // 1minute=1mile // 1mile=1.852km
!   The formula for the CFL time is :
```

```
!   Tcfl = DXY / (G / (FREQ1*4*Pi) ) with the constants Pi=3,14 and G=9.
!   DTXY  ~= 90% Tcfl
!   DTMAX ~= 3 * DTXY   (maximum global time step limit)
!
! * The refraction time step depends on how strong can be the current ve
!   on your grid :
!   DTKTH ~= DTMAX / 2   ! in case of no or light current velocities
!   DTKTH ~= DTMAX / 10  ! in case of strong current velocities
!
! * The source terms time step is usually defined between 5s and 60s.
!   A common value is 10s.
!   DTMIN ~= 10
!
! * namelist must be terminated with /
! * definitions & defaults:
!     TIMESTEPS%DTMAX      = 0.         ! maximum global time step (s)
!     TIMESTEPS%DTXY       = 0.         ! maximum CFL time step for x-y
!     TIMESTEPS%DTKTH      = 0.         ! maximum CFL time step for k-th
!     TIMESTEPS%DTMIN      = 0.         ! minimum source term time step
! -------------------------------------------------------------------- !
&TIMESTEPS_NML
  TIMESTEPS%DTMAX         =  480.
  TIMESTEPS%DTXY          =  160.
  TIMESTEPS%DTKTH         =  240.
  TIMESTEPS%DTMIN         =   10.
/


! -------------------------------------------------------------------- !
! Define the grid to preprocess via GRID_NML namelist
!
! * the tunable parameters for source terms, propagation schemes, and
!    numerics are read using namelists.
! * Any namelist found in the folowing sections is temporarily written
!   to param.scratch, and read from there if necessary.
! * The order of the namelists is immaterial.
! * Namelists not needed for the given switch settings will be skipped
!   automatically
!
! * grid type can be :
!    'RECT' : rectilinear
!    'CURV' : curvilinear
!    'UNST' : unstructured (triangle-based)
!
! * coordinate system can be :
```

```
!    'SPHE' : Spherical (degrees)
!    'CART' : Cartesian (meters)
!
! * grid closure can only be applied in spherical coordinates
!
! * grid closure can be :
!    'NONE' : No closure is applied
!    'SMPL' : Simple grid closure. Grid is periodic in the
!           : i-index and wraps at i=NX+1. In other words,
!           : (NX+1,J) => (1,J). A grid with simple closure
!           : may be rectilinear or curvilinear.
!    'TRPL' : Tripole grid closure : Grid is periodic in the
!           : i-index and wraps at i=NX+1 and has closure at
!           : j=NY+1. In other words, (NX+1,J<=NY) => (1,J)
!           : and (I,NY+1) => (NX-I+1,NY). Tripole
!           : grid closure requires that NX be even. A grid
!           : with tripole closure must be curvilinear.
!
! * The coastline limit depth is the value which distinguish the sea
!   points to the land points. All the points with depth values (ZBIN)
!   greater than this limit (ZLIM) will be considered as excluded points
!   and will never be wet points, even if the water level grows over.
!   It can only overwrite the status of a sea point to a land point.
!   The value must have a negative value under the mean sea level
!
! * The minimum water depth allowed to compute the model is the absolute
!   depth value (DMIN) used in the model if the input depth is lower to
!   avoid the model to blow up.
!
! * namelist must be terminated with /
! * definitions & defaults:
!     GRID%NAME             = 'unset'            ! grid name (30 char)
!     GRID%NML              = 'namelists.nml'    ! namelists filename
!     GRID%TYPE             = 'unset'            ! grid type
!     GRID%COORD            = 'unset'            ! coordinate system
!     GRID%CLOS             = 'unset'            ! grid closure
!
!     GRID%ZLIM             = 0.        ! coastline limit depth (m)
!     GRID%DMIN             = 0.        ! abs. minimum water depth (m)
! ------------------------------------------------------------------- !
&GRID_NML
  GRID%NAME             =  'GULF OF NOWHERE'
  GRID%NML              =  'namelists.nml'
  GRID%TYPE             =  'RECT'
  GRID%COORD            =  'SPHE'
```

```
  GRID%CLOS             =  'SMPL'
  GRID%ZLIM             =  -0.10
  GRID%DMIN             =  2.5
/



! ------------------------------------------------------------------- !
! Define the rectilinear grid type via RECT_NML namelist
! - only for RECT grids -
!
! * The minimum grid size is 3x3.
!
! * If the grid increments SX and SY are given in minutes of arc, the sc
!   factor SF must be set to 60. to provide an increment factor in degre
!
! * If CSTRG='SMPL', then SX is forced to 360/NX.
!
! * value <= value_read / scale_fac
!
! * namelist must be terminated with /
! * definitions & defaults:
!     RECT%NX               = 0        ! number of points along x-axis
!     RECT%NY               = 0        ! number of points along y-axis
!
!     RECT%SX               = 0.       ! grid increment along x-axis
!     RECT%SY               = 0.       ! grid increment along y-axis
!     RECT%SF               = 1.       ! scaling division factor for x-y
!
!     RECT%X0               = 0.       ! x-coordinate of lower-left corn
!     RECT%Y0               = 0.       ! y-coordinate of lower-left corn
!     RECT%SF0              = 1.       ! scaling division factor for x0,
! ------------------------------------------------------------------- !
&RECT_NML
  RECT%NX               =  720
  RECT%NY               =  360
/



! ------------------------------------------------------------------- !
! Define the curvilinear grid type via CURV_NML namelist
! - only for CURV grids -
!
! * The minimum grid size is 3x3.
```

```
!
! * If CSTRG='SMPL', then SX is forced to 360/NX.
!
! * value <= scale_fac * value_read + add_offset
!
! * IDLA : Layout indicator :
!                  1   : Read line-by-line bottom to top. (default)
!                  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. (default)
!                  2   : Fixed format.
!                  3   : Unformatted.
! * FORMAT : element format to read :
!               '(....)'  : auto detected (default)
!               '(f10.6)' : float type
!
! * Example :
!      IDF  SF    OFF  IDLA  IDFM  FORMAT    FILENAME
!      21   0.25 -0.5  3     1    '(....)'  'x.inp'
!      22   0.25  0.5  3     1    '(....)'  'y.inp'
!
! * namelist must be terminated with /
! * definitions & defaults:
!     CURV%NX               = 0        ! number of points along x-axis
!     CURV%NY               = 0        ! number of points along y-axis
!
!     CURV%XCOORD%SF         = 1.       ! x-coord scale factor
!     CURV%XCOORD%OFF        = 0.       ! x-coord add offset
!     CURV%XCOORD%FILENAME   = 'unset'  ! x-coord filename
!     CURV%XCOORD%IDF        = 21       ! x-coord file unit number
!     CURV%XCOORD%IDLA       = 1        ! x-coord layout indicator
!     CURV%XCOORD%IDFM       = 1        ! x-coord format indicator
!     CURV%XCOORD%FORMAT     = '(....)' ! x-coord formatted read format
!
!     CURV%YCOORD%SF         = 1.       ! y-coord scale factor
!     CURV%YCOORD%OFF        = 0.       ! y-coord add offset
!     CURV%YCOORD%FILENAME   = 'unset'  ! y-coord filename
!     CURV%YCOORD%IDF        = 22       ! y-coord file unit number
!     CURV%YCOORD%IDLA       = 1        ! y-coord layout indicator
!     CURV%YCOORD%IDFM       = 1        ! y-coord format indicator
!     CURV%YCOORD%FORMAT     = '(....)' ! y-coord formatted read format
! ------------------------------------------------------------------- !
&CURV_NML
```

```
  CURV%NX               =  720
  CURV%NY               =  360
!
  CURV%XCOORD%SF         = 0.25
  CURV%XCOORD%OFF        = -0.5
  CURV%XCOORD%FILENAME   = 'x.inp'
  CURV%XCOORD%IDLA       = 3
!
  CURV%YCOORD%SF         = 0.25
  CURV%YCOORD%OFF        = 0.5
  CURV%YCOORD%FILENAME   = 'y.inp'
  CURV%YCOORD%IDLA       = 3
/


! -------------------------------------------------------------------- !
! Define the unstructured grid type via UNST_NML namelist
! - only for UNST grids -
!
! * The minimum grid size is 3x3.
!
! * &MISC namelist must be removed
!
! * The depth value must have negative values under the mean sea level
!
! * The map value must be set as :
!    -2 : Excluded boundary point (covered by ice)
!    -1 : Excluded sea point (covered by ice)
!     0 : Excluded land point
!     1 : Sea point
!     2 : Active boundary point
!     3 : Excluded grid point
!     7 : Ice point
!
! * the file must be a GMESH grid file containing node and element lists
!
! * Extra open boundary list file with UGOBCFILE in namelist &UNST
!   An example is given in regtest ww3_tp2.7
!
! * value <= scale_fac * value_read
!
! * IDLA : Layout indicator :
!                  1   : Read line-by-line bottom to top. (default)
!                  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. (default)
!                  2   : Fixed format.
!                  3   : Unformatted.
! * FORMAT : element format to read :
!               '(....)'  : auto detected (default)
!               '(f10.6)' : float type
!
! * Example :
!      IDF  SF   IDLA  IDFM   FORMAT       FILENAME
!      20  -1.   4     2     '(20f10.2)'  'ngug.msh'
!
! * namelist must be terminated with /
! * definitions & defaults:
!     UNST%SF             = 1.       ! unst scale factor
!     UNST%FILENAME       = 'unset'  ! unst filename
!     UNST%IDF            = 20       ! unst file unit number
!     UNST%IDLA           = 1        ! unst layout indicator
!     UNST%IDFM           = 1        ! unst format indicator
!     UNST%FORMAT         = '(....)' ! unst formatted read format
!
!     UNST%UGOBCFILE      = 'unset'  ! additional boundary list file
! ------------------------------------------------------------------- !
&UNST_NML
  UNST%SF             = -1.
  UNST%FILENAME       = 'ngug.msh'
  UNST%IDLA           = 4
  UNST%IDFM           = 2
  UNST%FORMAT         = '(20f10.2)'
/


! ------------------------------------------------------------------- !
! Define the spherical multiple-cell grid via SMC_NML namelist
! - only for SMC grids -
!
! * SMC cell 'MCELS' and face 'ISIDE & JSIDE' arrays
!   and obstruction ratio 'SUBTR'.
!
! * The input boundary cell file 'BUNDY' is only needed when NBISMC > 0.
!   Boundary cell id list file (unit 35) is only required if boundary
!   cell number entered above is non-zero.  The cell id number should be
!   the sequential number in the cell array (unit 31) S625MCels.dat.
!
```

```
! * Extra cell and face arrays for Arctic part if switch ARC is selected
!
! * Example :
!      IDF  IDLA  IDFM  FORMAT   FILENAME
!      31   1     1    '(....)'  'S6125MCels.dat'
!      32   1     1    '(....)'  'S6125ISide.dat'
!      33   1     1    '(....)'  'S6125JSide.dat'
!      34   1     1    '(....)'  'SMC25Subtr.dat'
!      35   1     1    '(....)'  'S6125Bundy.dat'
!      36   1     1    '(....)'  'S6125MBArc.dat'
!      37   1     1    '(....)'  'S6125AISid.dat'
!      38   1     1    '(....)'  'S6125AJSid.dat'
!
! * namelist must be terminated with /
! * definitions & defaults:
!     SMC%MCELS%FILENAME       = 'unset'  ! MCels filename
!     SMC%MCELS%IDF            = 31       ! MCels file unit number
!     SMC%MCELS%IDLA           = 1        ! MCels layout indicator
!     SMC%MCELS%IDFM           = 1        ! MCels format indicator
!     SMC%MCELS%FORMAT         = '(....)' ! MCels formatted read format
!
!     SMC%ISIDE%FILENAME       = 'unset'  ! ISide filename
!     SMC%ISIDE%IDF            = 32       ! ISide file unit number
!     SMC%ISIDE%IDLA           = 1        ! ISide layout indicator
!     SMC%ISIDE%IDFM           = 1        ! ISide format indicator
!     SMC%ISIDE%FORMAT         = '(....)' ! ISide formatted read format
!
!     SMC%JSIDE%FILENAME       = 'unset'  ! JSide filename
!     SMC%JSIDE%IDF            = 33       ! JSide file unit number
!     SMC%JSIDE%IDLA           = 1        ! JSide layout indicator
!     SMC%JSIDE%IDFM           = 1        ! JSide format indicator
!     SMC%JSIDE%FORMAT         = '(....)' ! JSide formatted read format
!
!     SMC%SUBTR%FILENAME       = 'unset'  ! Subtr filename
!     SMC%SUBTR%IDF            = 34       ! Subtr file unit number
!     SMC%SUBTR%IDLA           = 1        ! Subtr layout indicator
!     SMC%SUBTR%IDFM           = 1        ! Subtr format indicator
!     SMC%SUBTR%FORMAT         = '(....)' ! Subtr formatted read format
!
!     SMC%BUNDY%FILENAME       = 'unset'  ! Bundy filename
!     SMC%BUNDY%IDF            = 35       ! Bundy file unit number
!     SMC%BUNDY%IDLA           = 1        ! Bundy layout indicator
!     SMC%BUNDY%IDFM           = 1        ! Bundy format indicator
!     SMC%BUNDY%FORMAT         = '(....)' ! Bundy formatted read format
!
```

```
!     SMC%MBARC%FILENAME       = 'unset'  ! MBArc filename
!     SMC%MBARC%IDF            = 36       ! MBArc file unit number
!     SMC%MBARC%IDLA           = 1        ! MBArc layout indicator
!     SMC%MBARC%IDFM           = 1        ! MBArc format indicator
!     SMC%MBARC%FORMAT         = '(....)' ! MBArc formatted read format
!
!     SMC%AISID%FILENAME       = 'unset'  ! AISid filename
!     SMC%AISID%IDF            = 37       ! AISid file unit number
!     SMC%AISID%IDLA           = 1        ! AISid layout indicator
!     SMC%AISID%IDFM           = 1        ! AISid format indicator
!     SMC%AISID%FORMAT         = '(....)' ! AISid formatted read format
!
!     SMC%AJSID%FILENAME       = 'unset'  ! AJSid filename
!     SMC%AJSID%IDF            = 38       ! AJSid file unit number
!     SMC%AJSID%IDLA           = 1        ! AJSid layout indicator
!     SMC%AJSID%IDFM           = 1        ! AJSid format indicator
!     SMC%AJSID%FORMAT         = '(....)' ! AJSid formatted read format
! ------------------------------------------------------------------- !
&SMC_NML
  SMC%MCELS%FILENAME       = 'S6125MCels.dat'
  SMC%ISIDE%FILENAME       = 'S6125ISide.dat'
  SMC%JSIDE%FILENAME       = 'S6125JSide.dat'
  SMC%SUBTR%FILENAME       = 'SMC25Subtr.dat'
  SMC%BUNDY%FILENAME       = 'S6125Bundy.dat'
  SMC%MBARC%FILENAME       = 'S6125MBArc.dat'
  SMC%AISID%FILENAME       = 'S6125AISid.dat'
  SMC%AJSID%FILENAME       = 'S6125AJSid.dat'
/



! ------------------------------------------------------------------- !
! Define the depth to preprocess via DEPTH_NML namelist
! - for RECT and CURV grids -
!
! * if no obstruction subgrid, need to set &MISC FLAGTR = 0
!
! * The depth value must have negative values under the mean sea level
!
! * value <= value_read * scale_fac
!
! * IDLA : Layout indicator :
!                  1   : Read line-by-line bottom to top.  (default)
!                  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.  (default)
!                 2   : Fixed format.
!                 3   : Unformatted.
! * FORMAT : element format to read :
!              '(....)'  : auto detected  (default)
!              '(f10.6)' : float type
!
! * Example :
!      IDF  SF     IDLA  IDFM   FORMAT    FILENAME
!      50   0.001  1     1     '(....)'  'GLOB-30M.bot'
!
! * namelist must be terminated with /
! * definitions & defaults:
!     DEPTH%SF             = 1.       ! scale factor
!     DEPTH%FILENAME       = 'unset'  ! filename
!     DEPTH%IDF            = 50       ! file unit number
!     DEPTH%IDLA           = 1        ! layout indicator
!     DEPTH%IDFM           = 1        ! format indicator
!     DEPTH%FORMAT         = '(....)' ! formatted read format
! ------------------------------------------------------------------- !
&DEPTH_NML
  DEPTH%SF             = 0.001
  DEPTH%FILENAME       = 'GLOB-30M.bot'
/



! ------------------------------------------------------------------- !
! Define the point status map via MASK_NML namelist
! - only for RECT and CURV grids -
!
! * If no mask defined, INBOUND can be used to set active boundaries
!
! * IDLA : Layout indicator :
!                 1   : Read line-by-line bottom to top.  (default)
!                 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.  (default)
!                 2   : Fixed format.
!                 3   : Unformatted.
! * FORMAT : element format to read :
```

```
!               '(....)'  : auto detected  (default)
!               '(f10.6)' : float type
!
! * Example :
!      IDF  IDLA  IDFM   FORMAT    FILENAME
!      60   1     1     '(....)'  'GLOB-30M.mask'
!
! * The legend for the input map is :
!    -2 : Excluded boundary point (covered by ice)
!    -1 : Excluded sea point (covered by ice)
!     0 : Excluded land point
!     1 : Sea point
!     2 : Active boundary point
!     3 : Excluded grid point
!     7 : Ice point
!
! * namelist must be terminated with /
! * definitions & defaults:
!     MASK%FILENAME         = 'unset'  ! filename
!     MASK%IDF              = 60       ! file unit number
!     MASK%IDLA             = 1        ! layout indicator
!     MASK%IDFM             = 1        ! format indicator
!     MASK%FORMAT           = '(....)' ! formatted read format
! ------------------------------------------------------------------- !
&MASK_NML
  MASK%FILENAME         = 'GLOB-30M.mask'
/




! ------------------------------------------------------------------- !
! Define the obstruction map via OBST_NML namelist
! - only for RECT and CURV grids -
!
! * only used if &MISC FLAGTR = 1 in param.nml
!                (transparencies at cell boundaries)
!          or if &MISC FLAGTR = 2 in param.nml
!                (transparencies at cell centers)
!          or if &MISC FLAGTR = 3 in param.nml
!                (transparencies at cell boundaries with cont. ice)
!          or if &MISC FLAGTR = 4 in param.nml
!                (transparencies at cell centers with cont. ice)
!
! * value <= value_read * scale_fac
```

```
!
! * IDLA : Layout indicator :
!                  1   : Read line-by-line bottom to top.  (default)
!                  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.  (default)
!                  2   : Fixed format.
!                  3   : Unformatted.
! * FORMAT : element format to read :
!               '(....)'  : auto detected  (default)
!               '(f10.6)' : float type
!
! * Example :
!      IDF  SF      IDLA  IDFM   FORMAT    FILENAME
!      70   0.0001  1     1     '(....)'  'GLOB-30M.obst'
!
! * If the file unit number equals 10, then the data is read from this
!   file. The data must follow the above record. No comment lines are
!   allowed within the data input.
!
! * In the case of unstructured grids, no obstruction file can be added
!
! * namelist must be terminated with /
! * definitions & defaults:
!     OBST%SF              = 1.       ! scale factor
!     OBST%FILENAME        = 'unset'  ! filename
!     OBST%IDF             = 70       ! file unit number
!     OBST%IDLA            = 1        ! layout indicator
!     OBST%IDFM            = 1        ! format indicator
!     OBST%FORMAT          = '(....)' ! formatted read format
! ------------------------------------------------------------------- !
&OBST_NML
  OBST%SF              = 0.0001
  OBST%FILENAME        = 'GLOB-30M.obst'
/



! ------------------------------------------------------------------- !
! Define the reflexion slope map via SLOPE_NML namelist
! - only for RECT and CURV grids -
!
! * only used if &REF1 REFMAP = 2 defined in param.nml
```

```
!
! * value <= value_read * scale_fac
!
! * IDLA : Layout indicator :
!                  1   : Read line-by-line bottom to top.  (default)
!                  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.  (default)
!                  2   : Fixed format.
!                  3   : Unformatted.
! * FORMAT : element format to read :
!               '(....)'  : auto detected  (default)
!               '(f10.6)' : float type
!
! * Example :
!      IDF  SF      IDLA  IDFM   FORMAT    FILENAME
!      80   0.0001  1     1     '(....)'  'GLOB-30M.slope'
!
! * In the case of unstructured grids, no sed file can be added
!
! * namelist must be terminated with /
! * definitions & defaults:
!     SLOPE%SF             = 1.       ! scale factor
!     SLOPE%FILENAME       = 'unset'  ! filename
!     SLOPE%IDF            = 80       ! file unit number
!     SLOPE%IDLA           = 1        ! layout indicator
!     SLOPE%IDFM           = 1        ! format indicator
!     SLOPE%FORMAT         = '(....)' ! formatted read format
! ------------------------------------------------------------------- !
&SLOPE_NML
  SLOPE%SF             = 0.0001
  SLOPE%FILENAME       = 'GLOB-30M.slope'
/



! ------------------------------------------------------------------- !
! Define the sedimentary bottom map via SED_NML namelist
!
! * only used if &SBT4 SEDMAPD50 = T defined in param.nml
!
! * value <= value_read * scale_fac
!
```

```
! * IDLA : Layout indicator :
!                  1   : Read line-by-line bottom to top.  (default)
!                  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.  (default)
!                  2   : Fixed format.
!                  3   : Unformatted.
! * FORMAT : element format to read :
!               '(....)'  : auto detected  (default)
!               '(f10.6)' : float type
!
! * Example :
!      IDF SF  IDLA  IDFM   FORMAT    FILENAME
!      90  1.  1     2     '(f10.6)' 'SED.txt'
!
! * In the case of unstructured grids, no sed file can be added
!
! * namelist must be terminated with /
! * definitions & defaults:
!     SED%SF               = 1.       ! scale factor
!     SED%FILENAME         = 'unset'  ! filename
!     SED%IDF              = 90       ! file unit number
!     SED%IDLA             = 1        ! layout indicator
!     SED%IDFM             = 1        ! format indicator
!     SED%FORMAT           = '(....)' ! formatted read format
! ------------------------------------------------------------------- !
&SED_NML
  SED%FILENAME         = 'SED.txt'
  SED%IDFM             = 2
  SED%FORMAT           = '(f10.6)'
/




! ------------------------------------------------------------------- !
! Define the input boundary points via INBND_COUNT_NML and
!                                      INBND_POINT_NML namelist
! - for RECT, CURV and UNST grids -
!
! * If no mask defined, INBOUND can be used
!
! * If the actual input data is not defined in the actual wave model run
```

```
!   the initial conditions will be applied as constant boundary conditio
!
! * The number of points is defined by INBND_COUNT
!
! * The points must start from index 1 to N
!
! * 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.
!
! * Included point :
!     grid points from segment data
!     Defines as lines identifying points at which
!     input boundary conditions are to be defined.
!
! * namelist must be terminated with /
! * definitions & defaults:
!     INBND_COUNT%N_POINT     = 0        ! number of segments
!
!     INBND_POINT(I)%X_INDEX  = 0        ! x index included point
!     INBND_POINT(I)%Y_INDEX  = 0        ! y index included point
!     INBND_POINT(I)%CONNECT  = F        ! connect flag
!
! OR
!     INBND_POINT(I)          = 0 0 F    ! included point
! ------------------------------------------------------------------- !
&INBND_COUNT_NML
  INBND_COUNT%N_POINT    = 2
/

&INBND_POINT_NML
  INBND_POINT(1)         = 2 2  F
  INBND_POINT(2)         = 2 11 T
/


! ------------------------------------------------------------------- !
! Define the excluded points and bodies via EXCL_COUNT_NML, EXCL_POINT_N
!                                           and EXCL_BODY_NML namelist
! - only for RECT and CURV grids -
!
! * If no mask defined, EXCL can NOT be used
!
```

```
! * The number of points and bodies are defined by EXCL_COUNT
!
! * The points and bodies must start from index 1 to N
!
! * 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.
!
! * Excluded point :
!     grid points from segment data
!     Defined as lines identifying points at which
!     input boundary conditions are to be excluded.
!
! * Excluded body:
!     Define a point in a closed body of sea points to remove the
!     entire body of sea points.
!
! * namelist must be terminated with /
! * definitions & defaults:
!     EXCL_COUNT%N_POINT      = 0        ! number of segments
!     EXCL_COUNT%N_BODY       = 0        ! number of bodies
!
!     EXCL_POINT(J)%X_INDEX   = 0        ! x index excluded point
!     EXCL_POINT(J)%Y_INDEX   = 0        ! y index excluded point
!     EXCL_POINT(J)%CONNECT   = F        ! connect flag
!
!     EXCL_BODY(K)%X_INDEX    = 0        ! x index excluded body
!     EXCL_BODY(K)%Y_INDEX    = 0        ! y index excluded body
! OR
!     EXCL_POINT(J)           = 0 0 F    ! excluded point
!     EXCL_BODY(K)            = 0 0      ! excluded body
! ------------------------------------------------------------------- !
&EXCL_COUNT_NML
  EXCL_COUNT%N_POINT  = 2
  EXCL_COUNT%N_BODY   = 1
/

&EXCL_POINT_NML
  EXCL_POINT(1)       = 20  2 F
  EXCL_POINT(2)       = 20 11 T
/

&EXCL_BODY_NML
```

```
  EXCL_BODY(1)        = 10 15
/


! ------------------------------------------------------------------- !
! Define the output boundary points via OUTBND_COUNT_NML and
!                                       OUTBND_LINE_NML namelist
! - only for RECT and CURV grids -
!
! * It will creates a nest file with output boundaries for a inner grid.
!   The prefered way to do it is to use ww3_bounc program.
!
! * These do not need to be defined for data transfer between grids in
!    the multi grid driver.
!
! * The number of lines are defined by OUTBND_COUNT
!
! * The lines must start from index 1 to N
!
! * 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.
!
! * Example for spherical grid in degrees :
!     '1.75  1.50  0.25 -0.10     3'
!     '2.25  1.50 -0.10  0.00    -6'
!     '0.10  0.10  0.10  0.00   -10'
!
! * namelist must be terminated with /
! * definitions & defaults:
!     OUTBND_COUNT%N_LINE   = 0               ! number of lines
!
!     OUTBND_LINE(I)%X0     = 0.              ! x index start point
!     OUTBND_LINE(I)%Y0     = 0.              ! y index start point
!     OUTBND_LINE(I)%DX     = 0.              ! x-along increment
!     OUTBND_LINE(I)%DY     = 0.              ! y-along increment
!     OUTBND_LINE(I)%NP     = 0               ! number of points
! OR
!     OUTBND_LINE(I)        = 0. 0. 0. 0. 0   ! included lines
! ------------------------------------------------------------------- !
&OUTBND_COUNT_NML
  OUTBND_COUNT%N_LINE    = 3
/

&OUTBND_LINE_NML
```

```
  OUTBND_LINE(1)         = 1.75  1.50  0.25 -0.10   3
  OUTBND_LINE(2)         = 2.25  1.50 -0.10  0.00  -6
  OUTBND_LINE(3)         = 0.10  0.10  0.10  0.00 -10
/




! --------------------------------------------------------------------- !
! WAVEWATCH III - end of namelist                                       !
! --------------------------------------------------------------------- !
```

end of example input file (namelist form)

G.2 ww3_strt

G.2.1 ww3_strt.inp

start of example input file (traditional form)

```
$ --------------------------------------------------------------------- $
$ WAVEWATCH III Initial conditions input file                           $
$---------------------------------------------------------------------- $
$ type of initial field ITYPE .
$
    1
$
$ ITYPE = 1 ---------------------------------------------------------- $
$ Gaussian in frequency and space, cos type in direction.
$ - fp and spread (Hz), mean direction (degr., oceanographic
$    convention) and cosine power, Xm and spread (degr. or m) Ym and
$    spread (degr. or m), Hmax (m) (Example for lon-lat grid in degr.).
$
$   0.10  0.01  270. 2  1. 0.5 1. 0.5 2.5
    0.10  0.01  270. 2  0. 1000. 1. 1000. 2.5
$   0.10  0.01  270. 2  0. 1000. 1. 1000. 0.01
$   0.10  0.01  270. 2  0. 1000. 1. 1000. 0.
```

```
$
$ ITYPE = 2 ---------------------------------------------------------- $
$ JONSWAP spectrum with Hasselmann et al. (1980) direct. distribution.
$ - alfa, peak freq. (Hz), mean direction (degr., oceanographical
$   convention), gamma, sigA, sigB, Xm and spread (degr. or m) Ym and
$   spread (degr. or m)  (Example for lon-lat grid in degr.).
$   alfa, sigA, sigB give default values if less than or equal to 0.
$
$   0.0081  0.1  270.  1.0 0. 0. 1. 100. 1. 100.
$
$ ITYPE = 3 ---------------------------------------------------------- $
$ Fetch-limited JONSWAP
$ - No additional data, the local spectrum is calculated using the
$   local wind speed and direction, using the spatial grid size as
$   fetch, and assuring that the spectrum is within the discrete
$   frequency range.
$
$ ITYPE = 4 ---------------------------------------------------------- $
$ User-defined spectrum
$ - Scale factor., defaults to 1 if less than or equal 0.
$ - Spectrum F(f,theta) (single read statement)
$
$ -0.1
$  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 1 4 2 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 1 2 3 2 1 1 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 1 3 9 7 5 3 2 1 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 1 3 4 3 2 1 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
$  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$
$ ITYPE = 5 ------------------------------------------------------------ $
$ Starting from calm conditions.
$ - No additional data.
$
$ ------------------------------------------------------------------- $
$ End of input file                                                     $
$ ------------------------------------------------------------------- $
```

end of example input file (traditional form)

G.3 ww3_bound

G.3.1 ww3_bound.inp

start of example input file (traditional form)

```
$ ------------------------------------------------------------------- $
$ WAVEWATCH III Ascii boundary input processing                        $
$-------------------------------------------------------------------- $
$
$ Boundary option: READ or WRITE
$
  WRITE
$
$ Interpolation method: 1: nearest
$                       2: linear interpolation
  2
$
$ Verbose mode [0-1]
$
  0
$
$ List of spectra files. These ASCII files use the WAVEWATCH III
$ format as described in the ww3_outp.inp file. The files are
$ defined relative to the directory in which the program is run.
```

```
$
$ Examples of such files can be found at (for example):
$   ftp://polar.ncep.noaa.gov/pub/waves/develop/glw.latest_run/
$                                                (the *.spec.gz files)
$   http://tinyurl.com/iowagaftp/HINDCAST/GLOBAL/2009_ECMWF/SPEC
$
$ If data is used other than from previous WAVEWATCH III runs, then
$ this data will need to be converted to the WAVEWATCH III format.
$
$ In the case of NetCDF files see ww3_bounc.inp
$
SPECTRI/mww3.W004N476.spec
SPECTRI/mww3.W0042N476.spec
SPECTRI/mww3.W0044N476.spec
SPECTRI/mww3.W0046N476.spec
SPECTRI/mww3.W0048N476.spec
SPECTRI/mww3.W005N476.spec
SPECTRI/mww3.W0052N476.spec
SPECTRI/mww3.W0054N476.spec
SPECTRI/mww3.W0056N476.spec
SPECTRI/mww3.W0058N489.spec
SPECTRI/mww3.W006N478.spec
SPECTRI/mww3.W006N482.spec
SPECTRI/mww3.W006N486.spec
SPECTRI/mww3.W006N489.spec
'STOPSTRING'
$
$ ------------------------------------------------------------------- $
$ End of input file                                                   $
$ ------------------------------------------------------------------- $
```

end of example input file (traditional form)

G.4 ww3_bounc

G.4.1 ww3_bounc.inp

start of example input file (traditional form)

```
$ ------------------------------------------------------------------- $
$ WAVEWATCH III NetCDF boundary input processing                       $
$------------------------------------------------------------------- $
$
$ Boundary option: READ or WRITE
$
 WRITE
$
$ Interpolation method: 1: nearest
$                       2: linear interpolation
 2
$ Verbose (0, 1, 2)
1
$
$ List of spectra files. These NetCDF files use the WAVEWATCH III
$ format as described in the ww3_ounp.inp file. The files are
$ defined relative to the directory in which the program is run.
$
SPECTRA_NC/ww3.62163_spec.nc
SPECTRA_NC/ww3.62069_spec.nc
'STOPSTRING'
$
$ ------------------------------------------------------------------- $
$ End of input file                                                   $
$ ------------------------------------------------------------------- $
```

end of example input file (traditional form)

G.4.2 ww3_bounc.nml

start of example input file (namelist form)

```
! ------------------------------------------------------------------- !
! WAVEWATCH III - ww3_bounc.nml - Boundary input post-processing      !
! ------------------------------------------------------------------- !


! ------------------------------------------------------------------- !
! Define the input boundaries to preprocess via BOUND_NML namelist
!
! * namelist must be terminated with /
```

```
! * definitions & defaults:
!     BOUND%MODE                = 'WRITE'           ! ['WRITE'|'READ']
!     BOUND%INTERP              = 2                 ! interpolation [1
!     BOUND%VERBOSE             = 1                 ! [0|1|2]
!     BOUND%FILE                = 'spec.list'       ! input _spec.nc l
! -------------------------------------------------------------------- !
&BOUND_NML
/

! -------------------------------------------------------------------- !
! WAVEWATCH III - end of namelist                                      !
! -------------------------------------------------------------------- !
```

end of example input file (namelist form)

G.5 ww3_prep

G.5.1 ww3_prep.inp

start of example input file (traditional form)

```
$ -------------------------------------------------------------------- $
$ WAVEWATCH III Field preprocessor input file                           $
$ -------------------------------------------------------------------- $
$ Major types of field and time flag
$   Field types  :  IC1   Ice thickness.
$                   IC5   Ice floe mean diameter.
$                   ICE   Ice concentrations.
$                   ISI   Icebergs and sea ice.
$                   LEV   Water levels.
$                   WND   Winds.
$                   WNS   Winds (including air-sea temp. dif.)
$                   CUR   Currents.
$                   DAT   Data for assimilation.
$
$   Format types :  AI    Transfer field 'as is'.
$                   LL    Field defined on rectilinear grid (in same
$                         coordinate system as model grid)
```

```
$                   F1    Field defined on curvilinear grid (in same
$                         coordinate system as model grid), coordinates
$                         of each grid point given in separate file.
$                   F2    Like F1, composite of 2 fields.
$
$       - Format type not used for field type 'DAT'.
$
$   Time flag    : If true, time is included in file.
$   Header flag  : If true, header is added to file.
$                  (necessary for reading, FALSE is used only for
$                   incremental generation of a data file.)
$
   'ICE' 'LL' F T
$
$ Additional time input ------------------------------------------------ $
$ If time flag is .FALSE., give time of field in yyyymmdd hhmmss format.
$
   19680606 053000
$
$ Additional input format type 'LL' ---------------------------------- $
$ Grid range (degr. or m) and number of points for axes, respectively.
$ Example for longitude-latitude grid.
$
    -0.25 2.5  15  -0.25  2.5  4
$
$ Additional input format type 'F1' or 'F2' -------------------------- $
$ Three or four additional input lines, to define the file(s) with
$ the grid information :
$ 1) Discrete size of input grid (NXI,NYI) and T/F flag identifying
$    closure in longitudes ("CLO"). Tripole input is not supported.
$ 2) Define type of file using the parameters FROM, IDLA, IDFM (see
$    input for grid preprocessor), and a format
$ 3) Unit number and (dummy) name of first file.
$ 4) Unit number and (dummy) name of second file (F2 only).
$
$ 15  3
$ 'UNIT' 3 1 '(.L.L.)'
$ 10 'll_file.1'
$ 10 'll_file.2'
$
$ Additional input for data ------------------------------------------- $
$ Dimension of data (0,1,2 for mean pars, 1D or 2D spectra), "record
$ length" for data, data value for missing data
$
$  0  4  -999.
```

```
$
$ Define data files ------------------------------------------------- $
$ The first input line identifies the file format with FROM, IDLA and
$ IDFM, the second (third) lines give the file unit number and name.
$
  'UNIT' 3 1 '(..T..)' '(..F..)'
  10 'data_file.1'
$ 10 'data_file.2'
$
$ If the above unit numbers are 10, data is read from this file
$ (no intermediate comment lines allowed),
$ This example is an ice concentration field.
$
   1. 1. 1. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0.
   1. 1. .5 .5 .5 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
   0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
   0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
$
$ This example is mean parameter assimilation data
$ First record gives number of data records, data are read as as
$ individual records of reals with record length as given above
$
$  3
$ 1.5  1.6 0.70 10.3
$ 1.7  1.5 0.75  9.8
$ 1.9  1.4 0.77 11.1
$
$ ------------------------------------------------------------------- $
$ End of input file                                                   $
$ ------------------------------------------------------------------- $
```

end of example input file (traditional form)

G.6 ww3_prnc

G.6.1 ww3_prnc.inp

start of example input file (traditional form)

```
$ ------------------------------------------------------------------- $
$ WAVEWATCH III Field preprocessor input file                          $
$ ------------------------------------------------------------------- $
$ Major types of field and time flag
$   Field types  :  IC1   Ice thickness.
$                   IC5   Ice floe mean diameter.
$                   ICE   Ice concentrations.
$                   ISI   Icebergs and sea ice.
$                   LEV   Water levels.
$                   WND   Winds.
$                   WNS   Winds (including air-sea temp. dif.)
$                   CUR   Currents.
$                   DAT   Data for assimilation.
$
$   Format types :  AI    Transfer field 'as is'. (ITYPE 1)
$                   LL    Field defined on regular longitude-latitude
$                         or Cartesian grid. (ITYPE 2)
$   Format types :  AT    Transfer field 'as is', performs tidal
$                         analysis on the time series (ITYPE 6)
$                         When using AT, another line should be added
$                         with the choice of tidal constituents:
$                         FAST or VFAST or a list: e.g. 'M2 S2'
$
$       - Format type not used for field type 'DAT'.
$
$   Time flag    : If true, time is included in file.
$   Header flag  : If true, header is added to file.
$                  (necessary for reading, FALSE is used only for
$                   incremental generation of a data file.)
$
  'WND' 'LL' T T
$
$ Name of spatial dimensions----------------------------------------- $
$ NB: time dimension is expected to be called 'time' and must respect
$     Julian or Gregorian calendar with leap day.
$
 longitude latitude
$
$ Variables to use --------------------------------------------------- $
$
  U V
$
$ Additional time input ---------------------------------------------- $
$ If time flag is .FALSE., give time of field in yyyymmdd hhmmss format.
$
```

```
$   19680606 053000
$
$ Define data files ------------------------------------------------- $
$ The input line identifies the filename using for the forcing field.
$
  'wind.nc'
$
$ ------------------------------------------------------------------- $
$ End of input file                                                   $
$ ------------------------------------------------------------------- $
```

end of example input file (traditional form)

G.6.2 ww3_prnc.nml

start of example input file (namelist form)

```
! ------------------------------------------------------------------- !
! WAVEWATCH III - ww3_prnc.nml - Field preprocessor                   !
! ------------------------------------------------------------------- !


! ------------------------------------------------------------------- !
! Define the forcing fields to preprocess via FORCING_NML namelist
!
! * only one FORCING%FIELD can be set at true
! * only one FORCING%grid can be set at true
! * tidal constituents FORCING%tidal is only available on grid%asis with
!
! * namelist must be terminated with /
! * definitions & defaults:
!     FORCING%TIMESTART            = '19000101 000000'  ! Start date for
!     FORCING%TIMESTOP             = '29001231 000000'  ! Stop date for
!
!     FORCING%FIELD%ICE_PARAM1     = f           ! Ice thickness
!     FORCING%FIELD%ICE_PARAM2     = f           ! Ice viscosity
!     FORCING%FIELD%ICE_PARAM3     = f           ! Ice density
!     FORCING%FIELD%ICE_PARAM4     = f           ! Ice modulus
!     FORCING%FIELD%ICE_PARAM5     = f           ! Ice floe mean diamete
!     FORCING%FIELD%MUD_DENSITY    = f           ! Mud density
!     FORCING%FIELD%MUD_THICKNESS  = f           ! Mud thickness
```

```
!     FORCING%FIELD%MUD_VISCOSITY  = f           ! Mud viscosity
!     FORCING%FIELD%WATER_LEVELS   = f           ! Level
!     FORCING%FIELD%CURRENTS       = f           ! Current
!     FORCING%FIELD%WINDS          = f           ! Wind
!     FORCING%FIELD%WIND_AST       = f           ! Wind and air-sea temp
!     FORCING%FIELD%ICE_CONC       = f           ! Ice concentration
!     FORCING%FIELD%ICE_BERG       = f           ! Icebergs and sea ice
!     FORCING%FIELD%DATA_ASSIM     = f           ! Data for assimilation
!
!     FORCING%GRID%ASIS            = f           ! Transfert field 'as i
!     FORCING%GRID%LATLON          = f           ! Define field on regul
!
!     FORCING%TIDAL                = 'unset'     ! Set the tidal constit
! ------------------------------------------------------------------- !
&FORCING_NML
  FORCING%TIMESTART         = '20100101 120000'
  FORCING%TIMESTOP          = '20101231 000000'
  FORCING%FIELD%WINDS   = t
  FORCING%GRID%LATLON   = t
/



! ------------------------------------------------------------------- !
! Define the content of the input file via FILE_NML namelist
!
! * input file must respect netCDF format and CF conventions
! * input file must contain :
!      -dimension : time, name expected to be called time
!      -dimension : longitude/latitude, names can defined in the namelis
!      -variable : time defined along time dimension
!      -attribute : time with attributes units written as ISO8601 conven
!      -attribute : time with attributes calendar set to standard as CF
!      -variable : longitude defined along longitude dimension
!      -variable : latitude defined along latitude dimension
!      -variable : field defined along time,latitude,longitude dimension
! * FILE%VAR(I) must be set for each field component
!
! * namelist must be terminated with /
! * definitions & defaults:
!     FILE%FILENAME      = 'unset'           ! relative path input file
!     FILE%LONGITUDE     = 'unset'           ! longitude/x dimension nam
!     FILE%LATITUDE      = 'unset'           ! latitude/y dimension name
!     FILE%VAR(I)        = 'unset'           ! field component
!     FILE%TIMESHIFT     = '00000000 000000' ! shift the time value to '
```

```
! ------------------------------------------------------------------- !
&FILE_NML
  FILE%FILENAME      = 'wind.nc'
  FILE%LONGITUDE     = 'longitude'
  FILE%LATITUDE      = 'latitude'
  FILE%VAR(1)        = 'U'
  FILE%VAR(2)        = 'V'
/


! ------------------------------------------------------------------- !
! WAVEWATCH III - end of namelist                                     !
! ------------------------------------------------------------------- !
```

end of example input file (namelist form)

G.7 ww3_prtide

G.7.1 ww3_prtide.inp

start of example input file (traditional form)

```
$ ------------------------------------------------------------------- $
$ WAVEWATCH III Field preprocessor input file                          $
$ ------------------------------------------------------------------- $
$ types of field
$   Field types  :  LEV   Water levels.
$                   CUR   Currents.
  'CUR'
$
$ List of tidal constituents----------------------------------------- $
$
 Z0 M2
$
$ Maximum allowed values    ------------------------------------------ $
$ First line: name of tidal constituents for which the max. are defined
$             these should be chosen among the ones avaialable in the
```

```
$             tidal analysis.
$             If analysis was performed with ww3_prnc, the default list
$             is Z0 SSA MSM MSF MF 2N2 MU2 N2 NU2 M2 S2 K2 MSN2 MN4 M4
$                MS4  S4 M6 2MS6 M8
$ Second line: values of maximum magnitude of the amplitude
$ at points where not values are defined or where these maxima are
$ exceeded, the constituents are extrapolated from neighbors
$ (e.g. tidal flats ...)
 Z0   SSA MSF
 1.0  0.5 0.5
$
$  Start time  step        end time
   19680606 000000  1800 19680607 120000
$
$ Define data files ---------------------------------------------------- $
$ The input line identifies the filename using for the forcing field.
$
  'ww3_tide'
$ ---------------------------------------------------------------------- $
$ End of input file                                                      $
$ ---------------------------------------------------------------------- $
```

end of example input file (traditional form)

G.8 ww3_shel

G.8.1 ww3_shel.inp

start of example input file (traditional form)

```
$ ---------------------------------------------------------------------- $
$ WAVEWATCH III shel input file                                          $
$ ---------------------------------------------------------------------- $
$ Define input to be used with F/T/C flag for use or nor or coupling and
$ T/F flag for definition as a homogeneous field.
$
$ Include ice and mud parameters only if IC1/2/3/4 used :
   F F     Ice parameter 1
```

```
   F F     Ice parameter 2
   F F     Ice parameter 3
   F F     Ice parameter 4
   F F     Ice parameter 5
   F F     Mud parameter 1
   F F     Mud parameter 2
   F F     Mud parameter 3
   F F     Water levels
   F F     Currents
   T T     Winds
   T       Ice concentrations
   F       Assimilation data : Mean parameters
   F       Assimilation data : 1-D spectra
   F       Assimilation data : 2-D spectra
$
$ Time frame of calculations ------------------------------------------ $
$ - Starting time in yyyymmdd hhmmss format.
$ - Ending time in yyyymmdd hhmmss format.
$
   19680606 000000
   19680606 060000
$
$ Define output data --------------------------------------------------- $
$
$ Define output server mode. This is used only in the parallel version
$ of the model. To keep the input file consistent, it is always needed.
$ IOSTYP = 1 is generally recommended. IOSTYP > 2 may be more efficient
$ for massively parallel computations. Only IOSTYP = 0 requires a true
$ parallel file system like GPFS.
$
$    IOSTYP = 0 : No data server processes, direct access output from
$                 each process (requires true parallel file system).
$             1 : No data server process. All output for each type
$                 performed by process that performs computations too.
$             2 : Last process is reserved for all output, and does no
$                 computing.
$             3 : Multiple dedicated output processes.
$
   2
$
$ 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 logical flags to activate output
$          fields as defined in section 2.4 of the manual. The logical
$          flags are not supplied if no output is requested. The logical
$          flags can be placed on multiple consecutive lines. However,
$          the total number and order of the logical flags is fixed.
$                               The raw data file is out_grd.ww3,
$                               see w3iogo.ftn for additional doc.
$
   19680606 000000   3600  19680608 000000
$-------------------------------------------------------------------
$ Output request flags identifying fields.
$
$ The table below provides a full definition of field output parameters
$  as well as flags indicating if they are available in different field
$  output output file types (ASCII, grib, NetCDF).
$ Further definitions are found in section 2.4 of the manual.
$
$ Selection of field outputs may be made in two ways:
$   F/T flags: first flag is set to F, requests made per group (1st line
$              followed by parameter flags (total of 10 groups).
$   Namelists: first line is set to N, next line contains parameter
$              symbol as per table below.
$
$   Example of F/T flag use is given in this sample ww3_shel.inp, below.
$    For namelist usage, see the sample ww3_ounf.inp for an example.
$
$ ----------------------------------------
$ Output field parameter definitions table
$ ----------------------------------------
$
$ All parameters listed below are available in output file of the types
$  ASCII and NetCDF. If selected output file types are grads or grib,
$  some parameters may not be available. The first two columns in the
$  table below identify such cases by flags, cols 1 (GRB) and 2 (GXO)
$  refer to grib (ww3_grib) and grads (gx_outf), respectively.
$
$ Columns 3 and 4 provide group and parameter numbers per group.
$ Columns 5, 6 and 7 provide:
$   5 - code name (internal)
$   6 - output tags (names used is ASCII file extensions, NetCDF
$       variable names and namelist-based selection (see ww3_ounf.inp)
```

```
$   7 - Long parameter name/definition
$
$  G  G
$  R  X Grp  Param Code     Output  Parameter/Group
$  B  O Numb Numbr Name        Tag  Definition
$  ------------------------------------------------
$        1                         Forcing Fields
$   -----------------------------------------------
$  T  T  1     1   DW         DPT   Water depth.
$  T  T  1     2   C[X,Y]     CUR   Current velocity.
$  T  T  1     3   UA         WND   Wind speed.
$  T  T  1     4   AS         AST   Air-sea temperature difference.
$  T  T  1     5   WLV        WLV   Water levels.
$  T  T  1     6   ICE        ICE   Ice concentration.
$  T  T  1     7   IBG        IBG   Iceberg-induced damping.
$  T  T  1     8   D50        D50   Median sediment grain size.
$  T  T  1     9   IC1        IC1   Ice thickness.
$  T  T  1    10   IC5        IC5   Ice flow diameter.
$   -----------------------------------------------
$        2                         Standard mean wave Parameters
$   -----------------------------------------------
$  T  T  2     1   HS         HS    Wave height.
$  T  T  2     2   WLM        LM    Mean wave length.
$  T  T  2     3   T02        T02   Mean wave period (Tm0,2).
$  T  T  2     4   T0M1       T0M1  Mean wave period (Tm0,-1).
$  T  T  2     5   T01        T01   Mean wave period (Tm0,1).
$  T  T  2     6   FP0        FP    Peak frequency.
$  T  T  2     7   THM        DIR   Mean wave direction.
$  T  T  2     8   THS        SPR   Mean directional spread.
$  T  T  2     9   THP0       DP    Peak direction.
$  T  T  2    10   HIG        HIG   Infragravity height
$  T  T  2    11   STMAXE     MXE   Max surface elev (STE)
$  T  T  2    12   STMAXD     MXES  St Dev of max surface elev (STE)
$  T  T  2    13   HMAXE      MXH   Max wave height (STE)
$  T  T  2    14   HCMAXE     MXHC  Max wave height from crest (STE)
$  T  T  2    15   HMAXD      SDMH  St Dev of MXC (STE)
$  T  T  2    16   HCMAXD     SDMHC St Dev of MXHC (STE)
$  F  T  2    17   WBT        WBT   Dominant wave breaking probability b
$   -----------------------------------------------
$        3                         Spectral Parameters (first 5)
$   -----------------------------------------------
$  F  F  3     1   EF         EF    Wave frequency spectrum
$  F  F  3     2   TH1M       TH1M  Mean wave direction from a1,b2
$  F  F  3     3   STH1M      STH1M Directional spreading from a1,b2
$  F  F  3     4   TH2M       TH2M  Mean wave direction from a2,b2
```

```
$  F  F  3     5   STH2M      STH2M Directional spreading from a2,b2
$  F  F  3     6   WN         WN    Wavenumber array
$   -------------------------------------------------
$        4                           Spectral Partition Parameters
$   -------------------------------------------------
$  T  T  4     1   PHS        PHS   Partitioned wave heights.
$  T  T  4     2   PTP        PTP   Partitioned peak period.
$  T  T  4     3   PLP        PLP   Partitioned peak wave length.
$  T  T  4     4   PDIR       PDIR  Partitioned mean direction.
$  T  T  4     5   PSI        PSPR  Partitioned mean directional spread.
$  T  T  4     6   PWS        PWS   Partitioned wind sea fraction.
$  T  T  4     7   PTHP0      PDP   Peak wave direction of partition.
$  T  T  4     8   PQP        PQP   Goda peakdedness parameter of partit
$  T  T  4     9   PPE        PPE   JONSWAP peak enhancement factor of p
$  T  T  4    10   PGW        PGW   Gaussian frequency width of partitio
$  T  T  4    11   PSW        PSW   Spectral width of partition.
$  T  T  4    12   PTM1       PTM10 Mean wave period (m-1,0) of partitio
$  T  T  4    13   PT1        PT01  Mean wave period (m0,1) of partition
$  T  T  4    14   PT2        PT02  Mean wave period (m0,2) of partition
$  T  T  4    15   PEP        PEP   Peak spectral density of partition.
$  T  T  4    16   PWST       TWS   Total wind sea fraction.
$  T  T  4    17   PNR        PNR   Number of partitions.
$   -------------------------------------------------
$        5                           Atmosphere-waves layer
$   -------------------------------------------------
$  T  T  5     1   UST        UST   Friction velocity.
$  F  T  5     2   CHARN      CHA   Charnock parameter
$  F  T  5     3   CGE        CGE   Energy flux
$  F  T  5     4   PHIAW      FAW   Air-sea energy flux
$  F  T  5     5   TAUWI[X,Y] TAW   Net wave-supported stress
$  F  T  5     6   TAUWN[X,Y] TWA   Negative part of the wave-supported
$  F  F  5     7   WHITECAP   WCC   Whitecap coverage
$  F  F  5     8   WHITECAP   WCF   Whitecap thickness
$  F  F  5     9   WHITECAP   WCH   Mean breaking height
$  F  F  5    10   WHITECAP   WCM   Whitecap moment
$  F  F  5    11   FWS        FWS   Wind sea mean period
$   -------------------------------------------------
$        6                           Wave-ocean layer
$   -------------------------------------------------
$  F  F  6     1   S[XX,YY,XY] SXY  Radiation stresses.
$  F  F  6     2   TAUO[X,Y]  TWO   Wave to ocean momentum flux
$  F  F  6     3   BHD        BHD   Bernoulli head (J term)
$  F  F  6     4   PHIOC      FOC   Wave to ocean energy flux
$  F  F  6     5   TUS[X,Y]   TUS   Stokes transport
$  F  F  6     6   USS[X,Y]   USS   Surface Stokes drift
```

```
$  F  F  6     7   [PR,TP]MS  P2S   Second-order sum pressure
$  F  F  6     8   US3D       USF   Spectrum of surface Stokes drift
$  F  F  6     9   P2SMS      P2L   Micro seism  source term
$  F  F  6    10   TAUICE     TWI   Wave to sea ice stress
$  F  F  6    11   PHICE      FIC   Wave to sea ice energy flux
$  F  F  6    12   USSP       USP   Partitioned surface Stokes drift
$   ------------------------------------------------
$        7                           Wave-bottom layer
$   ------------------------------------------------
$  F  F  7     1   ABA        ABR   Near bottom rms amplitides.
$  F  F  7     2   UBA        UBR   Near bottom rms velocities.
$  F  F  7     3   BEDFORMS   BED   Bedforms
$  F  F  7     4   PHIBBL     FBB   Energy flux due to bottom friction
$  F  F  7     5   TAUBBL     TBB   Momentum flux due to bottom friction
$   ------------------------------------------------
$        8                           Spectrum parameters
$   ------------------------------------------------
$  F  F  8     1   MSS[X,Y]   MSS   Mean square slopes
$  F  F  8     2   MSC[X,Y]   MSC   Spectral level at high frequency tai
$  F  F  8     3   WL02[X,Y]  WL02  East/X North/Y mean wavelength compo
$  F  F  8     4   ALPXT      AXT   Correl sea surface gradients (x,t)
$  F  F  8     5   ALPYT      AYT   Correl sea surface gradients (y,t)
$  F  F  8     6   ALPXY      AXY   Correl sea surface gradients (x,y)
$   ------------------------------------------------
$        9                           Numerical diagnostics
$   ------------------------------------------------
$  T  T  9     1   DTDYN      DTD   Average time step in integration.
$  T  T  9     2   FCUT       FC    Cut-off frequency.
$  T  T  9     3   CFLXYMAX   CFX   Max. CFL number for spatial advectio
$  T  T  9     4   CFLTHMAX   CFD   Max. CFL number for theta-advection.
$  F  F  9     5   CFLKMAX    CFK   Max. CFL number for k-advection.
$   ------------------------------------------------
$        10                          User defined
$   ------------------------------------------------
$  F  F  10    1              U1    User defined #1. (requires coding ..
$  F  F  10    2              U2    User defined #1. (requires coding ..
$   ------------------------------------------------
$
$     Section 4 consist of a set of fields, index 0 = wind sea, index
$     1:NOSWLL are first NOSWLL swell fields.
$
$ Actual active parameter selection section
$
$ (1) Forcing Fields
  T
```

```
$ DPT CUR WND AST WLV ICE IBG D50 IC1 IC5
  T   T   T   T   T   F   F   F   F   F
$ (2) Standard mean wave Parameters
  T
$ HS  LM  T02 T0M1 T01 FP DIR SPR DP
  T   T   T   T   T   T   T   T   T
$ (3) Frequency-dependent parameters
  T
$ EF TH1M STH1M TH2M STH2M WN
  T   T   T   F   F   F
$ (4) Spectral Partition Parameters
  T
$ PHS PTP PLP PDIR PSPR PNR PDP PQP PPE PGW PSW PTM10 PT01 PT02 PEP PWS
  T   T   T   T    T    T   T   T   T   T   T   T     T    T    T   T
$ (5) Atmosphere-waves layer
  T
$ UST CHA CGE FAW TAW TWA WCC WCF WCH WCM FWS
  T   T   T   T   T   T   T   T   T   T   T
$ (6) Wave-Ocean layer
  T
$ SXY TWO BHD FOC TUS USS P2S USF P2L TWI FIC USP
  T   T   T   T   T   T   T   F   F   F   F   T
$ (7) Wave-bottom layer
  T
$ ABR UBR BED FBB TBB
  T   T   T   T   T
$ (8) Spectrum parameters
  T
$ MSS MSC WL02 AXT AYT AXY
  T   T   T   T   T   T
$ (9) Numerical diagnostics
  T
$ DTD FC  CFX CFD CFK
  T   T   T   T   T
$ (10) User defined (NOEXTR flags needed)
  F
$  U1  U2
$ T   T
$
$-------------------------------------------------------------------
$
$ Type 2 : Point output
$          Standard line and a number of lines identifying the
$          longitude, latitude and name (C*10) of output points.
$          The list is closed by defining a point with the name
```

```
$          'STOPSTRING'. No point info read if no point output is
$          requested (i.e., no 'STOPSTRING' needed).
$          Example for spherical grid.
$                               The raw data file is out_pnt.ww3,
$                               see w3iogo.ftn for additional doc.
$
$   NOTE : Spaces may be included in the name, but this is not
$          advised, because it will break the GrADS utility to
$          plots spectra and source terms, and will make it more
$          difficult to use point names in data files.
$
   19680606 000000    900  19680608 000000
$
    -0.25 -0.25 'Land      '
     0.0   0.0  'Point_1   '
     2.0   1.0  'Point_2   '
     1.8   2.2  'Point_3   '
     2.1   0.9  'Point_4   '
     5.0   5.0  'Outside   '
$
     0.0   0.0  'STOPSTRING'
$
$ Type 3 : Output along track.
$          Flag for formatted input file.
$                         The data files are track_i.ww3 and
$                         track_o.ww3, see w3iotr.ftn for ad. doc.
$
   19680606 000000   1800  19680606 013000
     T
$
$ Type 4 : Restart files (no additional data required).
$                               The data file is restartN.ww3, see
$                               w3iors.ftn for additional doc.
$
   19680606 030000  3600  19680607 030000
$
$ Type 5 : Boundary data (no additional data required).
$                               The data file is nestN.ww3, see
$                               w3iobcmd.ftn for additional doc.
$
   19680606 000000   3600  20010102 000000
$
$ Type 6 : Separated wave field data (dummy for now).
$          First, last step IX and IY, flag for formatted file
$
```

```
   19680606 000000   3600  20010102 000000
      0 999 1 0 999 1 T
$
$ Type 7 : Coupling. (must be fully commented if not used with switch CO
$          Namelist type selection is used here.
$          Diagnostic fields to exchange. (see namcouple for more inform
$
$  19680606 000000   3600  20010102 000000
$  N
$
$   - Sent fields by ww3:
$       - Ocean model : T0M1 OCHA OHS DIR BHD TWO UBR FOC TAW TUS USS LM
$       - Atmospheric model : ACHA AHS TP (or FP) FWS
$       - Ice model : IC5 TWI
$
$  CHA
$
$   - Received fields by ww3:
$       - Ocean model : SSH CUR
$       - Atmospheric model : WND
$       - Ice model : ICE IC1 IC5
$
$  WND
$
$ 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, oceanogr.
$ convention degrees)) and air-sea temperature difference (degrees C).
$ 'STP' is mandatory stop string.
$ Also defined here are the speed with which the grid is moved
$ continuously, ID string 'MOV', parameters as for 'CUR'.
$
   'LEV' 19680606 010000    1.00
   'CUR' 19680606 073125    2.0    25.
   'WND' 19680606 000000   20.    145.    2.0
   'MOV' 19680606 013000    4.0    25.
   'STP'
$
$ ------------------------------------------------------------------- $
$ End of input file                                                   $
$ ------------------------------------------------------------------- $
```

end of example input file (traditional form)

G.8.2 ww3_shel.nml

start of example input file (namelist form)

```
! ------------------------------------------------------------------- !
! WAVEWATCH III - ww3_shel.nml - single-grid model                    !
! ------------------------------------------------------------------- !


! ------------------------------------------------------------------- !
! Define top-level model parameters via DOMAIN_NML namelist
!
! * IOSTYP defines the output server mode for parallel implementation.
!             0 : No data server processes, direct access output from
!                 each process (requires true parallel file system).
!             1 : No data server process. All output for each type
!                 performed by process that performs computations too.
!             2 : Last process is reserved for all output, and does no
!                 computing.
!             3 : Multiple dedicated output processes.
!
! * namelist must be terminated with /
! * definitions & defaults:
!     DOMAIN%IOSTYP =  1                 ! Output server type
!     DOMAIN%START  = '19680606 000000'  ! Start date for the entire mod
!     DOMAIN%STOP   = '19680607 000000'  ! Stop date for the entire mode
! ------------------------------------------------------------------- !
&DOMAIN_NML
  DOMAIN%START  = '20100101 120000'
  DOMAIN%STOP   = '20101231 000000'
/



! ------------------------------------------------------------------- !
! Define each forcing via the INPUT_NML namelist
!
! * The FORCING flag can be  : 'F' for "no forcing"
!                              'T' for "external forcing file"
!                              'H' for "homogeneous forcing input"
!                              'C' for "coupled forcing field"
!
! * homogeneous forcing is not available for ICE_CONC
!
```

```
! * The ASSIM flag can :  'F' for "no forcing"
!                         'T' for "external forcing file"
!
! * namelist must be terminated with /
! * definitions & defaults:
!     INPUT%FORCING%WATER_LEVELS  = 'F'
!     INPUT%FORCING%CURRENTS      = 'F'
!     INPUT%FORCING%WINDS         = 'F'
!     INPUT%FORCING%ICE_CONC      = 'F'
!     INPUT%FORCING%ICE_PARAM1    = 'F'
!     INPUT%FORCING%ICE_PARAM2    = 'F'
!     INPUT%FORCING%ICE_PARAM3    = 'F'
!     INPUT%FORCING%ICE_PARAM4    = 'F'
!     INPUT%FORCING%ICE_PARAM5    = 'F'
!     INPUT%FORCING%MUD_DENSITY   = 'F'
!     INPUT%FORCING%MUD_THICKNESS = 'F'
!     INPUT%FORCING%MUD_VISCOSITY = 'F'
!     INPUT%ASSIM%MEAN            = 'F'
!     INPUT%ASSIM%SPEC1D          = 'F'
!     INPUT%ASSIM%SPEC2D          = 'F'
! -------------------------------------------------------------------- !
&INPUT_NML
  INPUT%FORCING%WINDS = 'T'
  INPUT%FORCING%WATER_LEVELS  = 'C'
  INPUT%FORCING%CURRENTS      = 'C'
/




! -------------------------------------------------------------------- !
! Define the output types point parameters via OUTPUT_TYPE_NML namelist
!
! * the point file is a space separated values per line : lon lat 'name'
!
! * the full list of field names is :
!  All parameters listed below are available in output file of the types
!  ASCII and NetCDF. If selected output file types are grads or grib,
!  some parameters may not be available. The first two columns in the
!  table below identify such cases by flags, cols 1 (GRB) and 2 (GXO)
!  refer to grib (ww3_grib) and grads (gx_outf), respectively.
!
! Columns 3 and 4 provide group and parameter numbers per group.
! Columns 5, 6 and 7 provide:
!   5 - code name (internal)
```

```
!   6 - output tags (names used is ASCII file extensions, NetCDF
!       variable names and namelist-based selection
!   7 - Long parameter name/definition
!
!  G  G
!  R  X Grp  Param Code     Output  Parameter/Group
!  B  O Numb Numbr Name        Tag  Definition
!  ------------------------------------------------
!        1                          Forcing Fields
!   -----------------------------------------------
!  T  T  1     1   DW         DPT   Water depth.
!  T  T  1     2   C[X,Y]     CUR   Current velocity.
!  T  T  1     3   UA         WND   Wind speed.
!  T  T  1     4   AS         AST   Air-sea temperature difference.
!  T  T  1     5   WLV        WLV   Water levels.
!  T  T  1     6   ICE        ICE   Ice concentration.
!  T  T  1     7   IBG        IBG   Iceberg-induced damping.
!  T  T  1     8   D50        D50   Median sediment grain size.
!  T  T  1     9   IC1        IC1   Ice thickness.
!  T  T  1    10   IC5        IC5   Ice flow diameter.
!   -----------------------------------------------
!        2                          Standard mean wave Parameters
!   -----------------------------------------------
!  T  T  2     1   HS         HS    Wave height.
!  T  T  2     2   WLM        LM    Mean wave length.
!  T  T  2     3   T02        T02   Mean wave period (Tm0,2).
!  T  T  2     4   T0M1       T0M1  Mean wave period (Tm0,-1).
!  T  T  2     5   T01        T01   Mean wave period (Tm0,1).
!  T  T  2     6   FP0        FP    Peak frequency.
!  T  T  2     7   THM        DIR   Mean wave direction.
!  T  T  2     8   THS        SPR   Mean directional spread.
!  T  T  2     9   THP0       DP    Peak direction.
!  T  T  2    10   HIG        HIG   Infragravity height
!  T  T  2    11   STMAXE     MXE   Max surface elev (STE)
!  T  T  2    12   STMAXD     MXES  St Dev of max surface elev (STE)
!  T  T  2    13   HMAXE      MXH   Max wave height (STE)
!  T  T  2    14   HCMAXE     MXHC  Max wave height from crest (STE)
!  T  T  2    15   HMAXD      SDMH  St Dev of MXC (STE)
!  T  T  2    16   HCMAXD     SDMHC St Dev of MXHC (STE)
!  F  T  2    17   WBT        WBT   Dominant wave breaking probability b
!   -----------------------------------------------
!        3                          Spectral Parameters (first 5)
!   -----------------------------------------------
!  F  F  3     1   EF         EF    Wave frequency spectrum
!  F  F  3     2   TH1M       TH1M  Mean wave direction from a1,b2
```

```
!  F  F  3     3   STH1M      STH1M Directional spreading from a1,b2
!  F  F  3     4   TH2M       TH2M  Mean wave direction from a2,b2
!  F  F  3     5   STH2M      STH2M Directional spreading from a2,b2
!  F  F  3     6   WN         WN    Wavenumber array
!   -------------------------------------------------
!        4                           Spectral Partition Parameters
!   -------------------------------------------------
!  T  T  4     1   PHS        PHS   Partitioned wave heights.
!  T  T  4     2   PTP        PTP   Partitioned peak period.
!  T  T  4     3   PLP        PLP   Partitioned peak wave length.
!  T  T  4     4   PDIR       PDIR  Partitioned mean direction.
!  T  T  4     5   PSI        PSPR  Partitioned mean directional spread.
!  T  T  4     6   PWS        PWS   Partitioned wind sea fraction.
!  T  T  4     7   PDP        PDP   Peak wave direction of partition.
!  T  T  4     8   PQP        PQP   Goda peakdedness parameter of partit
!  T  T  4     9   PPE        PPE   JONSWAP peak enhancement factor of p
!  T  T  4    10   PGW        PGW   Gaussian frequency width of partitio
!  T  T  4    11   PSW        PSW   Spectral width of partition.
!  T  T  4    12   PTM1       PTM10 Mean wave period (m-1,0) of partitio
!  T  T  4    13   PT1        PT01  Mean wave period (m0,1) of partition
!  T  T  4    14   PT2        PT02  Mean wave period (m0,2) of partition
!  T  T  4    15   PEP        PEP   Peak spectral density of partition.
!  T  T  4    16   PWST       TWS   Total wind sea fraction.
!  T  T  4    17   PNR        PNR   Number of partitions.
!   -------------------------------------------------
!        5                           Atmosphere-waves layer
!   -------------------------------------------------
!  T  T  5     1   UST        UST   Friction velocity.
!  F  T  5     2   CHARN      CHA   Charnock parameter
!  F  T  5     3   CGE        CGE   Energy flux
!  F  T  5     4   PHIAW      FAW   Air-sea energy flux
!  F  T  5     5   TAUWI[X,Y] TAW   Net wave-supported stress
!  F  T  5     6   TAUWN[X,Y] TWA   Negative part of the wave-supported
!  F  F  5     7   WHITECAP   WCC   Whitecap coverage
!  F  F  5     8   WHITECAP   WCF   Whitecap thickness
!  F  F  5     9   WHITECAP   WCH   Mean breaking height
!  F  F  5    10   WHITECAP   WCM   Whitecap moment
!  F  F  5    11   FWS        FWS   Wind sea mean period
!   -------------------------------------------------
!        6                           Wave-ocean layer
!   -------------------------------------------------
!  F  F  6     1   S[XX,YY,XY] SXY  Radiation stresses.
!  F  F  6     2   TAUO[X,Y]  TWO   Wave to ocean momentum flux
!  F  F  6     3   BHD        BHD   Bernoulli head (J term)
!  F  F  6     4   PHIOC      FOC   Wave to ocean energy flux
```

```
!  F  F  6     5   TUS[X,Y]   TUS   Stokes transport
!  F  F  6     6   USS[X,Y]   USS   Surface Stokes drift
!  F  F  6     7   [PR,TP]MS  P2S   Second-order sum pressure
!  F  F  6     8   US3D       USF   Spectrum of surface Stokes drift
!  F  F  6     9   P2SMS      P2L   Micro seism  source term
!  F  F  6    10   TAUICE     TWI   Wave to sea ice stress
!  F  F  6    11   PHICE      FIC   Wave to sea ice energy flux
!  F  F  6    12   USSP       USP   Partitioned surface Stokes drift
!   -----------------------------------------------
!        7                           Wave-bottom layer
!   -----------------------------------------------
!  F  F  7     1   ABA        ABR   Near bottom rms amplitides.
!  F  F  7     2   UBA        UBR   Near bottom rms velocities.
!  F  F  7     3   BEDFORMS   BED   Bedforms
!  F  F  7     4   PHIBBL     FBB   Energy flux due to bottom friction
!  F  F  7     5   TAUBBL     TBB   Momentum flux due to bottom friction
!   -----------------------------------------------
!        8                           Spectrum parameters
!   -----------------------------------------------
!  F  F  8     1   MSS[X,Y]   MSS   Mean square slopes
!  F  F  8     2   MSC[X,Y]   MSC   Spectral level at high frequency tai
!  F  F  8     3   WL02[X,Y]  WL02  East/X North/Y mean wavelength compo
!  F  F  8     4   ALPXT      AXT   Correl sea surface gradients (x,t)
!  F  F  8     5   ALPYT      AYT   Correl sea surface gradients (y,t)
!  F  F  8     6   ALPXY      AXY   Correl sea surface gradients (x,y)
!   -----------------------------------------------
!        9                           Numerical diagnostics
!   -----------------------------------------------
!  T  T  9     1   DTDYN      DTD   Average time step in integration.
!  T  T  9     2   FCUT       FC    Cut-off frequency.
!  T  T  9     3   CFLXYMAX   CFX   Max. CFL number for spatial advectio
!  T  T  9     4   CFLTHMAX   CFD   Max. CFL number for theta-advection.
!  F  F  9     5   CFLKMAX    CFK   Max. CFL number for k-advection.
!   -----------------------------------------------
!        10                          User defined
!   -----------------------------------------------
!  F  F  10    1              U1    User defined #1. (requires coding ..
!  F  F  10    2              U2    User defined #1. (requires coding ..
!   -----------------------------------------------
!
!     Section 4 consist of a set of fields, index 0 = wind sea, index
!     1:NOSWLL are first NOSWLL swell fields.
!
!
! * output track file formatted (T) or unformated (F)
```

```
!
! * coupling fields exchanged list is :
!   - Sent fields by ww3:
!       - Ocean model : TOM1 OCHA OHS DIR BHD TWO UBR FOC TAW TUS USS LM
!       - Atmospheric model : ACHA AHS TP (or FP) FWS
!       - Ice model : IC5 TWI
!   - Received fields by ww3:
!       - Ocean model : SSH CUR
!       - Atmospheric model : WND
!       - Ice model : ICE IC1 IC5
!
! * namelist must be terminated with /
! * definitions & defaults:
!     TYPE%FIELD%LIST         =  'unset'
!     TYPE%POINT%FILE         =  'points.list'
!     TYPE%TRACK%FORMAT       =  T
!     TYPE%PARTITION%X0       =  0
!     TYPE%PARTITION%XN       =  0
!     TYPE%PARTITION%NX       =  0
!     TYPE%PARTITION%Y0       =  0
!     TYPE%PARTITION%YN       =  0
!     TYPE%PARTITION%NY       =  0
!     TYPE%PARTITION%FORMAT   =  T
!     TYPE%COUPLING%SENT      = 'unset'
!     TYPE%COUPLING%RECEIVED  = 'unset'
!
! ------------------------------------------------------------------- !
&OUTPUT_TYPE_NML
  TYPE%FIELD%LIST     = 'HS DIR SPR WND ICE CUR LEV'
/



! ------------------------------------------------------------------- !
! Define output dates via OUTPUT_DATE_NML namelist
!
! * start and stop times are with format 'yyyymmdd hhmmss'
! * if time stride is equal '0', then output is disabled
! * time stride is given in seconds
!
! * namelist must be terminated with /
! * definitions & defaults:
!     DATE%FIELD%START         =  '19680606 000000'
!     DATE%FIELD%STRIDE        =  '0'
!     DATE%FIELD%STOP          =  '19680607 000000'
```

```
!     DATE%POINT%START         =  '19680606 000000'
!     DATE%POINT%STRIDE        =  '0'
!     DATE%POINT%STOP          =  '19680607 000000'
!     DATE%TRACK%START         =  '19680606 000000'
!     DATE%TRACK%STRIDE        =  '0'
!     DATE%TRACK%STOP          =  '19680607 000000'
!     DATE%RESTART%START       =  '19680606 000000'
!     DATE%RESTART%STRIDE      =  '0'
!     DATE%RESTART%STOP        =  '19680607 000000'
!     DATE%BOUNDARY%START      =  '19680606 000000'
!     DATE%BOUNDARY%STRIDE     =  '0'
!     DATE%BOUNDARY%STOP       =  '19680607 000000'
!     DATE%PARTITION%START     =  '19680606 000000'
!     DATE%PARTITION%STRIDE    =  '0'
!     DATE%PARTITION%STOP      =  '19680607 000000'
!     DATE%COUPLING%START      =  '19680606 000000'
!     DATE%COUPLING%STRIDE     =  '0'
!     DATE%COUPLING%STOP       =  '19680607 000000'
!
!     DATE%RESTART             =  '19680606 000000' '0' '19680607 000000
! ------------------------------------------------------------------- !
&OUTPUT_DATE_NML
  DATE%FIELD%START         = '20100101 000000'
  DATE%FIELD%STRIDE        = '3600'
  DATE%FIELD%STOP          = '20101231 000000'
  DATE%POINT%START         = '20100101 000000'
  DATE%POINT%STRIDE        = '3600'
  DATE%POINT%STOP          = '20101231 000000'

  DATE%RESTART             = '20101231 000000' '43200' '20501231 000000'
/




! ------------------------------------------------------------------- !
! Define homogeneous input via HOMOG_COUNT_NML and HOMOG_INPUT_NML namel
!
! * the number of each homogeneous input is defined by HOMOG_COUNT
! * the total number of homogeneous input is automatically calculated
! * the homogeneous input must start from index 1 to N
! * if VALUE1 is equal 0, then the homogeneous input is desactivated
! * NAME can be IC1, IC2, IC3, IC4, IC5, MDN, MTH, MVS, LEV, CUR, WND, I
! * each homogeneous input is defined over a maximum of 3 values detaill
!     - IC1 is defined by thickness
```

```
!     - IC2 is defined by viscosity
!     - IC3 is defined by density
!     - IC4 is defined by modulus
!     - IC5 is defined by floe diameter
!     - MDN is defined by density
!     - MTH is defined by thickness
!     - MVS is defined by viscosity
!     - LEV is defined by height
!     - CUR is defined by speed and direction
!     - WND is defined by speed, direction and airseatemp
!     - ICE is defined by concentration
!     - MOV is defined by speed and direction
!
! * namelist must be terminated with /
! * definitions & defaults:
!     HOMOG_COUNT%N_IC1            =  0
!     HOMOG_COUNT%N_IC2            =  0
!     HOMOG_COUNT%N_IC3            =  0
!     HOMOG_COUNT%N_IC4            =  0
!     HOMOG_COUNT%N_IC5            =  0
!     HOMOG_COUNT%N_MDN            =  0
!     HOMOG_COUNT%N_MTH            =  0
!     HOMOG_COUNT%N_MVS            =  0
!     HOMOG_COUNT%N_LEV            =  0
!     HOMOG_COUNT%N_CUR            =  0
!     HOMOG_COUNT%N_WND            =  0
!     HOMOG_COUNT%N_ICE            =  0
!     HOMOG_COUNT%N_MOV            =  0
!
!     HOMOG_INPUT(I)%NAME           =  'unset'
!     HOMOG_INPUT(I)%DATE           =  '19680606 000000'
!     HOMOG_INPUT(I)%VALUE1         =  0
!     HOMOG_INPUT(I)%VALUE2         =  0
!     HOMOG_INPUT(I)%VALUE3         =  0
! ------------------------------------------------------------------- !
&HOMOG_COUNT_NML
  HOMOG_COUNT%N_WND        = 2
  HOMOG_COUNT%N_LEV        = 1
/

&HOMOG_INPUT_NML
  HOMOG_INPUT(1)%NAME       = 'WND'
  HOMOG_INPUT(1)%DATE       ='20100610 000000'
  HOMOG_INPUT(1)%VALUE1     = 5.
  HOMOG_INPUT(1)%VALUE2     = 90.
```

```
  HOMOG_INPUT(2)%NAME       = 'WND'
  HOMOG_INPUT(2)%DATE       = '20100610 060000'
  HOMOG_INPUT(2)%VALUE1     = 25.
  HOMOG_INPUT(2)%VALUE2     = 120.

  HOMOG_INPUT(3)%NAME       = 'LEV'
  HOMOG_INPUT(3)%DATE       = '20100610 060000'
  HOMOG_INPUT(3)%VALUE1     = 5.
/


! -------------------------------------------------------------------- !
! WAVEWATCH III - end of namelist                                      !
! -------------------------------------------------------------------- !
```

end of example input file (namelist form)

G.9 ww3_gspl

G.9.1 ww3_gspl.inp

start of example input file (traditional form)

```
$ -------------------------------------------------------------------- $
$ WAVEWATCH III Grid splitting input file                              $
$ -------------------------------------------------------------------- $
$ Grid identifier (file extension for mod_def file of grid to be split)
$
  'glo_2d'
$
$ Number of sub-grids to be created, maximum number of iterations,
```

```
$ target grid point count std in percent. user defined halo extension
$ (default should be 2, used because of inconsistencies between halo
$ computation in this code and in the main wave model code). Increase
$ the latter number if ww3_multi fails on halo overlaps between
$ equally ranked grids.
$
  12  250  0.75  2
$
$ IDLA, IDFM, scale and RFORM for bottom, obstruction and mask files.
$ Note that the third file is integers. Suggest IDFM = 1 and IDLA = 1
$
   3   2   1.0   '(12F11.3)'
   3   2   1.0   '(26F5.2)'
   3   2   1     '(66I2)'
$
$ lowest and highest fraction of communicator to be used for grid.
$ and flag for running grids side-by-side inside fraction
$  F: for test purposes only, defeats most reasons for splitting
$  T: normal operations
$
   0.4  1.  F
$ -------------------------------------------------------------------- $
$ End of input file                                                    $
$ -------------------------------------------------------------------- $
```

end of example input file (traditional form)

G.10 ww3_multi

G.10.1 ww3_multi.inp

start of example input file (traditional form)

```
$ -------------------------------------------------------------------- $
$ WAVEWATCH III multi-grid model driver input file                     $
$ -------------------------------------------------------------------- $
$
$ ********************************************************************
```

```
$ *** NOTE : This is an example file from the mww3_test_05 script ***
$ ***        Unlike other input example files this one CANNOT     ***
$ ***        be run as an independent interactive run             ***
$ ************************************************************************
$
$ The first input line sets up the general multi-grid model definition
$ by defining the following six parameters :
$
$   1) Number of wave model grids.                          ( NRGRD )
$   2) Number of grids defining input fields.               ( NRINP )
$   3) Flag for using unified point output file.           ( UNIPTS )
$   4) Output server type as in ww3_shel.inp
$   5) Flag for dedicated process for unified point output.
$   6) Flag for grids sharing dedicated output processes.
$
  3 1 T 1 T T
$
$ ------------------------------------------------------------------- $
$ If there are input data grids defined ( NRINP > 0 ), then these
$ grids are defined first. These grids are defined as if they are wave
$ model grids using the file mod_def.MODID. Each grid is defined on
$ a separate input line with MODID, and eight input flags identifying
$ the presence of 1) water levels 2) currents 3) winds 4) ice and
$ 5-7) assimilation data as in the file ww3_shel.inp.
$
  'input' F F T F F F F
$
$ In this example, we need the file mod_def.input to define the grid
$ and the file wind.input to provide the corresponding wind data.
$
$ ------------------------------------------------------------------- $
$ If all point output is gathered in a unified point output file
$ ( UNIPTS = .TRUE. ), then the output spectral grid needs to be
$ defined. This information is taken from a wave model grid, and only
$ the spectral definitions from this grid are relevant. Define the
$ name of this grid here
$
  'points'
$
$ In this example, we need the file mod_def.points to define the
$ spectral output grid, and the point output will be written to the
$ file out_pnt.points
$
$ ------------------------------------------------------------------- $
$ Now each actual wave model grid is defined using 13 parameters to be
```

```
$ read from a single line in the file. Each line contains the following
$ parameters
$     1)   Define the grid with the extension of the mod_def file.
$    2-8)  Define the inputs used by the grids with 8 keywords
$          corresponding to the 8 flags defining the input in the
$          input files. Valid keywords are:
$            'no'      : This input is not used.
$            'native'  : This grid has its own input files, e.g. grid
$                        grdX (mod_def.grdX) uses ice.grdX.
$            'MODID'   : Take input from the grid identified by
$                        MODID. In the example below, all grids get
$                        their wind from wind.input (mod_def.input).
$     9)   Rank number of grid (internally sorted and reassigned).
$    10)   Group number (internally reassigned so that different
$          ranks result in different group numbers.
$   11-12) Define fraction of communicator (processes) used for this
$          grid. '0.00 1.00' is appropriate in many cases. Partial
$          fractions, i.e. settings other than '0.00 1.00', are
$          intended for equal rank grids, to improve scaling. The
$          commented example provided here (partial fractions with
$          non-equal rank) is not generally recommended.
$    13)   Flag identifying dumping of boundary data used by this
$          grid. If true, the file nest.MODID is generated.
$
  'grd1'  'no' 'no' 'input' 'no' 'no' 'no' 'no'   1  1  0.00 1.00  F
  'grd2'  'no' 'no' 'input' 'no' 'no' 'no' 'no'   2  1  0.00 1.00  F
  'grd3'  'no' 'no' 'input' 'no' 'no' 'no' 'no'   3  1  0.00 1.00  F
$ 'grd1'  'no' 'no' 'input' 'no' 'no' 'no' 'no'   1  1  0.00 0.50  F
$ 'grd2'  'no' 'no' 'input' 'no' 'no' 'no' 'no'   2  1  0.25 0.75  F
$ 'grd3'  'no' 'no' 'input' 'no' 'no' 'no' 'no'   3  1  0.50 1.00  F
$
$ In this example three grids are used requiring the files
$ mod_def.grdN. All files get their winds from the grid 'input'
$ defined by mod_def.input, and no other inputs are used. In the lines
$ that are commented out, each grid runs on a part of the pool of
$ processes assigned to the computation.
$
$ Limitations relevant to irregular (curvilinear) grids:
$   1) Equal rank is not supported when one or more is an irregular
$       grid. Use non-equal rank instead. (see wmgridmd.ftn)
$   2) Non-native input grids: feature is not supported when either
$      an input grid or computational grids is irregular.
$      (see wmupdtmd.ftn)
$   3) Irregular grids with unified point output: This is supported
$      but the feature has not been verified for accuracy.
```

```
$      (see wmiopomd.ftn)
$
$ -------------------------------------------------------------------- $
$ Starting and ending times for the entire model run
$
   19680606 000000   19680607 000000
$
$ -------------------------------------------------------------------- $
$ Specific multi-scale model settings (single line).
$    Flag for masking computation in two-way nesting (except at
$                                                      output times).
$    Flag for masking at printout time.
$
  F F
$
$ -------------------------------------------------------------------- $
$ Conventional output requests as in ww3_shel.inp.  Will be applied
$ to all grids.
$
   19680606 000000   3600  19680607 000000
$--------------------------------------------------------------------
$
$ Output request flags identifying fields as in ww3_shel.inp. See that
$ file for a full documentation of field output options. Namelist type
$ selection is used here (for alternative F/T flags, see ww3_shel.inp).
$
  N
  DPT CUR WND HS T0M1 FP DP PHS PTP PDIR
$
$--------------------------------------------------------------------
$
$ NOTE: If UNIPTS = .TRUE. then the point output needs to be defined
$       here and cannot be redefined  below.
$
   19680606 000000   3600  19680608 000000
       0.E3     0.E3  'eye       '
       0.E3    50.E3  'mN        '
     -35.E3    35.E3  'mNW       '
     -50.E3     0.E3  'mW        '
     -35.E3   -35.E3  'mSW       '
       0.E3   -50.E3  'mS        '
      35.E3   -35.E3  'mSE       '
      50.E3     0.E3  'mE        '
      35.E3    35.E3  'mNE       '
       0.E3   100.E3  'aN        '
```

```
     -70.E3    70.E3  'aNW       '
    -100.E3     0.E3  'aW        '
     -70.E3   -70.E3  'aSW       '
       0.E3  -100.E3  'aS        '
      70.E3   -70.E3  'aSE       '
     100.E3     0.E3  'aE        '
      70.E3    70.E3  'aNE       '
       0.E3   210.E3  'bN        '
    -150.E3   150.E3  'bNW       '
    -210.E3     0.E3  'bW        '
    -150.E3  -150.E3  'bSW       '
       0.E3  -210.E3  'bS        '
     150.E3  -150.E3  'bSE       '
     210.E3     0.E3  'bE        '
     150.E3   150.E3  'bNE       '
       0.E3   800.E3  'cN        '
    -550.E3   550.E3  'cNW       '
    -800.E3     0.E3  'cW        '
    -550.E3  -550.E3  'cSW       '
       0.E3  -800.E3  'cS        '
     550.E3  -550.E3  'cSE       '
     800.E3     0.E3  'cE        '
     550.E3   550.E3  'cNE       '
       0.E3     0.E3  'STOPSTRING'
$
$  Four additional output types: see ww3_shel.inp for documentation.
$
$  track output
   19680606 000000      0  19680608 000000
$
$  restart files
   19680606 000000      0  19680608 000000
$
$  boundary output
   19680606 000000      0  19680608 000000
$
$  separated wave field data
   19680606 000000      0  19680608 000000
$
$ -------------------------------------------------------------------- $
$ Output requests per grid and type to overwrite general setup
$ as defined above. First record per set is the grid name MODID
$ and the output type number. Then follows the standard time string,
$ and conventional data as per output type. In mww3_test_05 this is
$ not used. Below, one example generating partitioning output for
```

```
$ the inner grid is included but commented out.
$
$ 'grd3'  6
$  19680606 000000    900  19680608 000000
$     0 999 1 0 999 1 T
$
$ ------------------------------------------------------------------- $
$ Mandatory end of output requests per grid, identified by output
$ type set to 0.
$
  'the_end'  0
$
$ ------------------------------------------------------------------- $
$ Moving grid data as in ww3_shel.inp. All grids will use same data.
$
  'MOV' 19680606 000000   5.  90.
  'STP'
$
$ ------------------------------------------------------------------- $
$ End of input file                                                   $
$ ------------------------------------------------------------------- $
```

end of example input file (traditional form)

G.10.2 ww3_multi.nml

start of example input file (namelist form)

```
! ------------------------------------------------------------------- !
! WAVEWATCH III - ww3_multi.nml - multi-grid model                    !
! ------------------------------------------------------------------- !


! ------------------------------------------------------------------- !
! Define top-level model parameters via DOMAIN_NML namelist
!
! * IOSTYP defines the output server mode for parallel implementation.
!             0 : No data server processes, direct access output from
!                 each process (requires true parallel file system).
!             1 : No data server process. All output for each type
!                 performed by process that performs computations too.
```

```
!             2 : Last process is reserved for all output, and does no
!                 computing.
!             3 : Multiple dedicated output processes.
!
! * namelist must be terminated with /
! * definitions & defaults:
!     DOMAIN%NRINP  =  0  ! Number of grids defining input fields.
!     DOMAIN%NRGRD  =  1  ! Number of wave model grids.
!     DOMAIN%UNIPTS =  F  ! Flag for using unified point output file.
!     DOMAIN%IOSTYP =  1  ! Output server type
!     DOMAIN%UPPROC =  F  ! Flag for dedicated process for unified point
!     DOMAIN%PSHARE =  F  ! Flag for grids sharing dedicated output proc
!     DOMAIN%FLGHG1 =  F  ! Flag for masking computation in two-way nest
!     DOMAIN%FLGHG2 =  F  ! Flag for masking at printout time
!     DOMAIN%START  = '19680606 000000'  ! Start date for the entire mod
!     DOMAIN%STOP   = '19680607 000000'  ! Stop date for the entire mode
! -------------------------------------------------------------------- !
&DOMAIN_NML
  DOMAIN%NRINP  = 3
  DOMAIN%NRGRD  = 5
  DOMAIN%UNIPTS = T
  DOMAIN%START  = '20100101 120000'
  DOMAIN%STOP   = '20101231 000000'
/



! -------------------------------------------------------------------- !
! Define each input grid via the INPUT_GRID_NML namelist
!
! * index I must match indexes from 1 to DOMAIN%NRINP
! * INPUT(I)%NAME must be set for each active input grid I
!
! * namelist must be terminated with /
! * definitions & defaults:
!     INPUT(I)%NAME                  = 'unset'
!     INPUT(I)%FORCING%WATER_LEVELS  = F
!     INPUT(I)%FORCING%CURRENTS      = F
!     INPUT(I)%FORCING%WINDS         = F
!     INPUT(I)%FORCING%ICE_CONC      = F
!     INPUT(I)%FORCING%ICE_PARAM1    = F
!     INPUT(I)%FORCING%ICE_PARAM2    = F
!     INPUT(I)%FORCING%ICE_PARAM3    = F
!     INPUT(I)%FORCING%ICE_PARAM4    = F
!     INPUT(I)%FORCING%ICE_PARAM5    = F
```

```
!     INPUT(I)%FORCING%MUD_DENSITY   = F
!     INPUT(I)%FORCING%MUD_THICKNESS = F
!     INPUT(I)%FORCING%MUD_VISCOSITY = F
!     INPUT(I)%ASSIM%MEAN            = F
!     INPUT(I)%ASSIM%SPEC1D          = F
!     INPUT(I)%ASSIM%SPEC2D          = F
! -------------------------------------------------------------------- !
&INPUT_GRID_NML

  INPUT(1)%NAME                  = 'atm'
  INPUT(1)%FORCING%WINDS         = T
  INPUT(1)%FORCING%MUD_VISCOSITY = T
  INPUT(1)%ASSIM%MEAN            = T

  INPUT(2)%NAME                  = 'ocn'
  INPUT(2)%FORCING%WATER_LEVELS  = T
  INPUT(2)%FORCING%CURRENTS      = T

  INPUT(3)%NAME                  = 'ice'
  INPUT(3)%FORCING%ICE_CONC      = T
  INPUT(3)%FORCING%ICE_PARAM1    = T
  INPUT(3)%FORCING%ICE_PARAM2    = T

/



! -------------------------------------------------------------------- !
! Define each model grid via the MODEL_GRID_NML namelist
!
! * index I must match indexes from 1 to DOMAIN%NRGRD
! * MODEL(I)%NAME must be set for each active model grid I
! * FORCING can be set as :
!    - 'no'          : This input is not used.
!    - 'native'      : This grid has its own input files, e.g. grid
!                      grdX (mod_def.grdX) uses ice.grdX.
!    - 'INPUT%NAME'  : Take input from the grid identified by
!                      INPUT%NAME.
! * RESOURCE%RANK_ID : Rank number of grid (internally sorted and reassi
! * RESOURCE%GROUP_ID : Group number (internally reassigned so that diff
!                                     ranks result in different group nu
! * RESOURCE%COMM_FRAC : Fraction of communicator (processes) used for t
! * RESOURCE%BOUND_FLAG : Flag identifying dumping of boundary data used
!                         grid. If true, the file nest.MODID is generate
!
```

```
! * Limitations relevant to irregular (curvilinear) grids:
!   1) Equal rank is not supported when one or more is an irregular
!       grid. Use non-equal rank instead. (see wmgridmd.ftn)
!   2) Non-native input grids: feature is not supported when either
!      an input grid or computational grids is irregular.
!      (see wmupdtmd.ftn)
!   3) Irregular grids with unified point output: This is supported
!      but the feature has not been verified for accuracy.
!      (see wmiopomd.ftn)
!
! * namelist must be terminated with /
! * definitions & defaults:
!     MODEL(I)%NAME                  = 'unset'
!     MODEL(I)%FORCING%WATER_LEVELS  = 'no'
!     MODEL(I)%FORCING%CURRENTS      = 'no'
!     MODEL(I)%FORCING%WINDS         = 'no'
!     MODEL(I)%FORCING%ICE_CONC      = 'no'
!     MODEL(I)%FORCING%ICE_PARAM1    = 'no'
!     MODEL(I)%FORCING%ICE_PARAM2    = 'no'
!     MODEL(I)%FORCING%ICE_PARAM3    = 'no'
!     MODEL(I)%FORCING%ICE_PARAM4    = 'no'
!     MODEL(I)%FORCING%ICE_PARAM5    = 'no'
!     MODEL(I)%FORCING%MUD_DENSITY   = 'no'
!     MODEL(I)%FORCING%MUD_THICKNESS = 'no'
!     MODEL(I)%FORCING%MUD_VISCOSITY = 'no'
!     MODEL(I)%ASSIM%MEAN            = 'no'
!     MODEL(I)%ASSIM%SPEC1d          = 'no'
!     MODEL(I)%ASSIM%SPEC2d          = 'no'
!     MODEL(I)%RESOURCE%RANK_ID      = I
!     MODEL(I)%RESOURCE%GROUP_ID     = 1
!     MODEL(I)%RESOURCE%COMM_FRAC    = 0.00,1.00
!     MODEL(I)%RESOURCE%BOUND_FLAG   = F
!
!     MODEL(4)%FORCING = 'no' 'no' 'no' 'no' 'no' 'no'
!
!     MODEL(2)%RESOURCE = 1 1 0.00 1.00 F
! ------------------------------------------------------------------- !
&MODEL_GRID_NML

  MODEL(1)%NAME                  = 'grd1'
  MODEL(1)%FORCING%WINDS         = 'atm'
  MODEL(1)%FORCING%CURRENTS      = 'ocn'
  MODEL(1)%FORCING%WATER_LEVELS  = 'ocn'

  MODEL(2)%NAME                  = 'grd2'
```

```
  MODEL(2)%FORCING%WINDS         = 'atm'
  MODEL(2)%FORCING%CURRENTS      = 'ocn'
  MODEL(2)%FORCING%WATER_LEVELS  = 'ocn'
  MODEL(2)%FORCING%ICE_CONC      = 'ice'

  MODEL(3)%NAME                  = 'grd3'
  MODEL(3)%FORCING%WINDS         = 'atm'
  MODEL(3)%FORCING%CURRENTS      = 'ocn'
  MODEL(3)%FORCING%WATER_LEVELS  = 'ocn'
  MODEL(3)%FORCING%ICE_CONC      = 'ice'

  MODEL(4)%NAME = 'grd4'
  MODEL(5)%NAME = 'grd5'

  MODEL(4)%FORCING = 'ocn' 'ocn' 'atm' 'ice' 'ice' 'ice'
  MODEL(5)%FORCING = 'ocn' 'ocn' 'atm' 'ice' 'ice' 'ice'


  MODEL(1)%RESOURCE = 1 1 0.00 0.50 T
  MODEL(2)%RESOURCE = 2 1 0.25 0.75 F
  MODEL(3)%RESOURCE = 3 1 0.50 1.00 F
  MODEL(4)%RESOURCE = 4 1 0.00 1.00 F
  MODEL(5)%RESOURCE = 4 1 0.00 1.00 F

  MODEL(5)%RESOURCE%BOUND_FLAG = T

/


! -------------------------------------------------------------------- !
! Define the output types point parameters via OUTPUT_TYPE_NML namelist
!
! * index I must match indexes from 1 to DOMAIN%NRGRD
!
! * ALLTYPE will apply the output types for all the model grids
!
! * ITYPE(I) will apply the output types for the model grid number I
!
! * need DOMAIN%UNIPTS equal true to use a unified point output file
!
! * the point file is a space separated values per line : lon lat 'name'
!
! * the detailed list of field names is given in model/nml/ww3_shel.nml
!  DPT CUR WND AST WLV ICE IBG D50 IC1 IC5
!  HS LM T02 T0M1 T01 FP DIR SPR DP HIG
```

```
!  EF TH1M STH1M TH2M STH2M WN
!  PHS PTP PLP PDIR PSPR PWS PDP PQP PPE PGW PSW PTM10 PT01 PT02 PEP TWS
!  UST CHA CGE FAW TAW TWA WCC WCF WCH WCM FWS
!  SXY TWO BHD FOC TUS USS P2S USF P2L TWI FIC
!  ABR UBR BED FBB TBB
!  MSS MSC WL02 AXT AYT AXY
!  DTD FC CFX CFD CFK
!  U1 U2
!
! * output track file formatted (T) or unformated (F)
!
! * namelist must be terminated with /
! * definitions & defaults:
!     ALLTYPE%FIELD%LIST         =  'unset'
!     ALLTYPE%POINT%NAME         =  'unset'
!     ALLTYPE%POINT%FILE         =  'points.list'
!     ALLTYPE%TRACK%FORMAT       =  T
!     ALLTYPE%PARTITION%X0       =  0
!     ALLTYPE%PARTITION%XN       =  0
!     ALLTYPE%PARTITION%NX       =  0
!     ALLTYPE%PARTITION%Y0       =  0
!     ALLTYPE%PARTITION%YN       =  0
!     ALLTYPE%PARTITION%NY       =  0
!     ALLTYPE%PARTITION%FORMAT   =  T
!
!     ITYPE(3)%TRACK%FORMAT      =  F
! -------------------------------------------------------------------- !
&OUTPUT_TYPE_NML
  ALLTYPE%POINT%NAME     = 'points'
  ALLTYPE%FIELD%LIST     = 'HS DIR SPR'

  ITYPE(1)%FIELD%LIST    = 'HS DIR SPR WND ICE CUR LEV'
/



! -------------------------------------------------------------------- !
! Define output dates via OUTPUT_DATE_NML namelist
!
! * index I must match indexes from 1 to DOMAIN%NRGRD
! * ALLDATE will apply the output dates for all the model grids
! * IDATE(I) will apply the output dates for the model grid number i
! * start and stop times are with format 'yyyymmdd hhmmss'
! * if time stride is equal '0', then output is disabled
! * time stride is given in seconds
```

```
! * it is possible to overwrite a global output date for a given grid
!
! * namelist must be terminated with /
! * definitions & defaults:
!     ALLDATE%FIELD%START         =  '19680606 000000'
!     ALLDATE%FIELD%STRIDE        =  '0'
!     ALLDATE%FIELD%STOP          =  '19680607 000000'
!     ALLDATE%POINT%START         =  '19680606 000000'
!     ALLDATE%POINT%STRIDE        =  '0'
!     ALLDATE%POINT%STOP          =  '19680607 000000'
!     ALLDATE%TRACK%START         =  '19680606 000000'
!     ALLDATE%TRACK%STRIDE        =  '0'
!     ALLDATE%TRACK%STOP          =  '19680607 000000'
!     ALLDATE%RESTART%START       =  '19680606 000000'
!     ALLDATE%RESTART%STRIDE      =  '0'
!     ALLDATE%RESTART%STOP        =  '19680607 000000'
!     ALLDATE%BOUNDARY%START      =  '19680606 000000'
!     ALLDATE%BOUNDARY%STRIDE     =  '0'
!     ALLDATE%BOUNDARY%STOP       =  '19680607 000000'
!     ALLDATE%PARTITION%START     =  '19680606 000000'
!     ALLDATE%PARTITION%STRIDE    =  '0'
!     ALLDATE%PARTITION%STOP      =  '19680607 000000'
!
!     ALLDATE%RESTART             =  '19680606 000000' '0' '19680607 000
!
!     IDATE(3)%PARTITION%START    =  '19680606 000000'
! -------------------------------------------------------------------- !
&OUTPUT_DATE_NML
  ALLDATE%FIELD%START         = '20100101 000000'
  ALLDATE%FIELD%STRIDE        = '3600'
  ALLDATE%FIELD%STOP          = '20101231 000000'
  ALLDATE%POINT%START         = '20100101 000000'
  ALLDATE%POINT%STRIDE        = '3600'
  ALLDATE%POINT%STOP          = '20101231 000000'

  ALLDATE%RESTART             = '20101231 000000' '43200' '20501231 0000

  IDATE(5)%PARTITION%START    = '20100601 000000'
  IDATE(5)%PARTITION%STRIDE   = '3600'
  IDATE(5)%PARTITION%START    = '20101201 000000'
/



! -------------------------------------------------------------------- !
```

```
! Define homogeneous input via HOMOG_COUNT_NML and HOMOG_INPUT_NML namel
!
! * the number of each homogeneous input is defined by HOMOG_COUNT
! * the total number of homogeneous input is automatically calculated
! * the homogeneous input must start from index 1 to N
! * if VALUE1 is equal 0, then the homogeneous input is desactivated
! * NAME can only be MOV
! * each homogeneous input is defined over a maximum of 3 values detaill
!     - MOV is defined by speed and direction
!
! * namelist must be terminated with /
! * definitions & defaults:
!     HOMOG_COUNT%N_MOV             =  0
!
!     HOMOG_INPUT(I)%NAME           =  'unset'
!     HOMOG_INPUT(I)%DATE           =  '19680606 000000'
!     HOMOG_INPUT(I)%VALUE1         =  0
!     HOMOG_INPUT(I)%VALUE2         =  0
!     HOMOG_INPUT(I)%VALUE3         =  0
! ------------------------------------------------------------------- !
&HOMOG_COUNT_NML
  HOMOG_COUNT%N_MOV         = 1
/

&HOMOG_INPUT_NML
  HOMOG_INPUT(1)%NAME       = 'MOV'
  HOMOG_INPUT(1)%DATE       ='20100610 000000'
  HOMOG_INPUT(1)%VALUE1     = 5.
  HOMOG_INPUT(1)%VALUE2     = 90.
/


! ------------------------------------------------------------------- !
! WAVEWATCH III - end of namelist                                     !
! ------------------------------------------------------------------- !
```

end of example input file (namelist form)

G.11 ww3_gint

G.11.1 ww3_gint.inp

start of example input file (traditional form)

```
$ -------------------------------------------------------------------- $
$ WAVEWATCH III Grid integration input file                            $
$ -------------------------------------------------------------------- $
$ Time, time increment and number of outputs
$
  19680606 060000 10800. 1
$
$ Total number of grids (NGR). The code assumes that the first NGR-1
$ grids are the input grids and the last grid is the target grid in
$ which the output fields are to be interpolated. It also assumes
$ that all the grids have the same output fields switched on
$
$ NGR
$
 4
$
$ Grid Ids
$
 'grd1'
 'grd2'
 'grd3'
 'grd4'
$
$ In this example grd1, grd2 and grd3 are the input grids. For each
$ of these grids a mod_def.grdN and an out_grd.grdN are available.
$ The target grid is grd4, and a mod_def.grd4 is also made available.
$ Upon execution of the code an out_grd.grd4 is generated via
$ interpolation of output fields from the various out_grd.grdN
$ (N varying from 1 to 3) files.
$
$ Choice of no extrapolation [0] or to extrapolate [1] with wet points [
 0
$
$ -------------------------------------------------------------------- $
$ End of input file                                                    $
$ -------------------------------------------------------------------- $
```

end of example input file (traditional form)

G.12 ww3_outf

G.12.1 ww3_outf.inp

start of example input file (traditional form)

```
$ ------------------------------------------------------------------- $
$ WAVEWATCH III Grid output post-processing                            $
$-------------------------------------------------------------------- $
$ Time, time increment and number of outputs
$
  19680606 060000 10800. 1
$
$ Output request flags identifying fields as in ww3_shel.inp. See this
$ file for a full documentation of the field output options.
$
  N
  DPT HS FP T01 WL02 ALPXT ALPYT ALPXY
$
$ Output type ITYPE [0,1,2,3], and IPART [ 0,...,NOSWLL ]
$
  1  0
$ ------------------------------------------------------------------- $
$ 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 longitude, latitude
$            and parameter value(s) per record (defined points only),
$
$ 2 11 2 11 1 2
$
$ For each field and time a new file is generated with the file name
$ ww3.yymmddhh.xxx, where yymmddhh is a conventional time indicator,
$ 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 preprocessor. 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 (traditional form)

G.13 ww3_ounf

G.13.1 ww3_ounf.inp

start of example input file (traditional form)

```
$ -------------------------------------------------------------------- $
$ WAVEWATCH III Grid output post-processing                             $
$--------------------------------------------------------------------- $
$ First output time (yyyymmdd hhmmss), increment of output (s),
```

```
$ and number of output times.
$
  19850101 000000  3600.  1000
$
$ Fields requested ---------------------------------------------------- $
$
$ Output request flags identifying fields as in ww3_shel.inp. See that
$ file for a full documentation of field output options. Namelist type
$ selection is used here (for alternative F/T flags, see ww3_shel.inp).
$
$ DPT CUR WND AST WLV ICE IBG D50 IC1 IC5 HS LM T02 T0M1 T01 FP DIR SPR
$ DP HIG EF TH1M STH1M TH2M STH2M WN PHS PTP PLP PDIR PSPR PWS PDP
$ PQP PPE PGW PSW PTM10 PT01 PT02 PEP TWS PNR UST CHA CGE FAW TAW TWA WC
$ WCF WCH WCM SXY TWO BHD FOC TUS USS P2S USF P2L TWI FIC ABR UBR BED
$ FBB TBB MSS MSC DTD FC CFX CFD CFK U1 U2
$
 N
 DPT HS FP T01
$
$-------------------------------------------------------------------- $
$ netCDF version [3,4]
$        and variable type 4 [2 = SHORT, 3 = it depends , 4 = REAL]
$ swell partitions [0 1 2 3 4 5]
$ variables in same file [T] or not [F]
$
 3 4
 0 1 2
 T
$
$ -------------------------------------------------------------------- $
$ File prefix
$ number of characters in date [4(yearly),6(monthly),8(daily),10(hourly)
$ IX and IY ranges [regular:IX NX IY NY, unstructured:IP NP 1 1]
$
 ww3.
 6
$
$ ----------------- FOR SMC GRID ONLY ---------------------------------$
$ SMC output type:
$    1 = Flat points file (1D sea point array)
$    2 = Regular gridded (2D lat/lon array over region)
$
$  1
$
$ For SMC output, the IX/IY range line is replaced with a domain
```

```
$ lat/lon range and ‘cellfac‘ parameter for SMC type 2 output.
$ First/Last lat/lon can be set to -999.9 to use edge of SMC grid.
$
$ For SMC type 1 output, only those points within the specified
$ lat/lon ranges will be extracted:
$
$   first_lon, first_lat, last_lon, last_lat
$
$ For type 2 output, the range is used in conjunction with a fifth
$ integer ‘cellfac‘ parameter to specify the regular output grid
$ to area average the SMC grid to. In this case, the output grid will
$ be aligned to nearest largest SMC grid points within the selected
$ region. Therefore, the output grid start and end lat/lons may differ
$ slightly from what is requested. In order to obtain a fully populated
$ regular grid the extents specified should encompass the SW corner of
$ the bottom left cell, and NE corner of the top right cell required.
$ ‘cellfac‘ is an integer value that selects the target grid cell size
$ as multiple of the smallest SMC grid cell. It must be a power of 2.
$ e.g. 1 = highest resolution, 2 = half resolution of smallest cell,
$ 4 = 1/4 res of smallest cell, etc.
$
$   first_lon, first_lat, last_lon, last_lat, cellfac
$
$ Example: Extract high resolution data for U.K.:
$
$ -13.50  46.85  5.50  61.0  1
$
$
$ ---------------- FOR NON-SMC GRIDS -------------------------------- $
$ IX, IY range:
$ 1 1000000 1 1000000
$
$ For each field and time a new file is generated with the file name
$ ww3.date_xxx.nc , where date is a conventional time indicator with S3
$ characters, and xxx is a field identifier.
$
$ ------------------------------------------------------------------- $
$ End of input file                                                   $
$ ------------------------------------------------------------------- $
```

end of example input file (traditional form)

G.13.2 ww3_ounf.nml

start of example input file (namelist form)

```
! ------------------------------------------------------------------- !
! WAVEWATCH III - ww3_ounf.nml - Grid output post-processing          !
! ------------------------------------------------------------------- !


! ------------------------------------------------------------------- !
! Define the output fields to postprocess via FIELD_NML namelist
!
! * the detailed list of field names FIELD%LIST is given in ww3_shel.nml
!  DPT CUR WND AST WLV ICE IBG D50 IC1 IC5
!  HS LM T02 T0M1 T01 FP DIR SPR DP HIG
!  EF TH1M STH1M TH2M STH2M WN
!  PHS PTP PLP PDIR PSPR PWS PDP PQP PPE PGW PSW PTM10 PT01 PT02 PEP TWS
!  UST CHA CGE FAW TAW TWA WCC WCF WCH WCM FWS
!  SXY TWO BHD FOC TUS USS P2S USF P2L TWI FIC
!  ABR UBR BED FBB TBB
!  MSS MSC WL02 AXT AYT AXY
!  DTD FC CFX CFD CFK
!  U1 U2
!
! * namelist must be terminated with /
! * definitions & defaults:
!     FIELD%TIMESTART           = '19000101 000000'  ! Stop date for th
!     FIELD%TIMESTRIDE          = '0'                ! Time stride for
!     FIELD%TIMECOUNT           = '1000000000'       ! Number of time s
!     FIELD%TIMESPLIT           = 6                  ! [4(yearly),6(mon
!     FIELD%LIST                = 'unset'            ! List of output f
!     FIELD%PARTITION           = '0 1 2 3'          ! List of wave par
!     FIELD%SAMEFILE            = T                  ! All the variable
!     FIELD%TYPE                = 3                  ! [2 = SHORT, 3 =
! ------------------------------------------------------------------- !
&FIELD_NML
  FIELD%TIMESTART        =  '20100101 000000'
  FIELD%TIMESTRIDE       =  '3600'
  FIELD%LIST             =  'DPT WND HS FP DIR SPR MSS'
  FIELD%SAMEFILE         =  F
  FIELD%TYPE             =  4
/
```

```
! ------------------------------------------------------------------- !
! Define the content of the output file via FILE_NML namelist
!
! * namelist must be terminated with /
! * definitions & defaults:
!     FILE%PREFIX        = 'ww3.'            ! Prefix for output file na
!     FILE%NETCDF        = 3                 ! Netcdf version [3|4]
!     FILE%IX0           = 1                 ! First X-axis or node inde
!     FILE%IXN           = 1000000000        ! Last X-axis or node index
!     FILE%IY0           = 1                 ! First Y-axis index
!     FILE%IYN           = 1000000000        ! Last Y-axis index
! ------------------------------------------------------------------- !
&FILE_NML
  FILE%NETCDF        = 4
/

! ------------------------------------------------------------------- !
! Define the content of the output file via SMC_NML namelist
!
! * For SMC grids, IX0, IXN, IY0 and IYN from FILE_NML are not used.
!   Two types of output are available:
! *   TYPE=1: Flat 1D "seapoint" array of grid cells.
! *   TYPE=2: Re-gridded regular grid with cell sizes being an integer
! *           multiple of the smallest SMC grid cells size.
!
! * Note that the first/last longitudes and latitudes will be adjusted
!  to snap to the underlying SMC grid edges. CELFAC is only used for
!  type 2 output and defines the output cell sizes as an integer
!  multiple of the smallest SMC Grid cell size. CELFAC should be a
!  power of 2, e.g: 1,2,4,8,16, etc...
!
! * namelist must be terminated with /
! * definitions & defaults:
!     SMC%TYPE          = 1              ! SMC Grid type (1 or 2)
!     SMC%SX0           = -999.9         ! First longitude
!     SMC%EX0           = -999.9         ! Last longitude
!     SMC%SY0           = -999.9         ! First latitude
!     SMC%EY0           = -999.9         ! Last latitude
!     SMC%CELFAC        = 1              ! Cell size factor (SMCTYPE=2 o
!     SMC%NOVAL         = UNDEF          ! Fill value for wet cells with
! ------------------------------------------------------------------- !
&SMC_NML
/
```

```
! ------------------------------------------------------------------- !
! WAVEWATCH III - end of namelist                                     !
! ------------------------------------------------------------------- !
```

end of example input file (namelist form)

G.14 gx_outf

G.14.1 gx_outf.inp

start of example input file (traditional form)

```
$ ------------------------------------------------------------------- $
$ WAVEWATCH III Grid output post-processing ( GrADS )                 $
$-------------------------------------------------------------------- $
$ Time, time increment and number of outputs.
$
  19680606 000000  3600. 25
$
$ Output request flags identifying fields as in ww3_shel.inp. See that
$ file for a full documentation of field output options. Namelist type
$ selection is used here (for alternative F/T flags, see ww3_shel.inp).
$
  N
  DPT HS FP T01
$
$-------------------------------------------------------------------
$ Grid range in discrete counters IXmin,max, IYmin,max, flags for
$ including sea and boundary points in map
$
  0 999 0 999  T T
$
$ NOTE : In the Cartesian grid version of the code, X and Y are
$        converted to longitude and latitude assuming that 1 degree
$        equals 100 km if the maximum of X or Y is larger than 1000km.
$        For maxima between 100 and 1000km 1 degree is assumed to be
$        10km etc. Adjust labels in GrADS scripts accordingly.
```

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

end of example input file (traditional form)

G.15 ww3_grib

G.15.1 ww3_grib.inp

start of example input file (traditional form)

```
$ ------------------------------------------------------------------- $
$ WAVEWATCH III Grid output post-processing ( GRIB )                  $
$-------------------------------------------------------------------- $
$ Time, time increment and number of outputs.
$
  19680606 000000  3600. 3
$
$ Output request flags identifying fields as in ww3_shel.inp. See that
$ file for a full documentation of field output options. Namelist type
$ selection is used here (for alternative F/T flags, see ww3_shel.inp).
$
  N
  DPT HS FP T01
$
$ Additional info needed for grib file
$   Forecast time, center ID, generating process ID, grid definition,
$   GDS/BMS flag and grid definition template number GDTN (0 = regular;
$   30 = Lambert Conformal, only these two types available now)
$
  19680606 010000 7 10 255 192 0
$
$ if GDTN is 30 (lambert conformal) read next line with proj parms
$ LATAN1, LONV, DSX, DSY, SCNMOD, LATIN1, LATIN2, LATSP, LONSP
$ Example for the GLW grid at NCEP
$ 25 265 2.539703 2.539703 64 25 25 -90 0
```

```
$ Other curvilinear grids not yet implemented
$
$ -------------------------------------------------------------------- $
$ End of input file                                                    $
$ -------------------------------------------------------------------- $
```

end of example input file (traditional form)

G.16 ww3_outp

G.16.1 ww3_outp.inp

start of example input file (traditional form)

```
$ -------------------------------------------------------------------- $
$ WAVEWATCH III Point output post-processing                           $
$--------------------------------------------------------------------- $
$ First output time (yyyymmdd hhmmss), increment of output (s),
$ and number of output times.
$
  19680606 060000  3600.  7
$
$ Points requested ---------------------------------------------------- $
$ Define points for which output is to be generated.
$
$ 1
$ 2
  3
$ 4
$
$ mandatory end of list
 -1
$
$ Output type ITYPE [0,1,2,3,4]
$
  1
$ -------------------------------------------------------------------- $
$ ITYPE = 0, inventory of file.
```

```
$           No additional input, the above time range is ignored.
$
$ ------------------------------------------------------------------- $
$ ITYPE = 1, Spectra.
$         - Sub-type OTYPE :  1 : Print plots.
$                             2 : Table of 1-D spectra
$                             3 : Transfer file.
$         - Scaling factors for 1-D and 2-D spectra Negative factor
$           disables, output, factor = 0. gives normalized spectrum.
$         - Unit number for transfer file, also used in table file
$           name.
$         - Flag for unformatted transfer file.
$
  1   0.  0.  33  F
$
$ The transfer file contains records with the following contents.
$
$ - File ID in quotes, number of frequencies, directions and points.
$   grid name in quotes (for unformatted file C*21,3I,C*30).
$ - Bin frequencies 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 using free format throughout.
$ This data 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'
$                             6 : WMO standard output
$         - Unit number for file, also used in file name.
$
$   6  66
$
```

```
$ 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, ice and total source term.
$         - scale ISCALE for OTYPE=2,3
$                             0 : Dimensional.
$                             1 : Nondimensional in terms of U10
$                             2 : Nondimensional in terms of U*
$                            3-5: like 0-2 with f normalized with fp.
$         - Flag for unformatted transfer file.
$
$ 1  0. 0. 50   T T T T T T T 0  F
$
$ The transfer file contains records with the following contents.
$
$ - File ID in quotes, number of frequencies, directions and points,
$   flags for spectrum and source terms (C*21, 3I, 6L)
$ - Bin frequencies 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                     |          |
$ - Sice(f,theta) if requested                    |          |
$ - Stot(f,theta) if requested                    |          |
$                                                 -+         -+
```

```
$ ------------------------------------------------------------------- $
$ ITYPE = 4, Spectral partitions and bulletins
$          - Sub-type OTYPE :  1 : Spectral partitions
$                              2 : Bulletins ASCII format
$                              3 : Bulletins CSV format
$                              4 : Bulletins ASCII and CSV formats
$          - Unit number for transfer file, also used in table file
$            name.
$          - Reference date/time in YYYYMMDD HHMMSS format, used for
$            including in bulletin legend, and computing forecast time
$            in CSV type output (if the first field is negative, the
$            reference time becomes the first simulation time slice)
$          - Three-character code indicating time zone (UTC, EST etc)
$
$   4  2 19680606 060000 'UTC'
$
$ The transfer file contains records with the following contents.
$
$ - File ID in quotes, number of frequencies, directions and points.
$   grid name in quotes (for unformatted file C*21,3I,C*30).
$ - Bin frequencies 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
$                                                 -+        -+
$
$ ------------------------------------------------------------------- $
$ End of input file                                                   $
$ ------------------------------------------------------------------- $
```

end of example input file (traditional form)

G.17 ww3_ounp

G.17.1 ww3_ounp.inp

start of example input file (traditional form)

```
$ -------------------------------------------------------------------- $
$ WAVEWATCH III NETCDF Point output post-processing                    $
$-------------------------------------------------------------------- $
$ First output time (yyyymmdd hhmmss), increment of output (s),
$ and number of output times.
$
  19850101 000000  3600.  1000
$
$ Points requested ---------------------------------------------------- $
$
$ Define points index for which output is to be generated.
$ If no one defined, all points are selected
$ One index number per line, negative number identifies end of list.
$ 1
$ 2
$ mandatory end of list
 -1
$
$-------------------------------------------------------------------- $
$ file prefix
$ number of characters in date [4(yearly),6(monthly),8(daily),10(hourly)
$ netCDF version [3,4]
$ points in same file [T] or not [F]
$                    and max number of points to be processed in one pas
$ output type ITYPE [0,1,2,3]
$ flag for global attributes WW3 [0] or variable version [1-2-3-4]
$ flag for dimensions order time,station [T] or station,time [F]
$
 ww3.
 6
 4
 T 150
 1
 0
 T
$
$ -------------------------------------------------------------------- $
```

```
$ ITYPE = 0, inventory of file.
$            No additional input, the above time range is ignored.
$
$ ------------------------------------------------------------------- $
$ ITYPE = 1, netCDF Spectra.
$          - Sub-type OTYPE :  1 : Print plots.
$                              2 : Table of 1-D spectra
$                              3 : Transfer file.
$                              4 : Spectral partitioning.
$          - Scaling factors for 1-D and 2-D spectra Negative factor
$            disables, output, factor = 0. gives normalized spectrum.
$
  3  1  0
$
$ The transfer file contains records with the following contents.
$
$ - File ID in quotes, number of frequencies, directions and points.
$   grid name in quotes (for unformatted file C*21,3I,C*30).
$ - Bin frequencies 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
$                                              -+          -+
$
$ ------------------------------------------------------------------- $
$ ITYPE = 2, netCDF 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'
$                              6 : WMO standard output
$  4
$
$ ------------------------------------------------------------------- $
$ ITYPE = 3, netCDF 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.
$          - Flags for spectrum, input, interactions, dissipation,
$            bottom, ice 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.
$
$  4  0  0  T T T T T T T  0
$
$ The transfer file contains records with the following contents.
$
$ - File ID in quotes, number of frequencies, directions and points,
$   flags for spectrum and source terms (C*21, 3I, 6L)
$ - Bin frequencies 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                     |         |
$ - Sice(f,theta) if requested                    |         |
$ - Stot(f,theta) if requested                    |         |
$                                                 -+        -+
$
$ ------------------------------------------------------------------- $
$ End of input file                                                   $
$ ------------------------------------------------------------------- $
```

end of example input file (traditional form)

G.17.2 ww3_ounp.nml

start of example input file (namelist form)

```
! ------------------------------------------------------------------- !
! WAVEWATCH III - ww3_ounp.nml - Point output post-processing         !
! ------------------------------------------------------------------- !


! ------------------------------------------------------------------- !
! Define the output fields to postprocess via POINT_NML namelist
!
!
! * namelist must be terminated with /
! * definitions & defaults:
!     POINT%TIMESTART            = '19000101 000000'  ! Stop date for th
!     POINT%TIMESTRIDE           = '0'                ! Time stride for
!     POINT%TIMECOUNT            = '1000000000'       ! Number of time s
!     POINT%TIMESPLIT            = 6                  ! [4(yearly),6(mon
!     POINT%LIST                 = 'all'              ! List of points i
!     POINT%SAMEFILE             = T                  ! All the points i
!     POINT%BUFFER               = 150                ! Number of points
!     POINT%TYPE                 = 1                  ! [0=inventory | 1
!     POINT%DIMORDER             = T                  ! [time,station=T
! ------------------------------------------------------------------- !
&POINT_NML
  POINT%TIMESTART        =  '20100101 000000'
  POINT%TIMESTRIDE       =  '3600'
/


! ------------------------------------------------------------------- !
! Define the content of the output file via FILE_NML namelist
!
! * namelist must be terminated with /
! * definitions & defaults:
!     FILE%PREFIX        = 'ww3.'            ! Prefix for output file na
!     FILE%NETCDF        = 3                 ! Netcdf version [3|4]
! ------------------------------------------------------------------- !
&FILE_NML
  FILE%NETCDF        = 4
/
```

```
! ------------------------------------------------------------------- !
! Define the type 0, inventory of file
!
! * namelist must be terminated with /
! * definitions & defaults:
!           No additional input, the above time range is ignored.
! ------------------------------------------------------------------- !



! ------------------------------------------------------------------- !
! Define the type 1, spectra via SPECTRA_NML namelist
!
! Table of 1-D spectra content :
!   - time, station id, station name, longitude, latitude
!   - frequency : unit Hz, center band frequency - linear log scale (XFR
!   - ffp, f, th1m, sth1m, alpha : 1D spectral parameters
!   - dpt, ust, wnd, wnddir : mean parameters
!
! Transfert file content :
!   - time, station id, station name, longitude, latitude
!   - frequency : unit Hz, center band frequency - linear log scale (XFR
!   - frequency1 : unit Hz, lower band frequency
!   - frequency2 : unit Hz, upper band frequency
!   - direction : unit degree, convention to, origin East, trigonometric
!   - efth(time,station,frequency,direction) : 2D spectral density
!   - dpt, wnd, wnddir, cur, curdir : mean parameters
!
! Spectral partitioning content :
!   - time, station id, station name, longitude, latitude
!   - npart : number of partitions
!   - hs, tp, lm, th1m, sth1m, ws, tm10, t01, t02 : partitioned paramete
!   - dpt, wnd, wnddir, cur, curdir : mean parameters
!
!
! * namelist must be terminated with /
! * definitions & defaults:
!     SPECTRA%OUTPUT        = 3            ! 1: Print plots
!                                          ! 2: Table of 1-D spectra
!                                          ! 3: Transfer file
!                                          ! 4: Spectral partitioning
!     SPECTRA%SCALE_FAC     = 1            ! Scale factor (-1=disabled)
!     SPECTRA%OUTPUT_FAC    = 0            ! Output factor (0=normalized
! ------------------------------------------------------------------- !
&SPECTRA_NML
```

```
/


! ------------------------------------------------------------------- !
! Define the type 2, mean parameter via PARAM_NML namelist
!
! Forcing parameters content :
! - dpt, wnd, wnddir, cur, curdir
!
! Mean wave parameters content :
! - hs, lm, tr, th1p, sth1p, fp, th1m, sth1m
!
! Nondimensional parameters (U*) content :
! - ust, efst, fpst, cd, alpha
!
! Nondimensional parameters (U10) content :
! - wnd, efst, fpst, cd, alpha
!
! Validation table content :
! - wnd, wnddir, hs, hsst, cpu, cmu, ast
!
! WMO stantdard output content :
! - wnd, wnddir, hs, tp
!
! * namelist must be terminated with /
! * definitions & defaults:
!     PARAM%OUTPUT      = 4               ! 1: Forcing parameters
!                                         ! 2: Mean wave parameters
!                                         ! 3: Nondimensional pars. (U*
!                                         ! 4: Nondimensional pars. (U1
!                                         ! 5: Validation table
!                                         ! 6: WMO standard output
! ------------------------------------------------------------------- !
&PARAM_NML
/


! ------------------------------------------------------------------- !
! Define the type 3, source terms via SOURCE_NML namelist
!
! Table of 1-D S(f) content :
!   - time, station id, station name, longitude, latitude
!   - frequency : unit Hz, center band frequency
!   - ef(frequency)   : 1D spectral density
!   - Sin(frequency)  : input source term
```

```
!   - Snl(frequency)  : non linear interactions source term
!   - Sds(frequency)  : dissipation source term
!   - Sbt(frequency)  : bottom source term
!   - Sice(frequency) : ice source term
!   - Stot(frequency) : total source term
!   - dpt, ust, wnd : mean parameters
!
! Table of 1-D inverse time scales (1/T = S/F) content :
!   - time, station id, station name, longitude, latitude
!   - frequency : unit Hz, center band frequency
!   - ef(frequency)   : 1D spectral density
!   - tini(frequency)  : input inverse time scales source term
!   - tnli(frequency)  : non linear interactions inverse time scales sou
!   - tdsi(frequency)  : dissipation inverse time scales source term
!   - tbti(frequency)  : bottom inverse time scales source term
!   - ticei(frequency) : ice inverse time scales source term
!   - ttoti(frequency) : total inverse time scales source term
!   - dpt, ust, wnd : mean parameters
!
! Transfert file content :
!   - time, station id, station name, longitude, latitude
!   - frequency : unit Hz, center band frequency - linear log scale (XFR
!   - frequency1 : unit Hz, lower band frequency
!   - frequency2 : unit Hz, upper band frequency
!   - direction : unit degree, convention to, origin East, trigonometric
!   - efth(frequency,direction) : 2D spectral density
!   - Sin(frequency,direction)  : input source term
!   - Snl(frequency,direction)  : non linear interactions source term
!   - Sds(frequency,direction)  : dissipation source term
!   - Sbt(frequency,direction)  : bottom source term
!   - Sice(frequency,direction) : ice source term
!   - Stot(frequency,direction) : total source term
!   - dpt, wnd, wnddir, cur, curdir, ust : mean parameters
!
!
! * namelist must be terminated with /
! * definitions & defaults:
!     SOURCE%OUTPUT         = 4            ! 1: Print plots
!                                          ! 2: Table of 1-D S(f)
!                                          ! 3: Table of 1-D inverse tim
!                                          ! 4: Transfer file
!     SOURCE%SCALE_FAC      = 0            ! Scale factor (-1=disabled)
!     SOURCE%OUTPUT_FAC     = 0            ! Output factor (0=normalized
!     SOURCE%TABLE_FAC      = 0            ! Table factor
!                                                        0 : Dimension
```

```
!                                                                 1 : Nondimens
!                                                                 2 : Nondimens
!                                                                 3-5: like 0-2
!     SOURCE%SPECTRUM      = T             ! [T|F]
!     SOURCE%INPUT         = T             ! [T|F]
!     SOURCE%INTERACTIONS  = T             ! [T|F]
!     SOURCE%DISSIPATION   = T             ! [T|F]
!     SOURCE%BOTTOM        = T             ! [T|F]
!     SOURCE%ICE           = T             ! [T|F]
!     SOURCE%TOTAL         = T             ! [T|F]
! ------------------------------------------------------------------- !
&SOURCE_NML
/


! ------------------------------------------------------------------- !
! WAVEWATCH III - end of namelist                                     !
! ------------------------------------------------------------------- !
```

end of example input file (namelist form)

G.18 gx_outp

G.18.1 gx_outp.inp

start of example input file (traditional form)

```
$ ------------------------------------------------------------------- $
$ WAVEWATCH III Point output post-processing ( GrADS )                $
$-------------------------------------------------------------------- $
$ First output time (yyyymmdd hhmmss), increment of output (s),
$ and number of output times.
$
  19680606 000000  3600. 7
$
$ Points requested ---------------------------------------------------- $
$ Define points for which output is to be generated.
$
```

```
$ 1
$ 2
  3
$ 4
$ mandatory end of list
 -1
$
$ -------------------------------------------------------------------- $
$   Flags for plotting F, Sin, Snl, Sds, Sbt, Sice, Stot
$
  T T T T T T T
$
$ NOTE : In the Cartesian grid version of the code, X and Y are
$        converted to km. Use source_xy.gs instead of source.gs
$
$ -------------------------------------------------------------------- $
$ End of input file                                                    $
$ -------------------------------------------------------------------- $
```

end of example input file (traditional form)

G.19 ww3_trck

G.19.1 ww3_trck.inp

start of example input file (traditional form)

```
$ -------------------------------------------------------------------- $
$ WAVEWATCH III Track output post-processing                           $
$-------------------------------------------------------------------- $
$ The number of wavenumbers and directions need to be read in as they
$ determine the record length of the data file ...
$
  25  24
$
$ -------------------------------------------------------------------- $
$ End of input file                                                    $
$ -------------------------------------------------------------------- $
```

end of example input file (traditional form)

G.20 ww3_trnc

G.20.1 ww3_trnc.inp

start of example input file (traditional form)

```
$ ------------------------------------------------------------------- $
$ WAVEWATCH III Track output post-processing                           $
$-------------------------------------------------------------------- $
$ First output time (yyyymmdd hhmmss), increment of output (s),
$ and number of output times.
$
  19680606 000000  3600.  100000
$
$ Output type ------------------------------------------------------- $
$ netCDF version [3,4]
$ file prefix
$ number of characters in date [4(yearly),6(monthly),8(daily),10(hourly)
$
 3
 ww3.
 6
$
$ ------------------------------------------------------------------- $
$ End of input file                                                   $
$ ------------------------------------------------------------------- $
```

end of example input file (traditional form)

G.20.2 ww3_trnc.nml

start of example input file (namelist form)

```
! ------------------------------------------------------------------- !
! WAVEWATCH III - ww3_trnc.nml - Track output post-processing         !
! ------------------------------------------------------------------- !


! ------------------------------------------------------------------- !
! Define the output fields to postprocess via TRACK_NML namelist
!
! * namelist must be terminated with /
! * definitions & defaults:
!     TRACK%TIMESTART           = '19000101 000000'  ! Stop date for th
!     TRACK%TIMESTRIDE          = '0'                ! Time stride for
!     TRACK%TIMECOUNT           = '1000000000'       ! Number of time s
!     TRACK%TIMESPLIT           = 6                  ! [4(yearly),6(mon
! ------------------------------------------------------------------- !
&TRACK_NML
  TRACK%TIMESTART        =  '20100101 000000'
  TRACK%TIMESTRIDE       =  '3600'
/


! ------------------------------------------------------------------- !
! Define the content of the output file via FILE_NML namelist
!
! * namelist must be terminated with /
! * definitions & defaults:
!     FILE%PREFIX        = 'ww3.'            ! Prefix for output file na
!     FILE%NETCDF        = 3                 ! Netcdf version [3|4]
! ------------------------------------------------------------------- !
&FILE_NML
  FILE%NETCDF        = 4
/


! ------------------------------------------------------------------- !
! WAVEWATCH III - end of namelist                                     !
! ------------------------------------------------------------------- !
```

end of example input file (namelist form)

G.21 ww3_systrk

G.21.1 ww3_systrk.inp

start of example input file (traditional form)

```
$ ------------------------------------------------------------------- $
$ WAVEWATCH III Spatial and temporal tracking of wave systems         $
$-------------------------------------------------------------------- $
$ File name for raw partition data
$
  'partition.ww3'
$
$ First time level (yyyymmdd hhmmss), time increment and number of
$ time levels to process.
$
  20091122 000000  3600  4
$
$ Output type [1,3,4] [text file, netCDF version 3, netCDF version 4]
$ Note for NetCDF version 3 the TRKNC switch is needed and
$ for NetCDF version 4 the TRKNC and NC4 switches are needed.
   1
$
$ Wave tracking domain. First line: longitude limits, longitude interval
$ (NX-1); second line: latitude limits, latitude intervals (NY-1).
$
   100. 275. 175
     0.  55. 55
$
$ Parameters of tracking algorithm ---------------------------------- $
$   - dirKnob (deg), perKnob (s), hsKnob (m), wetPts (frac),
$     dirTimeKnob (deg), tpTimeKnob (s)
$   - seedLat, seedLon
$
 10. 1. 0.25 0.1 10. 1.
  0. 0.
$
$ Output points ----------------------------------------------------- $
$ Longitude, latitude. End with 0. 0. string on last line.
$
  222.54  40.75
  199.42  19.02
  205.94  23.55
```

```
  290.35  31.98
  347.60  48.70
  337.00  21.00
  197.94  24.32
  206.10  23.56
    0.    0.
$ ------------------------------------------------------------------- $
$ End of input file                                                   $
$ ------------------------------------------------------------------- $
```

end of example input file (traditional form)

G.22 ww3_uprstr

G.22.1 ww3_uprstr.inp

start of example input file (traditional form)

```
$ ------------------------------------------------------------------- $
$ WAVEWATCH III Update Restart input file                             $
$ ------------------------------------------------------------------- $
$                                                                     $
$ Time of Assimilation ----------------------------------------------- $
$ - Starting time in yyyymmdd hhmmss format.                          $
$                                                                     $
$ This is the assimilation starting time and has to be the same with  $
$ the time at the restart.ww3.                                        $
$                                                                     $
   19680607 120000                                                    $
$                                                                     $
$ Choose algorithm to update restart file                             $
$  UPDN for the Nth approach                                          $
$  The UPDN*, with N<2 the same correction factor is applied at all   $
$  the grid points                                                    $
$  UPDOC:: ELIMINATED                                                 $
$  UPDOF:: Option OF  All the spectra are updated with a constant     $
$          fac=HsAnl/HsBckg.                                          $
```

```
$          Expected input: PRCNTG, as defined at fac                    $
$  UPD1 :: ELIMINATED                                                   $
$  UPDN, with N>1 each gridpoint has its own update factor.             $
$  UPD2 :: Option 2    The fac(x,y,frq,theta), is calculated at each    $
$          grid point according to HsBckg and HsAnl                     $
$          Expected input the Analysis field, grbtxt format             $
$  UPD3 :: Option 3    The update factor is a surface with the shape    $
$          of the background spectrum.                                  $
$          Expected input: the Analysis field, grbtxt format and cap    $
$          for the last gross check.                                    $
$  UPD4 :: [NOT INCLUDED in this Version, Just keeping the spot]        $
$          Option 4    The generalization of the UPD3. The update       $
$          factor is the sum of surfaces which are applied on the       $
$          background spectrum.                                         $
$          The algorithm requires the mapping of each partition on the $
$          individual spectra; the map is used to determine the         $
$          weighting surfaces.                                          $
$          Expected input: the Analysis field, grbtxt format and the    $
$          functions(frq,theta) of the update to be applied.            $
   UPD3
$                                                                       $
$ PRCNTG is input for option 1 and it is the percentage of correction   $
$  applied  to all the gridpoints (e.g. 1.)                             $
$                                                                       $
   0.6754
$                                                                       $
$ PRCNTG_CAP is global input for option UPD2 and UPD3 and it is a cap   $
$ on the maximum correction applied to all the gridpoints (e.g. 0.5)    $
$                                                                       $
  10.
$                                                                       $
$ Name of the file with the SWH analysis from the DA system             $
$ suffix .grbtxt for text out of grib2 file.                            $
$                                                                       $
  anl.grbtxt
$                                                                       $
$ ---------------------------------------------------------------------- $
$ WAVEWATCH III EoF ww3_uprstr.inp                                      $
$ ---------------------------------------------------------------------- $
```

end of example input file (traditional form)

This page is intentionally left blank.