User Manual and Documentation of WIM

December 16, 2015

Contents

Chapter 1

Introduction

WIM (Wave in-ice Model) is a simple conceptual 1D spectral model designed to assess some processes evolving in the interactions between waves and sea ice.

Chapter 2

Physical aspects

The model capture two main features of the interactions between waves and sea ice: The attenuation of wave energy by ice and the fracturation of the ice by waves. This section review the physical processes implemented in WIM.

2.1 The basics: The wave spectrum

A record of the sea surface elevation $\eta(t)$ can be considered to be the sum of a large number of harmonic waves with random amplitude and phase.

$$\eta(t) = \sum_{i=1}^{N} \underline{a_i} cos(2\pi f_i t + \underline{\alpha_i}) \tag{2.1}$$

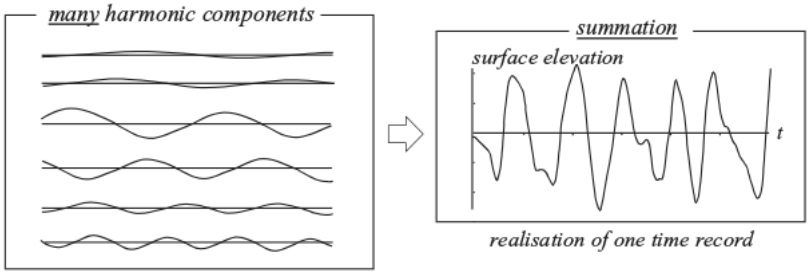

Figure 2.1: A wave record is a sum of many harmonic waves with different amplitudes and phases

The phases and amplitudes being random variables, they can be characterised with their probability density function. For each frequency, the phase $\underline{\alpha_i}$ is uniformly distributed between 0 and 2π and the amplitude $\underline{a_i}$ is rayleigh distributed.

$$P(a_i) = \frac{\pi}{2} \frac{a_i}{\mu_i^2} \exp\left(-\frac{\pi a_i^2}{4\mu_i^2}\right) \tag{2.2}$$

where μ_i is the expected value of the amplitude $\mu_i = E\{\underline{a}_i\}$. The function that shows this expected value over each frequency is called the amplitude spectrum.

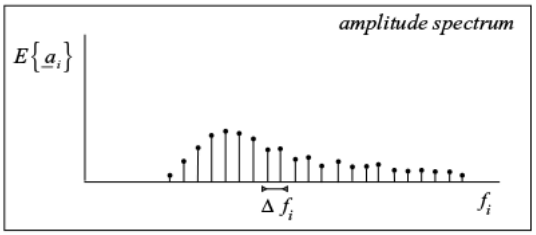

Figure 2.2: amplitude spectrum

But it is more relevant to consider the variance density spectrum

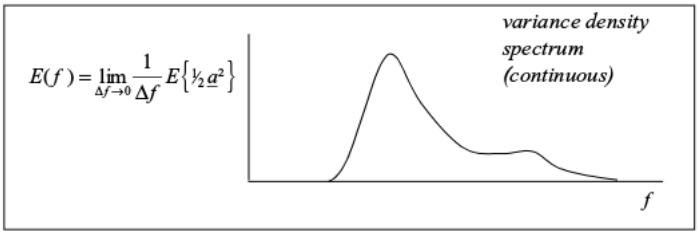

Figure 2.3: variance density spectrum

By multiplying the variance density spectrum by ρg, we obtain the *energy spectrum*

$$S_{energy} = \rho g S_{variance} \tag{2.3}$$

In WIM, the energy spectrum is described using an analytical function. There are many differents equation used to model the wave spectrum, in WIM it is possible to choose between a JONSWAP or a Bretschneider type spectrum. (see section 4.2.3)

2.2 Advection

The wave spectrum is advected by solving

$$\frac{\partial S}{\partial t} + c_g \frac{\partial S}{\partial x} = 0 \tag{2.4}$$

2.3 Attenuation

The attenuation source term is defined as:

$$R_{ice} = \widehat{\alpha} S \tag{2.5}$$

where $\widehat{\alpha}$ is a dimensional attenuation coefficient defined such as:

$$\widehat{\alpha} = \frac{\alpha c}{\langle D \rangle} \tag{2.6}$$

where α is the dimensionless attenuation coefficient, i.e. the (average) amount of attenuation per individual floe, which is a function of ice thickness and wave period.

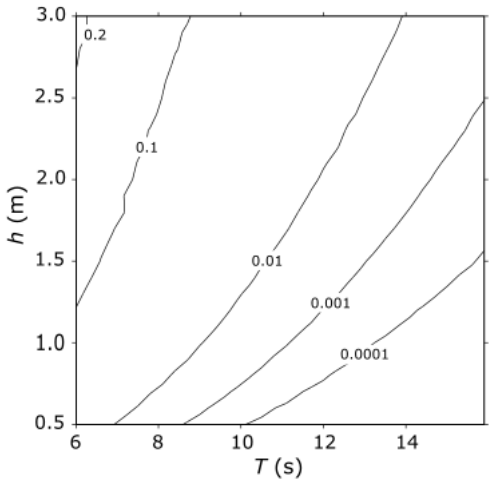

Figure 2.4: Dimensionless energy attenuation coefficient α given bythe model of kohout et meylan (2008)

2.4 floe breaking

The model of Williams et al.(2013) is used to parametrize the wave induced floe breaking.
The flexural strain imposed by passing waves on ice is defined as:

$$\varepsilon = \frac{h}{2}\frac{\partial^2 \eta}{\partial x^2} \tag{2.7}$$

where η is the sea surface elevation,h the ice thickness and x the horizontal distance. We can also define the significant strain

$$E_s = 2\sqrt{\langle \varepsilon^2 \rangle} \tag{2.8}$$

The mean square value is $\langle \varepsilon^2 \rangle = m_0 [\varepsilon]$ where $m_0 [\varepsilon]$ is the 0th moment of the strain spectrum. For a sinusoidal wave of the form $A_{ice} sin (k_{ice} x - \omega t)$ the maximum strain is

$$E = \frac{h}{2} k_{ice}^2 A_{ice}(\omega) \tag{2.9}$$

where k_{ice} is the wavelength in ice and A_{ice} is the amplitude in the ice. We can then define the 0th moment of the strain spectrum

$$m_0 [\varepsilon] = \int S(\omega) E^2(\omega) d\omega \tag{2.10}$$

If the significant strain E_s exceed a certain threshold $E_s > \varepsilon_c$ then the ice breaks and the maximumfloesiwe is set to

$$D_{max} = max\left[\lambda_w/2, D_{min}\right] \tag{2.11}$$

where D_{min} is the minimum floe size and λ_w the dominant wavelength defined as $\lambda_w = 2\pi/k_w$ and $k_w = k_{ice}(2\pi/T_w)$. T_w is the dominant period of the spectrum defined as:

$$T_w = 2\pi\sqrt{\frac{m_0\left[\eta\right]}{m_2\left[\eta\right]}} \tag{2.12}$$

where $m_n\left[\eta\right]$ is the nth moment of the wave spectrum defined as:

$$m_n\left[\eta\right] = \int \omega^n S(\omega) d\omega \tag{2.13}$$

Chapter 3

Getting started

3.1 Get WIM source code

The WIM source code is available on the plateform https://gitlasso.uqar.ca/. To download WIM from this plateform, please execute the following steps:

1. create the destination folder where the source code will be download by typing `mkdir <directory name>`
2. go to the destination folder `cd <directory name>` and execute `git clone https://gitlasso.uqar.ca/bauj0001/WIM2.git .`

3.1.1 Contents

There are 3 subdirectories:

1. `src`: This directory contains all the source files of the routines and subroutines of WIM and the makefile to link and compile all source files
2. `nml`: Contains the namelist file which allow to change model parameters without recompiling the code
3. `output`: Contains the outputs in NETCDF format.

3.2 Compile and execute WIM

Compilation in WIM is relatively easy. The following steps should however be executed correctly to compile the code without errors.

1. Open the **makefile** in `WIM/src/`
2. Choose the compiler that will be used to produce the executable by changing the variable COMPILER. By default, the compiler used is GFORTRAN but it should work with other compilers like INTEL FORTRAN
3. WIM uses the NETCDF library to save outputs. Ensure first that the fortran NETCDF library is currently installed on your machine. If it is not the case, you can download it at http://www.unidata.ucar.edu/

downloads/netcdf/index.jsp. Once the NETCDF library is installed on your machine, you should link it to WIM by indicating the path of the includes files, modules and libraries in the variables NETCDFINC, NETCDFMOD, NETCDFLIB.

4. You can now compile the code by typing `make` in `src/`. The executable will be created in the parent directory `WIM/`.You can also type `make clean` or `make mrproper` to remove the executable, the objects .o and modules .mod if needed.

5. Now execute WIM by typing `./<executable name>`

3.3 Beginning with WIM: using namelists

Designing a simulation with WIM is quite easy. The WIM model uses *namelists* allowing to change some parameter values without recompiling the source code. afterchanging parameters in the namelists, the values will be read by the routine **parameters.f90** (see *routines details* for more informations). All namelists are written in a single file `nml/parameter.nml`. The file is composed by 4 namelists:

- **model parameter**: This namelist allow to change some global parameters of the simulations such as the siwe of the domain, the name of the output directory, the name of the simulation etc..
- **waves parameters**: Specifies the waves parameters such as the significant height, the peak period and some features like allowing dispersion or not.
- **spectrum parameters**: Specifies the parameters to construct the wave spectrum such as the number of frequency bin, the minimal and maximal frequency.
- **ice parameters**: Specifies some ice features like the ice concentration, thickness, initial mean diameters of the floes etc..

After having changed the parameters in the namelists, you can now run WIM.

Chapter 4

Implementation

4.1 General

4.1.1 General algorithm

```
read namelists
allocate memory
initialize

do  time loop
 advection

 do space loop
  attenuation
  floe breaking
  compute new FSD
 end space loop

end time loop

write outputs
deallocate memory
```

4.1.2 Solving ODEs

Time ordinary differential equations are solved in WIM using the first-order Euler method. For a ODE of the general form

$$\frac{dX}{dt} = F(X, t) \tag{4.1}$$

The euler method give:

$$X_{n+1} = X_n + F(X_n, t_n)\Delta t \tag{4.2}$$

4.2 Routine details

4.2.1 main.f90

This is the main program of WIM. It calls all the other subroutines and does the time Loop. It also contains the subroutine `progress` which displays a progress bar in the consol while the model is running.

4.2.2 parameters.f90

The module `parameters.f90` contains all variable and parameters definitions used in the model. All others routines call this module with the Fortran function `USE PARAMETERS`. This module also contains some other subroutines:

1. **read namelists**: Read parameters value specified in the differents namelists contained in the file `parameters.nml` (see Beginning with WIM: using namelists)
2. **array allocation**: Allocate memory for the different arrays and matrix.

4.2.3 initialization.f90

In this routine, the initial spectrum and ice transect are built and initial values for arrays are set.

initial spectrum

It is possible de choose between different methods to build the initial spectrum by setting the parameter `init spec` in the namelist `spectrum parameters`. If `init spec`=1 then a JONSWAP spectrum is built according to

$$S_{init} = \alpha H_s^2 \frac{f_p^4}{f^5} \exp\left[\frac{-5}{4}\left(\frac{f_p}{f}\right)^4\right] \gamma^{\exp\left[\frac{-(f-fp)^2}{2\sigma^2 f_p^2}\right]} \tag{4.3}$$

where

$$\alpha = \frac{0.0624}{0.23 + 0.0336\gamma - \frac{0.185}{1.9+\gamma}}$$

and

$$\sigma = \begin{cases} 0.07, & \text{if } f \leq f_p \\ 0.09, & \text{if } f > f_p \end{cases}$$

(see http://hmf.enseeiht.fr/travaux/CD0910/bei/beiere/groupe4/node/59).
If `init spec`=2 then a Bretschneider spectrum is built according to

$$S_{init} = \frac{1.25 H_s^2 T^5}{8\pi T_p^2} e^{-1.25(T/T_p)^4} \tag{4.4}$$

where T_p is the peak period and H_s the significant height

Ice transect

Parameters for ice transect can be set in the namelist `ice parameters`. Ice concentration is uniform through the domain

$$C(x) = \begin{cases} 0, & \text{if } x < X_1 \\ \texttt{cice}, & \text{if } X_1 < x < X_2 \end{cases} \tag{4.5}$$

If `ice thick`=0 then the ice thickness is constant along the transect. if `ice thick`=1 the ice thickness depends on the distance from the ice edge according to

$$h(x) = \begin{cases} 0, & \text{if } x < X_1 \\ h_\infty \left(0.1 + 0.9\left(1 - e^{-(x - X_1/X_h)}\right)\right), & \text{if } X_1 < x < X_2 \end{cases} \tag{4.6}$$

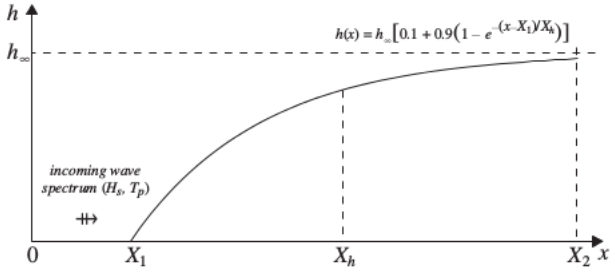

Figure 4.1: Schematical visualization of the ice transect

4.2.4 advection.f90

Here the wave spectrum is advected through the domain by solving

$$\frac{\partial S}{\partial t} + c_g \frac{\partial S}{\partial x} = 0 \tag{4.7}$$

. This equation is discretized using the second order Lax-Wendroff sheme with a superbee flux limiter.

Lax-Wendroff without Flux limiter

The Lax-Wendroff sheme is a *multistep* method where the quantity S_i^{t+1} is calculated using the quantities $S_{i-\frac{1}{2}}^{t+\frac{1}{2}}$ and $S_{i+\frac{1}{2}}^{t+\frac{1}{2}}$.
Using the central difference sheme, the quantity S_i^{t+1} can be approximated by

$$S_i^{t+1} = S_i^t - \frac{c_g \Delta t}{\Delta x}\left(S_{i+\frac{1}{2}}^{t+\frac{1}{2}} - S_{i-\frac{1}{2}}^{t+\frac{1}{2}}\right) \tag{4.8}$$

The both quantities $S_{i\pm\frac{1}{2}}^{t+\frac{1}{2}}$ are obtained by the Lax method:

$$S_{i-\frac{1}{2}}^{t+\frac{1}{2}} = \frac{1}{2}\left(S_i^t + S_{i-1}^t\right) - \frac{c_g \Delta t}{2\Delta x}\left(S_i^t - S_{i-1}^t\right) \tag{4.9}$$

$$S_{i+\frac{1}{2}}^{t+\frac{1}{2}} = \frac{1}{2}\left(S_i^t + S_{i+1}^t\right) - \frac{c_g \Delta t}{2\Delta x}\left(S_{i+1}^t - S_i^t\right) \tag{4.10}$$

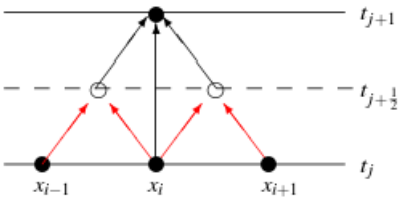

Figure 4.2: Schematical visualization of the Lax-Wendroff method

The Lax-Wendroff sheme is then:

$$S_i^{t+1} = S_i^t - \frac{c_g \Delta t}{2\Delta x}\left(S_{i+1}^t - S_{i-1}^t\right) + \frac{c_g^2 \Delta t^2}{2\Delta x^2}\left(S_{i+1}^t - 2S_i^t + S_{i-1}^t\right) \tag{4.11}$$

The Superbee Flux limiter

It can be seen above that the Lax-Wendroff sheme is a combination of a 1-order term and a 2-order term. To avoid oscillation near steps due to this second term, we add a flux limiter which will "weight" the two terms.

$$\phi(\theta) = \max\left[0, \min\left(2\theta, 1\right), \min\left(\theta, 2\right)\right] \tag{4.12}$$

where θ is given by

$$\theta = \frac{S_{i+1}^t - S_i^t}{S_i^t - S_{i-1}^t} \tag{4.13}$$

The Lax-Wendroff with superbee flux limiter is then

$$S_i^{t+1} = S_i^t - \phi(\theta)\frac{c_g \Delta t}{2\Delta x}\left(S_{i+1}^t - S_{i-1}^t\right) + \phi(\theta)\frac{c_g^2 \Delta t^2}{2\Delta x^2}\left(S_{i+1}^t - 2S_i^t + S_{i-1}^t\right) \tag{4.14}$$

Boundary conditions

Boundary conditions are set according to $\Delta S = 0$

Wave dispersion

In WIM it is possible to allow or not wave dispersion by changing the parameter `disp` in the namelist `waves parameters`. If `disp` take the value 1, dispersion is allowed and c_g depend on the wave period. If `disp` take the value 0, the spectrum is advected at all frequencies at the same speed $c_g = max[c_g(\omega)]$

4.2.5 attenuation.f90

In this routine, the dimensionless attenuation coefficient α from Kohout and Meylan (2008) is approximated and the attenuation source term is computed

4.2.6 fsd build.f90

In this routine the average floe size $\langle D \rangle$ for wave attenuation calculation. It is possible to choose between two methods by setting the parameter `FSD sheme` in the namelist `model parameters`. If `FSD sheme`= 0 then the method proposed by Dumont et al (2011) is used.

$$\langle D \rangle = \frac{\sum_{m=0}^{M} (\xi^2 f)^m \xi^{-m} D_{max}}{\sum_{m=0}^{M} (\xi^2 f)^m} \tag{4.15}$$

where f is the breaking probability (ie. the fragility of the floes), D_{min} the minimum floe size, ξ a parameter which determine the number of pieces each floe will be fragmented into and M the number of fragmentation steps defined as:

$$M = \log_{\xi}(D_{max}/D_{min}) \tag{4.16}$$

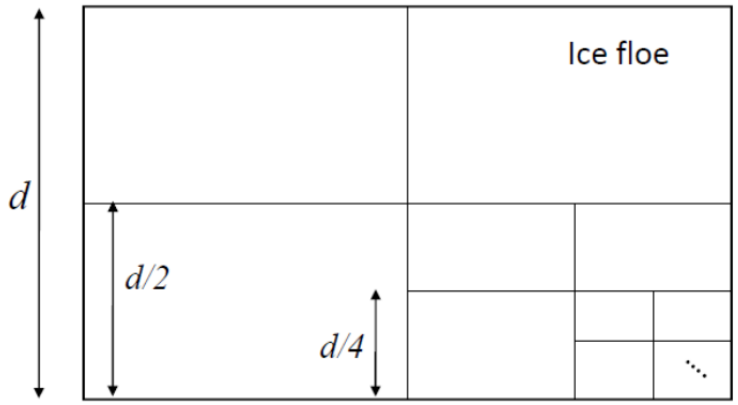

Figure 4.3: Schematical visualization of the ice transect

if `FSD sheme`= 1, a power law is used to compute $\langle D \rangle$

$$\langle D \rangle = A^{-1} \int_{D_{min}}^{D_{max}} D^{-\gamma+1} \tag{4.17}$$

where γ is the exponent of the power law and A a coefficient defined as

$$A = \frac{1}{1-\gamma} \left(D_{max}^{1-\gamma} - D_{min}^{1-\gamma} \right) \tag{4.18}$$

4.2.7 write output.f90

In this routine, the output file is created in NETCDF format. Here is an example of an output file:

```
Format:
        classic
Dimensions:
        omega     = 200
        x_axis    = 50
        time      = 50
        floe size = 100
Variables:
    omega
        Size:       200x1
        Dimensions: omega
        Datatype:   double
    x_axis
        Size:       50x1
        Dimensions: x_axis
        Datatype:   double
    time
        Size:       50x1
        Dimensions: time
        Datatype:   double
    floe size
        Size:       100x1
        Dimensions: floe size
        Datatype:   double
    Spectrum
        Size:       50x50x200
        Dimensions: time,x_axis,omega
        Datatype:   double
    att src term
        Size:       50x50x200
        Dimensions: time,x_axis,omega
        Datatype:   double
    Dave
        Size:       50x1
        Dimensions: x_axis
        Datatype:   double
    Dmax
        Size:       50x1
        Dimensions: x_axis
        Datatype:   double
    Ice concentration
        Size:       50x1
        Dimensions: x_axis
        Datatype:   double
    Ice thickness
        Size:       50x1
        Dimensions: x_axis
        Datatype:   double
    Floe size distribution
        Size:       50x50x100
        Dimensions: time,x_axis,floe size
        Datatype:   double
```

Figure 4.4: Example of an output file

Chapter 5

Graphical User Interface

This section is not yet available