The University of Miami Wave Model (UMWM)
Version 2

Description and User's Manual

M.A. Donelan and M. Curcic

September, 2016

Rosenstiel School of Marine and Atmospheric Science

University of Miami

Contents

1 Model description

The University of Miami Wave Model (UMWM) is a prediction model for wave energy and wind stress on the interface between a liquid and a gas. It proceeds through a numerical solution of the wave energy balance equation on a horizontal 2-dimensional grid. The wave energy is a positive definite quantity in logarithmically spaced frequency (f) bins and uniformly spaced directional (ϕ) bins. The energy spectrum is carried as the wavenumber-directional surface elevation variance spectrum and every frequency at each location is identified with the theoretical wavenumber. Thus wave energy is a 5-dimensional quantity (x, y, k, ϕ, t). Time (t) evolution is done by integrating in time the energy balance equation:

$$\frac{\partial E'}{\partial t} + \frac{\partial(\dot{\mathbf{x}}E')}{\partial \mathbf{x}} + \frac{\partial(\dot{k}E')}{\partial k} + \frac{\partial(\dot{\phi}E')}{\partial \phi} = \rho_w g \sum_{i=1}^{n} S_i \tag{1.1}$$

where $E'(\mathbf{x}, k, \phi, t)$ is the energy spectrum, and $\dot{\mathbf{x}} = \mathbf{c}_g + \mathbf{u}$, where $\mathbf{c}_g$ is the group velocity and $\mathbf{u}$ is the current in the wave boundary layer. The second, third and fourth terms on the left-hand side are the advection terms in geographical, wavenumber and directional space, respectively. Advection in wavenumber space is non-zero only in case of changing ocean currents or moving bottom. Here, currents are considered to be quasi-stationary on the time scale of wave growth/decay. Thus, this term is neglected. Advection in directional space (refraction) is non-zero in case of variable water depth and/or inhomogeneous currents. $\sum_{i=1}^{n} S_i$ are the source/sink functions that act to grow/decay the waves locally. ρ_w is the liquid density and g is acceleration due to gravity.

The energy spectrum and the variance spectrum are related by:

$$E'(\mathbf{x}, k, \phi) = \rho_w g E(\mathbf{x}, k, \phi) \tag{1.2}$$

For convenience, the variance spectrum is the predicted quantity.

2   Source functions

The source functions S are parametric descriptions of the various phenomena that increase, decrease or interchange among wavenumbers the energy in the wavenumber spectrum. The wavenumber spectrum is evaluated in separable magnitude and direction bins. The phenomena relevant to the prediction of storm waves in water of arbitrary depth are:

(i) Input of energy and momentum from the wind and export of wave energy and momentum to the wind when the waves overrun or run against the wind;

(ii) Dissipation of wave energy at and near the surface due to viscosity, ambient turbulence and breaking;

(iii) Enhanced dissipation at and near the top and bottom interfaces due to shoaling;

(iv) Movement of energy to lower wavenumbers (down-shifting) due to nonlinear interactions including breaking;

(v) Enhanced dissipation due to straining by longer waves.

The theoretical and experimental justifications for the source functions corresponding to these phenomena are given in Donelan et al. (2012). Here the source functions are simply listed.

2.1   Wind input function S_{in}

Jeffreys' (1924, 1925) sheltering hypothesis leads to S_{in} of the form:

$$S_{in} = A_1 \left(U_{\lambda/2} \cos\theta - c - u\cos\phi - v\sin\phi \right) \left| U_{\lambda/2} \cos\theta - c - u\cos\phi - v\sin\phi \right| \frac{k\omega}{g} \frac{\rho_a}{\rho_w} E(k,\phi) \quad (2.1)$$

where θ is the angle between wind direction, ψ, and waves of wavenumber, k and direction, ϕ. A_1 is the sheltering coefficient.

S_{in} is positive (energy and momentum transferred from wind to waves) when $U_{\lambda/2}\cos\theta > c + u\cos\phi + v\sin\phi$, and negative (energy and momentum transferred from waves (swell) to wind) when $0 < U_{\lambda/2}\cos\theta < c + u\cos\phi + v\sin\phi$ or when the waves (swell) propagate against the wind, $\cos\theta < 0$. As waves approach full development S_{in} goes to zero; i.e. the direct wind forcing vanishes. The sheltering coefficient, which describes the strength of the source/sink function, is different depending on whether S_{in} is positive (wind sea) or negative, when the waves run before the wind or against it (swell). The wind velocity is that at one half wavelength above the surface up to the top of the logarithmic layer, which is usually taken to be 20 m in the field.

4

$$A_1 = \begin{cases} variable, & \text{if } U_{\lambda/2}\cos\theta > c + u\cos\phi + v\sin\phi, \ \text{wind sea} \\ 0.001, & \text{if } 0 < U_{\lambda/2}\cos\theta < c + u\cos\phi + v\sin\phi, \ \text{swell with wind} \\ 0.1, & \text{if } \cos\theta < 0, \ \text{swell against wind} \end{cases} \quad (2.2)$$

2.2 Wave breaking dissipation function S_{ds}

The wave breaking dissipation source function is strongly nonlinear in the saturation spectrum, $B(k,\phi) = k^4 E(k,\phi)$. In addition the dissipation is enhanced by the straining due to the velocity field of all longer waves and increased by $coth(kd)$ due to the plunging breakers that occur for small values of kd (wavenumber times depth).

$$S_{ds}(k,\phi) = -A_2 \coth(0.2kd)\left(1 + A_3\overline{\chi^2}(k,\phi)\right)^2 B^n(k,\phi)\omega E(k,\phi) \quad (2.3)$$

where $\overline{\chi^2}(k,\phi)$ is the sum-squared slope in direction ϕ of all waves longer than $\lambda(k)$. Coefficients A_2, A_3, and n have values 42, 360, and 2.4 respectively.

2.3 Dissipation by turbulence function S_{dt}

Turbulence mixing in the wave boundary layer attenuates waves. The wave dissipation due to ambient turbulence is:

$$S_{dt} = -A_4 u_{*w} k E(k,\phi) \quad (2.4)$$

where u_{*w} is the friction velocity in the water near the surface and $A_4 = 0.002$.

2.4 Dissipation by viscosity function S_{dv}

Some wave energy is converted directly to heat through the action of the viscosity of the liquid. Viscous dissipation in the surface sublayer prevents the growth of waves until a threshold wind speed is exceeded. The theoretical viscous dissipation rate, $4\nu k^2$, has been verified in the laboratory for a range of viscosities, ν (Donelan and Plant, 2009).

$$S_{dv} = -4\nu k^2 E(k,\phi) \quad (2.5)$$

where ν is the kinematic viscosity of the liquid. It is negligible in clean water for all but the shortest waves ($\lambda < 20$ cm).

2.5 The non-linear wave-wave interaction function S_{nl}

A quantity proportional to the energy dissipated in spilling is passed to longer waves in the next two lower wavenumber bins. The amount transferred decays exponentially with the square of the relative frequency separation:

$$S_{nl}(k, \phi) = A_5 \left[b_1 S_{sb}(k - \Delta k, \phi) + b_2 S_{sb}(k - 2\Delta k, \phi) - S_{sb}(k, \phi) \right] \tag{2.6}$$

where $b_1 = \exp(-16(\frac{\Delta f}{f})^2)$, $b_2 = \exp(-16(2\frac{\Delta f}{f})^2)$, and b_1 and b_2 are normalized such that $b_1 + b_2 = 1$. A_5 is determined internally s a function of frequency bin spacing. S_{sb} is the wave breaking dissipation function due to spilling only: $S_{sb} = S_{ds} / \coth(kd)$.

2.6 The bottom friction function S_{bp}

The bottom friction function is related to orbital velocity at the bottom and the roughness of the bed. Komen et al. (1994) give the following form for a sandy bed:

$$S_{bf} = -G_f \frac{k}{\sinh 2kd} E(k, \phi) \tag{2.7}$$

where the roughness factor, G_f varies from 0.001 to 0.01 m/s, depending on bed roughness.

2.7 The bottom percolation function S_{bp}

On a porous bed the percolation of flow through the bed induces wave energy dissipation. The dissipation rate due to percolation is given by Shemdin et al. (1978):

$$S_{bp} = -G_p \frac{k}{\cosh^2 kd} E(k, \phi) \tag{2.8}$$

where the permeability factor, G_p varies from 0.0006 to 0.01 m/s, depending on sand grain size.

3 Stress calculation

3.1 Momentum flux from wind

Energy E and momentum M in the wave field are related by the phase speed:

$$E = Mc \tag{3.1}$$

The form drag components of the wind on the waves τ_x and τ_y are calculated from the wind input source function:

$$\tau_x = \rho_w g \int_{-\pi}^{\pi} \int_{k_{min}}^{k_{max}} \frac{S_{in}}{c} \cos \phi \, k \, dk \, d\phi \tag{3.2}$$

$$\tau_y = \rho_w g \int_{-\pi}^{\pi} \int_{k_{min}}^{k_{max}} \frac{S_{in}}{c} \sin \phi \, k \, dk \, d\phi \tag{3.3}$$

The form drag coefficient, Cd_f is then calculated:

$$Cd_f = \frac{\sqrt{\tau_x^2 + \tau_y^2}}{\rho_a U_z^2} \tag{3.4}$$

where U_z is the wind speed at the measured or modeled height z. The form drag is in the wave direction ϕ. The skin drag coefficient Cd_s, in the absence of waves, is in the wind direction ψ, and is computed from Von Karman's relationship:

$$Cd_s = \frac{u_*^2}{U^2(z)} = \frac{\kappa^2}{\left[\log \left(\frac{z}{z_0} \right) \right]^2} \tag{3.5}$$

where u_* is the friction velocity, $U(z)$ is wind speed at height z and κ is the Von Karman's constant. The roughness length z_0 is determined by the scale of the molecular sublayer for smooth flow:

$$z_0 = 0.132 \frac{\nu_a}{u_*} \tag{3.6}$$

where ν_a is the kinematic viscosity of air. In the presence of waves Cd_s is reduced due to the sheltering of the surface in the lee of steep waves. The degree of sheltering of the skin is taken to be proportional to the ratio of total drag to skin drag and may be as large as 50% corresponding to full sheltering of the lee face of each wave. The algorithm is as follows:

$$Cd_s^{new} = \frac{Cd_s^{old}}{3} \left(1 + 2 \frac{Cd_s^{old}}{Cd_s^{old} + Cd_f} \right) \tag{3.7}$$

In order to obtain the correct form stress magnitude, the high-frequency limit should be equal to or greather than 2 Hz. A power law wind-speed dependent tail is appended to the spectrum beyond the high-frequency limit. This additional form stress component, $\tau(tail)$, is in the wind direction, as is the skin stress, $\tau(skin)$.

These (form stress on the resolved waves, form stress on the spectral tail, and skin stress) are the stress components of the wind on the surface. They are the mechanical couplers with the atmospheric model. Both skin and form stress components, as well as total drag coefficient Cd, are part of the standard gridded output of UMWM.

3.2 Momentum flux into ocean

Similarly as momentum flux from wind, we can integrate source dissipation functions over the spectrum to obtain momentum fluxes into ocean top and bottom:

$$\tau_x^{OT} = \rho_w g \int_{-\pi}^{\pi} \int_{k_{min}}^{k_{max}} \frac{-S_{ds} - S_{dt} - S_{dv}}{c} \cos\phi \, k \, dk \, d\phi + \tau_x(tail) + \tau_x(skin) \tag{3.8}$$

$$\tau_y^{OT} = \rho_w g \int_{-\pi}^{\pi} \int_{k_{min}}^{k_{max}} \frac{-S_{ds} - S_{dt} - S_{dv}}{c} \sin\phi \, k \, dk \, d\phi + \tau_y(tail) + \tau_y(skin) \tag{3.9}$$

$$\tau_x^{OB} = \rho_w g \int_{-\pi}^{\pi} \int_{k_{min}}^{k_{max}} \frac{-S_{bf} - S_{bp}}{c} \cos\phi \, k \, dk \, d\phi \tag{3.10}$$

$$\tau_y^{OB} = \rho_w g \int_{-\pi}^{\pi} \int_{k_{min}}^{k_{max}} \frac{-S_{bf} - S_{bp}}{c} \sin\phi \, k \, dk \, d\phi \tag{3.11}$$

All momentum fluxes are defined as positive downward - i.e breaking of waves in x-direction would yield a positive value of momentum flux in that direction. These fluxes are part of UMWM standard gridded output.

4 Numerical approaches

4.1 Spatial discretization

The time evolution of the variance spectrum due to advection in Cartesian projection is given by:

$$\frac{\partial E}{\partial t} = -\frac{\partial[(c_g \cos\phi + u)E]}{\partial x} - \frac{\partial[(c_g \sin\phi + v)E]}{\partial y} - \frac{\partial(\dot{\phi}E)}{\partial\phi} \tag{4.1}$$

where u and v are ocean current components in x and y, respectively, and $\dot{\phi}$ is the rotation rate. Both geographical propagation and refraction terms are discretized using first order upstream differencing. This scheme is positive-definite, quantity conserving, implicitly diffusive and computationally efficient. A certain amount of diffusion is desirable in order to avoid swell separation between discrete directional and frequency bins. A spatial differencing operator is then discretized as:

$$\frac{\partial(\dot{x}E)}{\partial x} \approx \frac{\Phi_{i+1/2} - \Phi_{i-1/2}}{\Delta x} \tag{4.2}$$

where i is a discrete index along dimension x. Fluxes at cell edges $\Phi_{i+1/2}$ and $\Phi_{i-1/2}$ are defined as:

$$\Phi_{i+1/2} = \frac{\dot{x}_{i+1/2} + |\dot{x}_{i+1/2}|}{2}E_i + \frac{\dot{x}_{i+1/2} - |\dot{x}_{i+1/2}|}{2}E_{i+1} \tag{4.3}$$

$$\Phi_{i-1/2} = \frac{\dot{x}_{i-1/2} + |\dot{x}_{i-1/2}|}{2}E_{i-1} + \frac{\dot{x}_{i-1/2} - |\dot{x}_{i-1/2}|}{2}E_i \tag{4.4}$$

and

$$\dot{x}_{i\pm1/2} = \frac{\dot{x}_i + \dot{x}_{i\pm1}}{2} \tag{4.5}$$

The above treatment of flux differencing ensures upstream-definiteness. For propagation in two-dimensional space, the stability of the scheme is ensured for:

$$\mu = \frac{\dot{x}\Delta t}{\min(\Delta x, \Delta y)} < \frac{1}{\sqrt{2}} \tag{4.6}$$

where μ is the Courant number. Depending on the choice of number of directional bins, the stability criterion is more permissive:

$$\mu = \frac{\dot{x}\Delta t}{\min(\Delta x, \Delta y)} < \cos\left(\frac{\pi}{4} - \frac{\Delta\phi}{2}\right) \tag{4.7}$$

where $\Delta\phi$ is the directional bin size. To ensure that (4.7) holds, the number of directions must be divisible by 8.

For ocean grid cells next to the land, an open boundary condition is applied (energy can freely propagate into land). Same is applied for domain edges, except that there can be incoming wave energy from the boundaries, if provided by the user. In the case of global domain simulation, periodic boundary conditions are applied at East and West domain edges.

The rotation rate $\dot{\phi}$ in the refraction term is evaluated as:

$$\dot{\phi} = \frac{\partial(c\sin\phi + v)}{\partial x} - \frac{\partial(c\cos\phi + u)}{\partial y} \tag{4.8}$$

The change due to refraction is then computed using (4.2)-(4.5) with $\dot{x}$ replaced by $\dot{\phi}$. Positive and negative values of $\dot{\phi}$ correspond to counter-clockwise and clockwise rotation of energy, respectively. The stability constraint for the refraction term is the same as for one-dimensional advection:

$$\mu = \frac{\dot{\phi}\Delta t}{\Delta\phi} < 1 \tag{4.9}$$

For most domain cells, the allowed refraction time step is larger than the advective step given in (4.7). In case that condition (4.9) is violated, which can occur on sharp bathymetric or current gradients, the rotation at these points is limited so that $\mu = 1$. This affects the solution insignificantly, while maintaining computational efficiency. Because the domain is periodic in directional space, there is no need for boundary conditions.

4.2 Time discretization

Once all the source terms in (1.1) have been evaluated, $E(k, \phi)$ is integrated forward in time. We evaluate the contribution from source and advection terms separately:

$$\frac{\partial E}{\partial t} = \left(\frac{\partial E}{\partial t}\right)_s + \left(\frac{\partial E}{\partial t}\right)_a \tag{4.10}$$

The contribution from source terms can be written as:

$$\left(\frac{\partial E}{\partial t}\right)_s = \sum_{i=1}^{n} S_i^* E \tag{4.11}$$

where S_i^* is just S_i/E. Then, by integrating (4.11) over a finite time interval Δt, a solution is available in the form of:

$$E_s^{n+1} = E^n exp\left(\sum_{i=1}^{n} S_i^*\Delta t\right) \tag{4.12}$$

The time increment Δt is dynamically computed so that the variance spectrum E can only grow by a pre-determined, finite factor:

$$\frac{E_s^{n+1}}{E^n} = exp\left(\sum_{i=1}^{n} S_i^* \Delta t\right) < r \tag{4.13}$$

where r is usually set between 1.5 and 2. Lower values of r will draw E closer to the solution attractor.

Then, a time-splitting approach is used to achieve a more stable integration:

$$E^* = \frac{E^n + E_s^{n+1}}{2} \tag{4.14}$$

E^* is used to compute the advection and refraction terms described in the previous section. Finally, their contribution is evaluated by simple forward-Euler differencing:

$$E^{n+1} = E_s^{n+1} - \Delta t \left[\frac{\partial[(c_g \cos\phi + u)E^*]}{\partial x} + \frac{\partial[(c_g \sin\phi + v)E^*]}{\partial y} + \frac{\partial(\dot{\phi}E^*)}{\partial\phi}\right] \tag{4.15}$$

The above approach is applied to the prognostic part of the spectrum. A cut-off frequency, f_c, which separates the prognostic and diagnostic parts, is proportional to the peak frequency of the fully-developed Pierson-Moskowitz spectrum (Pierson and Moskowitz, 1964):

$$f_c = 4f_{PM} = \frac{0.53g}{U_{10}} \tag{4.16}$$

For all bins higher than f_c, the waves are assumed to be: in equilibrium with the wind – traveling centered on the wind direction – and their spectral densities are established from a balance of wind input and dissipation. This approach is justified by the presumption that the quasi-equilibrium range is wider in higher wind conditions.

5 Software implementation

UMWM source code is written in standard Fortran 90 programming language, and takes advantage of Unidata's NetCDF standard library for I/O (required), and parallel processing through the Message Passing Interface (MPI, optional). It is written in readable and transparent form (i.e. free of low-level instructions), and is well documented. The model may be defined on any structured, curvilinear grid. Most common applications are on a Cartesian and spherical (latitude-longitude) grid. More general and un-structured grids are planned to be implemented in future versions of the model. A choice is available between limited-area (regional) and global simulation. In the latter case, a periodic boundary condition in x-direction is applied automatically, and no additional action is required by the user.

In case of parallel processing, model's partitioning scheme (tile distribution) is designed in such a way that any number of processors is allowed for running the model. This may prove to be useful when a very limited number of processors is available, or in the case of coupled modeling where fine-tuning the work-load distribution between model components can be beneficial for computational efficiency.

The user is encouraged to look into the code for information that is not provided in this document. The model is open-source and is licensed under a GPL General Public License Version 3, `http://www.gnu.org/copyleft/gpl.html`.

6   Installation and setup

In order to successfully convey a numerical simulation with UMWM, a procedure is usually as follows:

1) Obtaining the software package;

2) Compiling the source code;

3) Setting up the simulation;

4) Running the model;

5) Reading model output data.

In general, steps 1 and 2 need to be done only once. A working Fortran compiler and NetCDF library must be provided by the user. Re-compiling the source code (step 2) is necessary only if change has been made to any of the source files. Naturally, step 3 needs to be repeated if the setup of the experiment is to be changed. If only input forcing files are changed, one needs only to re-run the model (step 4). Description of the output data files is provided in section 9.

6.1   Obtaining the software package

The current version of UMWM can be downloaded from this URL:

`http://rsmas.miami.edu/groups/umwm/download.html`

On a UNIX-like system (Linux, Mac OS X or other UNIX), unpack the downloaded file by typing `tar xzf umwm-1.x.x.tar.gz` (`1.x.x` should be replaced with the appropriate version number). This will create the directory `umwm`. In addition, UMWM requires NetCDF Fortran libraries for input and output. NetCDF libraries and installation instructions can be downloaded from:

`http://www.unidata.ucar.edu/downloads/netcdf/index.jsp`

Also, a common build tool `make` or `gmake` is recommended for easier compilation. `make` is included by default in most UNIX/Linux distributions, but may need to be installed separately on a Microsoft Windows system.

The directory tree in the parent `umwm` directory is:

`clean`: A script that cleans all output files from the `output/` directory.

`config`: Directory which contains compilation setup files.

`COPYING`: Licensing.

`input`: Directory where model input files need to be placed.

`INSTALL`: Short installation instructions.

`Makefile`: Compilation rules.

`namelists`: Directory which contains namelist input files for the model.

`output`: Directory where all model output will be written.

`restart`: Directory in which all model restart files are being read and written.

`src`: Directory which contains the source code of the model.

`tools`: Auxiliary pre-processing programs.

The next section describes the steps necessary to compile the model source code.

6.2 Compiling the source code

Before doing a simulation, the model code needs to be compiled. The source code is located in the `src/` directory.

The user must set up compiler and NetCDF library variables. This is done in the `config/` directory. For example, configuration files `intel` and `intel.mpi` contain configuration settings for the Intel Fortran Compiler, for serial and parallel execution, respectively. The appropriate configuration file should be then copied or linked to a file named `umwm.config` in the same directory. This is the file that is being read at compile-time for both the model and the auxiliary tools.

If you have successfully compiled and run UMWM on a platform that was not provided originally in the `config/` directory, please send an e-mail to `milan@orca.rsmas.miami.edu`, and we will include the configuration file for that platform in the next release.

Once the `umwm.config` file has been set-up, from the main UMWM directory type:

`make umwm`

to build the model code,

`make tools`

to build auxiliary pre-processing programs (see section 7), or just type:

`make`

to build both the model and auxiliary programs. The model executable, `umwm`, will appear in the main directory, and auxiliary programs executables will appear in the `tools/` directory.

6.3 Setting up the simulation

Once the executable `umwm` has been generated, the user should set up a desired experiment. The first step is to edit the main UMWM namelist that defines model parameters. The namelist is located in `namelists/main.nml`. An example `main.nml` file is shown below:

```
&DOMAIN
  isGlobal    = .F.                  ! Global (.T.) or regional (.F.)
  mm          = 399                  ! Domain size in x
  nm          = 359                  ! Domain size in y
  om          = 37                   ! Number of frequency bins
  pm          = 36                   ! Number of directions
  fmin        = 0.0313               ! Lowest frequency bin [Hz]
  fmax        = 2.0                  ! Highest frequency bin [Hz]
  fprog       = 0.5                  ! Highest prognostic frequency bin [Hz]
  startTimeStr = '2008-09-08_00:00:00' ! Simulation start time
```

```
  stopTimeStr  = '2008-09-13_00:00:00' ! Simulation end time
  dtg          = 3600                   ! Global (I/O) time step [s]
  restart      = .F.                    ! Restart from file
/
&PHYSICS
  g         = 9.80665 ! Gravitational acceleration [m/s^2]
  nu_air    = 1.56E-5 ! Kinematic viscosity of air [m^2/s]
  nu_water  = 0.9E-6  ! Kinematic viscosity of water [m^2/s]
  sfct      = 0.074   ! Surface tension [N/m]
  kappa     = 0.4     ! Von Karman constant
  z         = 10.     ! Height of the input wind speed [m]
  gustiness = 0.0     ! Random wind gustiness factor (should be between 0 and 0.2)
  dmin      = 10.     ! Depth limiter [m]
  explim    = 1.1     ! Exponent limiter (0.69 ~ 100% growth)
  sin_fac   = 0.11    ! Input factor from following winds
  sin_diss1 = 0.10    ! Damping factor from opposing winds
  sin_diss2 = 0.001   ! Damping factor from swell overrunning wind
  sds_fac   = 42.     ! Breaking dissipation factor
  sds_power = 2.4     ! Saturation spectrum power
  mss_fac   = 360.    ! Mean-square-slope adjustment to Sds
  sdt_fac   = 0.002   ! Dissipation due to turbulence factor
  sbf_fac   = 0.003   ! Bottom friction coefficient [m/s]
  sbp_fac   = 0.003   ! Bottom percolation coefficient [m/s]
/
&GRID
  gridFromFile  = .T.   ! Set to .T. if lon/lat fields are input from file
  delx          = 12000 ! Grid spacing in x [m] if gridFromFile = .F.
  dely          = 12000 ! Grid spacing in y [m] if gridFromFile = .F.
  topoFromFile  = .T.   ! Set to .T. to input bathymetry from file
  dpt           = 1000  ! Constant water depth [m] if topoFromFile = .F.
  fillEstuaries = .F.   ! Set to .T. to fill cells with 3 land neighbours
  fillLakes     = .F.   ! Set to .T. to fill user chosen seas/lakes
/
&FORCING
  winds         = .T. ! Wind input from file
  currents      = .F. ! Currents input from file
  air_density   = .T. ! Air density input from file
  water_density = .F. ! Water density input from file
/
&FORCING_CONSTANT
  wspd0         = 25   ! Wind speed [m/s]
  wdir0         = 0.   ! Wind direction [rad]
  uc0           = 0.   ! x-component ocean current [m/s]
  vc0           = 0.   ! y-component ocean current [m/s]
  rhoa0         = 1.2  ! Air density [kg/m^3]
  rhow0         = 1025 ! Water density [kg/m^3]
/
&OUTPUT
```

```
  outgrid       = 1   ! Gridded output interval   [hours]
  outspec       = 0   ! Spectrum output interval [hours]
  outrst        = 0   ! Restart output interval   [hours]
  xpl           = 160 ! Grid cell in x for stdout (screen)
  ypl           = 160 ! Grid cell in y for stdout (screen)
  stokes        = .T. ! Output Stokes drift velocity fields
/
&STOKES
  depths        = 0 1 2 3 4 5 10 15 20 25 30 35 40 50 60 70 80 90 100
/
```

All simulation parameters in `namelists/main.nml` are summarized below.

6.3.1 DOMAIN namelist

The `DOMAIN` namelist contains main parameters that define a wave model simulation. In particular, these include domain grid size in geographical and frequency-directional space and time limits of the simulation. Full list of `DOMAIN` nameslist parameters follow.

`isGlobal`: A boolean value describing whether the simulation is global or regional. If set to `.TRUE.`, the domain will be periodic in x.

`mm`: Domain size in x.

`nm`: Domain size in y.

`om`: Total number of discrete frequency bins. See `fmin` and `fmax` below.

`oa`: Highest prognostic frequency bin index. If set to zero, the cutoff frequency is computed dynamically and equals 4 times Pierson-Moskowitz peak frequency.

`pm`: Total number of discrete directional bins. For small to medium-size real case basins (e.g. Mediterranean, Gulf of Mexico), at least 24 is recommended. For larger basins (oceans) or global simulations, set to at least 32. Must be divisible by 4. A value of pm divisible by 8 is recommended for optimal propagation properties.

`fmin`: Minimum frequency [Hz]. Determines the longest wave resolved by the model. Domain and application specific. A value between 0.03 Hz and 0.05 Hz is commonly used for real case larger scale basins.

`fmax`: Maximum frequency [Hz]. Determines the shortest wave resolved by the model. Domain and application specific. Typical values for regional or global wave forecasting is 0.5 Hz or larger. 2.0 Hz (or larger) is required for accurate stress estimation.

`fprog`: Highest frequency [Hz] in the prognostic range of the spectrum. Can be used to limit the prognostic range if the cutoff frequency determined by local wind speed (see section 4.2 for more information) is too high, resulting in short time steps. If this is not desired, set to the value of `fmax`.

`startTimeStr`: A character string in the form of `YYYY-MM-DD_hh:mm:ss` that determines the start time of the simulation. See `restart` below.

stopTimeStr: A character string in the form of YYYY-MM-DD_hh:mm:ss that determines the stop time of the simulation.

dtg: An integer that determines global time step of the wave model in seconds. If forcing data is to be input from files (see forcing options below), *dtg* corresponds to the input time step. *dtg* is the shortest time interval at which output to file can be made (see output options below).

restart: A boolean that determines if the model is to be initiated from initial conditions written in a restart file. If set to .FALSE., the model will be initiated from a calm state. If set to .TRUE., a valid restart file umwmrst_YYYY-MM-DD_hh:mm:ss.nc must be present at run time in the restart/ directory. A valid restart file must have mm, nm, om, pm, fmin, fmax same as the current simulation setup. For more information about restarting UMWM from a file, see section 8.1.

6.3.2   PHYSICS namelist

The PHYSICS namelist contains parameters that govern the physical processes relevant to wave energy evolution.

g: Gravitational acceleration $[\frac{m}{s^2}]$.

nu_air: Kinematic viscosity of air $[\frac{m^2}{s}]$.

nu_water: Kinematic viscosity of water $[\frac{m^2}{s}]$.

sfct: Water surface tension $[\frac{N}{m}]$.

kappa: Von Karman constant.

z: Height of the input wind speed $[m]$.

gustiness: Addition of a stochastic component to x and y wind speed components. The value of this parameter is a maximum fraction of the wind speed to be added or substracted. For example, setting gustiness to 0.2 will allow wind speed components to be randomly modulated by up to 20%. This option may be used to introduce wind fluctuations when available wind forcing files are on coarse temporal and spatial resolutions. Must be positive, and should not be larger than 0.2.

dmin: Depth limiter $[m]$. Must be larger than 0. All sea points shallower than dmin will be set to this value. Values of a few meters or fractions of a meter are allowed, but may lead to a shorter time step and longer model integration time.

sin_fac: Wave growth factor from following winds. See A_1 in (2.1) and (2.2).

sin_diss1: Dissipation factor from opposing wind input. See A_1 in (2.1) and (2.2).

sin_diss2: Dissipation factor from swell overrunning wind. See A_1 in (2.1) and (2.2).

sds_fac: Dissipation factor from deep-water breaking waves (spilling). See A_2 in (2.3).

sds_power: Exponent (power index) of the saturation spectrum, equation (2.3).

mss_fac: Sum-square slope adjustment to S_{ds}. See A_3 in (2.3).

`sdt_fac`: Dissipation factor due to turbulence in the wave boundary layer (ocean top). See A_4 in (2.4).

`sbf_fac`: Bottom friction coefficient $[\frac{m}{s}]$. See G_f in (2.7). Typically with values between 0.001 and $0.01m/s$, depends on bed roughness.

`sbp_fac`: Bottom percolation coefficient $[\frac{m}{s}]$. See G_p in (2.8). Typically with values between 0.0006 and $0.01m/s$, depends on sand grain size.

6.3.3 GRID namelist

`gridFromFile`: A boolean parameter that determines whether grid cell size Δx and Δy are going to be computed from longitude and latitude fields in `input/umwm.gridtopo`. If set to `.FALSE.`, Δx and Δy are constant, and they are input as `delx` and `dely` below.

`delx`: Grid spacing in x $[m]$ if `gridFromFile` is set to `.FALSE.`. Must be larger than 0.

`dely`: Grid spacing in y $[m]$ if `gridFromFile` is set to `.FALSE.`. Must be larger than 0.

`topoFromFile`: When set to `.TRUE.` bathymetry field is to be input from file (`input/umwm.gridtopo`, netCDF variable `z`). If set to `.FALSE.`, bathymetry is constant and has the value specified by namelist variable `dpt` below.

`dpt`: Constant water depth $[m]$ if topoFromFile is set to `.FALSE.`. Must be larger than 0.

`fillEstuaries`: If set to `.TRUE.`, all sea points that have 3 land neighbors will be masked iteratively as land. Use to reduce computational time if wave simulation in narrow channels or estuaries is not desired. Do not use if running a coupled atmosphere-wave simulation where atmosphere and wave model have identical sea/land mask fields.

`fillLakes`: If set to `.TRUE.`, enclosed seas/lakes marked by the user will be filled with land mask. Do not use if running a coupled atmosphere-wave simulation where atmosphere and wave model have identical sea/land mask fields. For more information about how to specify and fill enclosed seas that are not desired in the wave simulation, see section 8.2.

6.3.4 FORCING namelist

The `FORCING` namelist contains switches that control input of forcing fields from file. There is one switch for each forcing field. If any of the switches is set to `.FALSE.`, a `FORCING_CONSTANT` namelist will be read. If any of the switches is set to `.TRUE.`, input forcing file(s) will be read.

`winds`: Wind speed $[\frac{m}{s}]$ input from file switch. If set to `.FALSE.`, parameters `wspd0` and `wdir0` in the `FORCING_CONSTANT` namelist will provide the values for constant wind speed and wind direction (mathematical convention), respectively.

`currents`: Ocean surface currents $[\frac{m}{s}]$ input from file switch. If set to `.FALSE.`, parameters `uc0` and `vc0` in the `FORCING_CONSTANT` namelist will provide the values for constant x and y components of ocean surface currents, respectively.

`air_density`: Surface air density $[\frac{kg}{m^3}]$ input from file switch. If set to `.FALSE.`, parameter `rhoa0` in the `FORCING_CONSTANT` namelist will provide the value for constant surface air density field.

`water_density`: Surface water density $[\frac{kg}{m^3}]$ input from file switch. If set to `.FALSE.`, parameter `rhow0` in the `FORCING_CONSTANT` namelist will provide the value for constant surface water density field.

6.3.5 FORCING_CONSTANT namelist

Parameters in this namelist are used only in the case where any of the forcing fields are not provided in the input files.

`wspd0`: Wind speed $[\frac{m}{s}]$. Used if `winds = .FALSE.`.

`wdir0`: Wind direction $[rad]$, mathematical convention. Used if `winds = .FALSE.`.

`uc0`: Ocean surface current, x-component $[\frac{m}{s}]$. Used if `currents = .FALSE.`.

`vc0`: Ocean surface current, y-component $[\frac{m}{s}]$. Used if `currents = .FALSE.`.

`rhoa0`: Surface air density $[\frac{kg}{m^3}]$. Used if `air_density = .FALSE.`.

`rhow0`: Surface water density $[\frac{kg}{m^3}]$. Used if `water_density = .FALSE.`.

6.3.6 OUTPUT namelist

Parameters in this namelist control the model output - gridded, spectrum, and restart. Gridded output contains integrated 2-dimensional quantities of the spectrum. Fields from the whole domain are stored in gridded output files. Spectrum output is defined at individual geographical locations in the domain (one file per location) and contain full wavenumber-directional spectrum and source functions S_{in}, S_{ds} and S_{nl}. Restart output files contain snapshots of the wave spectrum from the whole domain. They are used for continuing a model simulation at a later time (see section 8.1).

`outgrid`: Gridded output interval [hours]. Allowed values are 0, 1, 2, 3, 4, 6, 8, 12 and 24. If `outgrid` is larger than 0, gridded output files in the form of `umwmout_YYYY-MM-DD_hh:mm:ss.nc` will appear in the `output/` directory, where `YYYY-MM-DD_hh:mm:ss` denotes model time of the output.

`outspec`: Spectrum output interval [hours]. Allowed values are 0, 1, 2, 3, 4, 6, 8, 12 and 24. If `outspec` is larger than 0, another input file, `namelists/spectrum.nml` will be read. Although technically not a Fortran namelist file, it works in a similar way. An example spectrum input file may look like this:

```
LL
-89.66 25.89 NDBC42001
-93.67 25.79 NDBC42002
-96.66 25.96 NDBC42020
-94.41 29.23 NDBC42035
```

```
-88.80 30.09 NDBC42007
```

The first line of the file must contain either `XY` or `LL`. This 2-character string describes whether the coordinates of spectrum output location are given in model grid indices or in longitude and latitude, respectively. In case of `XY`, spectrum at exact specified grid cell will be output. In case of `LL`, spectrum at grid cell nearest to specified location will be output. The following lines each contain 3 elements: x and y coordinate (or longitude and latitude) and location identifier (character string, up to 40 characters, no spaces). A spectrum output file in the form of `umwmspc_ID_YYYY-MM-DD_hh:mm:ss.nc` will appear in the `output/` directory, where `ID` is the location identifier specified in `namelists/spectrum.nml`, and `YYYY-MM-DD_hh:mm:ss.nc` is the initial time of the simulation and has the same value as `startTimeStr`. All output times during the simulation are stored in the same file. Each requested spectrum location is stored in a separate file.

`outrst`: Restart output interval [hours]. Allowed values are `0`, `1`, `2`, `3`, `4`, `6`, `8`, `12` and `24`. If `outrst` is larger than 0, restart files in the form of `umwmrst_YYYY-MM-DD_hh:mm:ss` will appear in the `restart/` directory. These files can be used later to re-start a simulation in case of interruption or abnormal termination of the model (see section 8.1).

`xpl`: Grid cell index in x for stdout (screen).

`ypl`: Grid cell index in y for stdout (screen).

`stokes`: A boolean that controls the calculation and output of Stokes drift. If set to `.TRUE.`, the `STOKES` namelist (see next section) will be read.

6.3.7   STOKES namelist

If the parameter `stokes` in the `OUTPUT` namelist is set to `.TRUE.`, this namelist will be read by the model. It contains only one entry, `depths`, which is an array of depths at which the Stokes drift will be computed. Depth values are defined as positive numbers, e.g.:

```
&STOKES
  depths = 0 1 2 3 4 5 10 15 20 25 30 35 40 50 60 70 80 90 100
/
```

The number of levels at which Stokes drift velocities are to be computed is arbitrary, and can be chosen by the user.

6.4   Input files

For a typical regional-scale simulation, UMWM requires a grid and topography file `umwm.gridtopo` and input forcing files in the form of `umwmin_YYYY-MM-DD_hh:mm:ss.nc`. Both grid and topography file and forcing files must be written in NetCDF standard format, and must be present in the `input/` directory.

The `umwm.gridtopo` must contain 2-dimensional fields of longitude, latitude, and terrain elevation under names `lon`, `lat`, and "z", respectively. Longitude and latitude fields are used internally to calculate grid cell size fields Δx and Δy. Terrain elevation field `z` is defined as positive upward from mean sea level.

A forcing file may contain some or all of the following 2-dimensional fields:

uw: x-component of wind $[\frac{m}{s}]$. Used if the `winds` switch in the `OUTPUT` namelist is set to `.TRUE.`.

vw: y-component of wind $[\frac{m}{s}]$. Used if the `winds` switch in the `OUTPUT` namelist is set to `.TRUE.`.

uc: x-component of ocean surface currents $[\frac{m}{s}]$. Used if the `currents` switch in the `OUTPUT` namelist is set to `.TRUE.`.

vc: y-component of ocean surface currents $[\frac{m}{s}]$. Used if the `currents` switch in the `OUTPUT` namelist is set to `.TRUE.`.

rhoa: Air density $[\frac{kg}{m^3}]$. Used if the `air_density` switch in the `OUTPUT` namelist is set to `.TRUE.`.

rhow: Water density $[\frac{kg}{m^3}]$. Used if the `water_density` switch in the `OUTPUT` namelist is set to `.TRUE.`.

The forcing files required during model run time must be present in the `input/` directory.

7 Pre-processing tools

Several pre-processing tools are available to the user for generation of input grid and bathymetry files for UMWM, as well as input forcing files from WRF (Weather Research and Forecasting atmospheric model) output files. The program source files are located in the `tools/src/` directory. After successful compilation, the program executable files will be located in the `tools/` directory.

Individual pre-processing tools for UMWM are described in the following sections.

7.1 umwm_gridgen

Program `umwm_gridgen` may be used for a quick generation of a UMWM grid file (`umwm.grid`) in a regular spherical (lat-lon) projection. Before running the program, a `tools/namelists/gridgen.nml` namelist file must be configured:

```
&gridgen
  idm  = 400    ! Grid size in x (East)
  jdm  = 300    ! Grid size in y (North)
  dlon = 0.04   ! Grid increment in degrees longitude
  dlat = 0.04   ! Grid increment in degrees latitude
  lon0 = 162    ! Longitude of SW corner point; range is [-180,180]
  lat0 = 10     ! Latitude of SW corner point; exclusive range is (-90,90)
/
```

Once all the parameters in the namelist have been set up, the program can be run from the `tools/` directory by typing `umwm_gridgen`. This will generate a UMWM grid file `umwm.grid`. This file can then be used as input to the wave model itself, or to `umwm_topogen` for bathymetry field generation.

7.2 umwm_topogen

Once a UMWM grid file, `umwm.grid` has been generated, program `topogen` may be used to generate the bathymetry field for input to the wave model. The namelist file that needs to be editted is `tools/namelists/topogen.nml`:

```
&topogen
  umwmInputFile = 'umwm.grid'
  topoInputFile = 'ETOPO1_Ice_c_gmt4.grd'
  useSeamask    = .F.
/
```

`topogen` takes two files as input. First is a target grid file for which a bathymetry field is desired, in this case, UMWM grid file `umwm.grid`, and the second is the raw topography file to be read.

There are several requirements about the format of input files. Both need to be in netCDF format, with spatial dimensions defined as `x` and `y`. `umwmInputFile` must have 2-dimensional longitude and latitude fields defined as `lon` and `lat`, respectively. `topoInputFile` must have 1-dimensional longitude and latitude fields defined as `lon` and `lat`, respectively, and a 2-dimensional topography field (positive - up, negative - down) defined as `z`. This format is compatible with, e.g. ETOPO

1-minute topography in netCDF format by NOAA National Geophysical Data Center (topography data can be downloaded from `http://www.ngdc.noaa.gov/mgg/global/global.html`).

The third parameter in the namelist, `useSeamask`, may be used only if there is a `seamask` field defined in `umwmInputFile` (e.g. if the grid file was generated by `wrf2umwmgrid`, or added by user). If set to `.TRUE.`, the bathymetry field will be modified at points where its values and seamask do not match. This may be useful when coupling UMWM with an atmosphere or ocean model, and identical seamasks are desired.

7.3   wrf2umwmgrid

`wrf2umwmgrid` is a convenience program for extraction of 2-dimensional longitude, latitude and seamask fields from a WRF output file. The program takes a WRF file name (path) as a command line argument, e.g.:

`wrf2umwmgrid wrfout_d01_2008-09-06_00:00:00`

This will generate a UMWM grid file `umwm.grid`. This file can then be used as input to the wave model itself, or to `umwm_topogen` for bathymetry field generation.

`wrf2umwmgrid` will operate on any `wrfout*`, `wrfrst*` or `wrfinput*` file.

7.4   wrf2umwmin

Similarly to `wrf2umwmgrid`, this program can be used to convert a WRF output file to a UMWM input forcing file. Since WRF output does not contain any ocean fields, only wind components and air density may be computed using these files. The program takes a WRF file name (path) as a command line argument, e.g.:

`wrf2umwmin wrfout_d01_2008-09-06_00:00:00`

This will generate a UMWM input forcing file `umwmin_d01_2008-09-06_00:00:00.nc`. The air density field is calculated using the ideal gas law for moist air.

`wrf2umwmin` will operate on any `wrfout*`, `wrfrst*` or `wrfinput*` file.

8 Running the model

Once namelists have been configured and input files are provided, the user can start the simulation by executing `umwm`. If the model was compiled in the MPI mode, refer to the documentation of the MPI implementation about executing programs in parallel. On a UNIX/Linux system with a common modern MPI implementation (e.g. MPICH2 or OpenMPI), one would type:

```
mpiexec -n 16 umwm
```

This command would execute `umwm` on 16 parallel processes.

8.1 Restarting the model

If `restart` is set to `.TRUE.` in `namelists/main.nml`, the model will read initial conditions from a restart file. `startTimeStr` must match the time string in the restart file name. For example, if a user specifies `'2008-09-08_18:00:00'` as `startTimeStr`, `umwmrst_2008-09-08_18:00:00.nc` file must be present in the `restart/` directory.

8.2 Excluding enclosed seas

It is possible to mask out lakes or enclosed seas in the domain if they are not desired at model run-time. If the switch `fillLakes` in the `GRID` namelist is set to `.TRUE.`, UMWM will read the input file `namelists/exclude.nml` at run-time. This file should contain a list of grid coordinates (given as pair of integers each) that belong in enclosed seas that are to be filled. It is sufficient to specify one point coordinates per enclosed sea. One line must contain only one pair of grid indices. For example, `namelists/exclude.nml` containing lines:

```
  2     2
105    226
```

will instruct the model to fill lakes that contain grid cells at `(2,2)` and `(105,226)`.

9 Reading model output data

As mentioned in the previous section, the output files will appear in the `output/` directory. The user can open and read output files with any program in a language that has NetCDF libraries provided - Fortran, MATLAB, Python, C/C++, Java etc. In addition, convenience NetCDF viewing programs exist, e.g. Ncview - `http://meteora.ucsd.edu/~pierce/ncview_home_page.html`.

9.1 Grid definition file

A grid definition file is generated at the beginning of every simulation, and is written in `output/umwmout.grid`. It contains grid information that may be used for quantitative analysis of wave model results. This file may also be used as an input grid file to `umwm_topogen` (see section 7.2). The fields that are written in `umwmout.grid` are:

`lon`: 2-dimensional longitude field [$degE$].

`lat`: 2-dimensional latitude field [$degN$].

`dlon`: Grid cell increment in longitude [deg].

`dlat`: Grid cell increment in latitude [deg].

`dx`: Grid cell increment in x [m].

`dy`: Grid cell increment in y [m].

`area`: Grid cell area [m^2].

`depth`: Bathymetry [m] after processing by the wave model. This field is positive below the mean sea level, and has the values effective after the depth limiter has been applied (see section 6.3.2).

`seamask`: Integer field that masks sea points with a value of 1, and land points with a value of 0.

`nproc`: Integer field that marks tiles of the domain covered by each processor in MPI mode.

9.2 Gridded output

The integrated spectrum quantities that describe the wave field, as well as forcing fields, are being output in the file of the form `umwmout_YYYY-MM-DD_hh:mm:ss.nc`, where `YYYY-MM-DD_hh:mm:ss` represent the exact simulation time of the output.

The output fields are listed below:

`frequency`: Frequency range [Hz].

`theta`: Angles of directional bins [rad], defined using mathematical convention.

`lon`: 2-dimensional longitude field [deg].

`lat`: 2-dimensional latitude field [deg].

`seamask`: Integer field with values of 1 over ocean grid cells, and 0 over land.

`depth`: Water depth after processing at initialization time (downward is positive).

`wspd`: Wind speed $[\frac{m}{s}]$ forcing.

`wdir`: Wind direction $[rad]$, defined using mathematical convention.

`uc`: x-component of ocean surface currents $[\frac{m}{s}]$.

`vc`: y-component of ocean surface currents $[\frac{m}{s}]$.

`rhoa`: Air density $[\frac{kg}{m^3}]$.

`rhow`: Water density $[\frac{kg}{m^3}]$.

`taux_form`: x-component of form drag from wind $[\frac{N}{m^2}]$. See section 3.1 for more details on this and the following 3 fields.

`tauy_form`: y-component of form drag from wind $[\frac{N}{m^2}]$.

`taux_skin`: x-component of skin drag from wind $[\frac{N}{m^2}]$.

`tauy_skin`: y-component of skin drag from wind $[\frac{N}{m^2}]$.

`taux_ocn`: x-component of momentum flux from wave dissipation into ocean top $[\frac{N}{m^2}]$.

`tauy_ocn`: y-component of momentum flux from wave dissipation into ocean top $[\frac{N}{m^2}]$.

`taux_bot`: x-component of momentum flux from wave dissipation into ocean bottom $[\frac{N}{m^2}]$.

`tauy_bot`: y-component of momentum flux from wave dissipation into ocean bottom $[\frac{N}{m^2}]$.

`cd`: Atmospheric drag coefficient $[nondimensional]$. Calculated as described in 3.1.

`swh`: Significant wave height $[m]$. Equals the average of the highest third of all waves. Calculated in the model as the integral over the whole spectrum:

$$H_s = 4\sqrt{\int\int E(k,\phi)kdkd\phi} \tag{9.1}$$

`mwp`: Mean wave period $[s]$. Calculated in the model as:

$$\overline{T} = \sqrt{\frac{\int\int E(k,\phi)kdkd\phi}{\int\int f^2 E(k,\phi)kdkd\phi}} \tag{9.2}$$

`mwl`: Mean wave length $[m]$.

$$\overline{L} = 2\pi\sqrt{\frac{\int\int E(k,\phi)kdkd\phi}{\int\int k^2 E(k,\phi)kdkd\phi}} \qquad (9.3)$$

`mwd`: Mean wave direction $[rad]$, mathematical convention.

`dwp`: Dominant wave period $[s]$, corresponding to the discrete peak of the spectrum.

`dwl`: Dominant wave length $[m]$, corresponding to the discrete peak of the spectrum.

`dwd`: Dominant wave direction $[rad]$, corresponding to the discrete peak of the spectrum.

`u_stokes`: Three-dimensional field of x-component of Stokes drift, Lagrangian mean of wave induced fluid motion $[\frac{m}{s}]$. The extent of the third dimension (z, depth levels) is specified by the user (see 6.3.7).

`v_stokes`: Three-dimensional field of y-component of Stokes drift, Lagrangian mean of wave induced fluid motion $[\frac{m}{s}]$. The extent of the third dimension (z, depth levels) is specified by the user (see 6.3.7).

9.3   Spectrum output

Spectrum output at any geographical point in the domain may be optionally enabled through the switch in the `OUTPUT` namelist. For instructions on how to enable spectrum point output, see section 6.3.6).

The list of fields currently being output in a UMWM spectrum file follows.

`Frequency`: One-dimensional frequency array $[Hz]$.

`Wavenumber`: One-dimensional wavenumber array $[\frac{rad}{m}]$.

`Direction`: One-dimensional array of directional bins, mathematical convention $[rad]$.

`Longitude`: Longitude $[degE]$ of the point output. Available only if `topoFromFile` is set to `.TRUE.`.

`Latitude`: Latitude $[degN]$ of the point output. Available only if `topoFromFile` is set to `.TRUE.`.

`wspd`: One-dimensional array of wind speed $[\frac{m}{s}]$ as function of time at the location of point output.

`wdir`: One-dimensional array of wind direction $[rad]$, mathematical convention, as function of time at the location of point output.

`F`: Surface elevation variance spectrum in wavenumber-directional space $[\frac{m^4}{rad^3}]$.

`Sin`: Wind input source in wavenumber-directional space $[\frac{m^4}{rad^3 s}]$.

`Sds`: Wave dissipation sink in wavenumber-directional space $[\frac{m^4}{rad^3 s}]$.

Sdt: One-dimensional wave dissipation due to turbulence in the wave boundary layer in wavenumber-directional space $[\frac{m^4}{rad^3 s}]$.

Sdv: One-dimensional wave dissipation due to viscosity in wavenumber-directional space $[\frac{m^4}{rad^3 s}]$.

Sbf: One-dimensional wave dissipation bottom friction and bottom percolation (total) in wavenumber-directional space $[\frac{m^4}{rad^3 s}]$.

Snl: Non-linear wave-wave interaction source/sink function in wavenumber-directional space $[\frac{m^4}{rad^3 s}]$.

9.4   Restart output

If enabled using the `outrst` switch in the `OUTPUT` namelist(see section 6.3.6, UMWM restart files will be written at desired intervals in the `restart/` directory. These files contain full domain snapshots of the wave spectrum, and can be used to restart the simulation at a later time (see section 8.1). The spectrum data is written in an unstructured, 1-dimensional array, however, 1-dimensional latitude and longitude arrays are stored as well, so these files can also be used for analysis/post-processing when spectrum data from the whole domain is desired.

10 Revision history

Below is the summary of changes made to the UMWM source code since origination. It documents the UMWM version number, date of release, brief summary of changes made, and the author of the changes. The revision history can be also found in the header of the main UMWM program source file, `src/umwm.F90`.

Version	Date	Change log	Contributed by
2.0.0	2016-09-26	- Variable sheltering coefficient A1 - Updated default values of physics coefficients (see `namelists/main.nml`) - Changed plunging breaking factor to coth(0.2*kd) - Sds form now exactly matches eq. (16) in Donelan et al. 2012. - Added a ramp to explim for more stable integration from calm state - Skin drag dependence on Stokes drift - Additional wave diagnostics in the output - Added time dimension to NetCDF gridded output - Minor bug fixes	UM/RSMAS
1.0.1	2012-09-05	- Added Stokes drift calculation and output - Mean-square slope algorithm improved in `UMWM_physics.F90` - Grid spacing now computed from lat/lon input fields - Fixed filling of isolated seas/lakes - Source code clean-up - Minor bug fixes	UM/RSMAS
1.0.0	2012-04-01	- First public release	UM/RSMAS

11 References

Donelan, M. A., and W. J. Plant, 2009: A threshold for wind-wave growth. *J. Geophys. Res.*, 114, C07012, doi:10.1029/2008JC005238.

Donelan, M. A., M. Curcic, S.S. Chen, and A.K. Magnusson, 2012: Modeling waves and wind stress. *J. Geophys. Res.*,117, C00J23, doi:10.1029/2011JC007787.

Jeffreys, H., 1924: On the formation of waves by wind. *Proc. Roy. Soc. A*, 107, pp. 189-206.

Jeffreys, H., 1925: On the formation of waves by wind, II. *Proc. Roy. Soc. A*, 110, pp. 341-347.

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.

Pierson, W. L. 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, pp. 5181-5190.

Shemdin, O.H., K. Hasselmann, S.V. Hsiao, and K. Herterich, 1978: Non-linear and linear bottom interaction effects in shallow water, *Turbulent fluxes through the sea surface, wave dynamics and prediction,* A. Favre and K. Hasselmann (eds), Plenum, New York, pp. 347-372.