Report type: Memorandum Report

Title: Implementation of Curvilinear Coordinate System in the WAVEWATCH-III model

Authors: W. Erick Rogers and Timothy J. Campbell

Performing organization name and address:
Naval Research Laboratory
Oceanography Division
Code 7322
Stennis Space Center, MS 39529

Subject terms:
WAVEWATCH-III, wind waves, swell, numerical modeling, curvilinear

Name of responsible person: Erick Rogers
Telephone: 228 688 4727
E-mail: rogers@nrlssc.navy.mil

Title: Implementation of Curvilinear Coordinate System in the WAVEWATCH-III model

Abstract

This document describes modifications to the WAVEWATCH-III^{TM} wave model (Tolman 2002a) by the Naval Research Laboratory (NRL) Code 7322. This work is primarily concerned with the implementation of arbitrary structured grid (i.e. curvilinear) approach. Verification test cases are presented.

1. Introduction

High horizontal resolution is required for many wave model applications, especially in cases where horizontal gradients in bathymetry or surface currents are expected to strongly affect the wave field. However, it is equally important that high resolution is not applied where it is not needed, since computer resources are finite. If the modeler is interested in a specific, limited region, then nesting is a suitable solution. However, if the objective is to comprehensively model a large coastline, more elegant methods are required to optimize use of computer resources.

The pre-existing versions of WAVEWATCH-III only allow grids with uniform spacing and, in the case of grids prescribed in latitude/longitude coordinates, require that the grid axes strictly follow the four cardinal directions. The curvilinear technique, already in use by a number of ocean models and wave models, allows more efficient use of computational resources, applying higher spatial resolution where it is required, and aligning the grid with the coastline. The National Centers for Environmental Prediction (NCEP) has stated the goal of operational modeling of the entire US coastline at 5 km resolution by 2009 (Tolman, unpublished document). By making cross-shore resolution relatively high near the shoreline, curvilinear grids would allow valid predictions in water depths as shallow as 30 m for many regions.

For large scale modeling, additional grid projections are possible using a curvilinear model, e.g. Lambert conformal conic.

Another potential advantage of the curvilinear approach is potential reduction of interpolation requirements in communication between models. For example, the wave model grid may match that of the atmospheric model and/or ocean model that it is coupled with. However, it should be noted that resolution requirements for different models are typically dissimilar, so it is not necessarily advantageous to run models on the same grid.

The problems with computational efficiency associated with running WW3 at higher spatial resolution (e.g. 1 – 1000 m) is not addressed herein, but we expect that it will be addressed in a future effort. Until then, it will generally be more efficient to use a model with stationary and/or unconditionally stable nonstationary capability, such as SWAN (Booij et al. 1999), for such applications.

At time of writing, unstructured grid implementation for WW3 is nearing completion (F. Ardhuin, personal communication). This will obviously allow even greater flexibility in computational grid design.

This work was performed was performed during December 2007 – May 2008 using WW3 version 3.13. At time of writing, the modifications have been merged into (pre-release) version 3.14.

It is assumed that the reader is familiar with phase-averaged wave modeling in general and the WAVEWATCH-III (WW3) model in particular. However, the specifics of the WW3 design are in many cases not obvious, being different from other, comparable models such as WAM (WAMDIG 1988; Komen et al. 1994); therefore, some background information is given in the relevant sections below.

2. Auxiliary software development

2.1. Unified grid approach

In the pre-existing WW3 code, the coordinate system (spherical with units degrees latitude/longitude versus Cartesian with units meters *x*/*y*) is selected at compile time and is implemented via a pre-processing of the code itself, selectively activating/deactivating specific lines of code. The model has been modified such that the coordinate system is selected at run time instead. The choice between rectilinear and curvilinear, which is obviously a new feature, is also made at run time. Changes in support of the new run-time options have been incorporated into the user command script, ww3_grid.inp (see Appendix).

2.2. Grid Search and Re-Grid Utilities

Because the pre-existing WW3 code was restricted to rectilinear grids much of the interpolation and associated searching was performed using simple, typically inline, calculations. Remapping between curvilinear grids requires more general search and remapping algorithms. Generalized searching and remapping for logically rectangular grids has been implemented in a new "Grid Search and Re-Grid Utilities" module (W3GSRUMD). The W3GSRUMD data structures and procedures are restricted to logically rectangular (*i*,*j*) grids and cannot be applied to unstructured grids.

The grid-search-utility (GSU) object can be used for rapid searching of the associated grid to identify a grid cell that encloses a target point and to compute interpolation (currently, only bilinear) weights. The GSU object maintains internal pointers to the associated grid coordinate arrays. The GSU object also internally stores information about the grid to facilitate the search operations. Rapid searching is done using a bucket search algorithm. The search buckets are based on the bounding box for the associated grid and an optional user defined minimum number of grid cells per search bucket, n_{cb}. This parameter is set in the code, but is readily modified. The default value is $n_{cb} = 1$, which provides the most efficient searching. Increasing n_{cb} leads to fewer buckets (thus less memory usage) but slower searching. Since the relevant arrays have only two dimensions, memory usage will rarely, if ever, be a constraint.

The GSU object is an "opaque" object. This means that the internals of the object are not accessible outside the W3GSRUMD module. The burden is upon the model developer to invoke the destroy method when finished with a GSU object. If created GSU objects are not properly destroyed, then memory leaks may be introduced.

2.3. Test case software

The pre-existing WW3 software package includes sample test cases, as well as scripts for pre-processing and compiling the WW3 executables. For the present project, a shell script was created to facilitate rapid, repetitive recompiling and testing of modifications during the development process. The script utilizes a database of test cases that is maintained by the user. Necessary files and compile instructions for different test cases are stored in respective subdirectories. The script accepts as command line input, among other things: information on the compiler; instructions regarding which source code to use; information regarding whether the test case is a multi-grid test case; the number of processors to utilize; and whether to run only a subset (versus all) of the WW3 executables. A typical verification process works as follows:

1) User decides on test case and options (e.g. first order scheme vs. higher order scheme)
2) User runs script for baseline case (which is typically some form of the "official" WW3 code).
3) User runs script using the new code
4) User evaluates results using by either

a) executing a second script that in turn executes a series of "diff" commands on model ascii output, or

b) loading graphics software and visually evaluating the output.

Thus, there are typically only three line commands required (e.g. 2, 3, 4a above) for each test case.

The associated test case database created for this project includes many of the pre-existing test cases, as well as several new ones associated with curvilinear coordinates. These are detailed in Section 4.

3. Implementation of curvilinear grids

3.1. Grid quantities

For simplicity, we present here only the case for Cartesian (meters) coordinate system. Our primary references for this work are two unpublished documents by Henri Petit (formerly of Delft Hydraulics) and G. van Vledder (TU Delft), and two documents related to the SWAN model, the technical documentation for version 40.51A (SWAN 2007), and a conference paper, Booij et al. (1997). The first two documents are descriptions of numerical aspects of "PHIDIAS" model, and are referred to herein as the "PHIDIAS documentation".

In a rectilinear system, increasing the spatial index *i* directly translates to "increasing the *x* position", and similar for *j* and the *y* position; thus $n_x = n_i$ and $n_y = n_j$. The fundamental difference for a curvilinear coordinate system is that the globally defined measures of position, *x* and *y*, do not have a consistently defined correspondence

with the grid indices *i* and *j*. The user provides the grid as $x_{i,j}$ and $y_{i,j}$. The concept of n_x, n_y is no longer valid; rather, the spatial grids are sized by n_i, n_j only.

For finite difference operations, such as used by the geographic propagation scheme, the model requires information about the variation of the index space with regard to the positional (*x*,*y*) space. Since *i* is a counter, not a "space", index space is presented in the equations below as (*p*,*q*); however, there is a direct correspondence between *p* and *i*, with $\Delta p == 1$; and similar for *q* and *j*. The derivatives of (*p*,*q*) with respect to (*x*,*y*) cannot be calculated directly since (*x*,*y*) are defined on (*p*,*q*), not vice versa. Thus, the first step is to calculate the derivatives of (*x*,*y*) with respect to (*p*,*q*). A centered finite difference approximation is used for the interior points:

$$\frac{\partial x}{\partial p} \approx 0.5(x_{i+1,j} - x_{i-1,j}) \tag{3.1-1}$$

$$\frac{\partial y}{\partial p} \approx 0.5(y_{i+1,j} - y_{i-1,j}) \tag{3.1-2}$$

$$\frac{\partial x}{\partial q} \approx 0.5(x_{i,j+1} - x_{i,j-1}) \tag{3.1-3}$$

$$\frac{\partial y}{\partial q} \approx 0.5(y_{i,j+1} - y_{i,j-1}) \tag{3.1-4}$$

At this point, a Jacobian can be calculated, which can also be thought of as a metric for the local grid curvature:

$$\sqrt{G} = \frac{\partial x}{\partial p}\frac{\partial y}{\partial q} - \frac{\partial x}{\partial q}\frac{\partial y}{\partial p} \tag{3.1-5}$$

This allows the model to calculate the necessary quantities, the derivatives of (*p*,*q*) with respect to (*x*,*y*):

$$\frac{\partial p}{\partial x} = +\frac{1}{\sqrt{G}}\frac{\partial y}{\partial q} \tag{3.1-6}$$

$$\frac{\partial p}{\partial y} = -\frac{1}{\sqrt{G}}\frac{\partial x}{\partial q} \tag{3.1-7}$$

$$\frac{\partial q}{\partial x} = -\frac{1}{\sqrt{G}}\frac{\partial y}{\partial p} \tag{3.1-8}$$

$$\frac{\partial q}{\partial y} = +\frac{1}{\sqrt{G}}\frac{\partial x}{\partial p} \tag{3.1-9}$$

For the special case of a rectilinear grid, these derivatives become:

$$\left(\frac{\partial p}{\partial x}\right)_{rect} = \frac{1}{\Delta x} \;;\; \left(\frac{\partial p}{\partial y}\right)_{rect} = 0 \;;\; \left(\frac{\partial q}{\partial x}\right)_{rect} = 0 \;;\; \text{and} \;\; \left(\frac{\partial q}{\partial y}\right)_{rect} = \frac{1}{\Delta y}$$

Two other quantities are calculated which provide useful approximations for grid cell size, comparable to Δx and Δy of rectilinear grids:

$$H_p = \sqrt{G_{pp}} = \sqrt{\left(\frac{\partial x}{\partial p}\right)^2 + \left(\frac{\partial y}{\partial p}\right)^2} \tag{3.1-10}$$

$$H_q = \sqrt{G_{qq}} = \sqrt{\left(\frac{\partial x}{\partial q}\right)^2 + \left(\frac{\partial y}{\partial q}\right)^2} \tag{3.1-11}$$

3.2. Spatial Gradients

Spatial gradients of the depth and currents must be calculated in order to calculate spectral propagation speeds, e.g. for refraction. In the rectilinear WW3, the depth gradient is simply

$$\frac{\partial h}{\partial x} \approx \frac{h_{i+1,j} - h_{i-1,j}}{2\Delta x} \tag{3.2-1}$$

$$\frac{\partial h}{\partial y} \approx \frac{h_{i,j+1} - h_{i,j-1}}{2\Delta y} \tag{3.2-2}$$

The curvilinear calculation is via chain rule:

$$\frac{\partial h}{\partial x} = \frac{\partial h}{\partial p}\frac{\partial p}{\partial x} + \frac{\partial h}{\partial q}\frac{\partial q}{\partial x} \tag{3.2-3}$$

$$\frac{\partial h}{\partial y} = \frac{\partial h}{\partial p}\frac{\partial p}{\partial y} + \frac{\partial h}{\partial q}\frac{\partial q}{\partial y} \tag{3.2-4}$$

with $\frac{\partial h}{\partial p} \approx 0.5(h_{i+1,j} - h_{i-1,j})$, $\frac{\partial h}{\partial q} \approx 0.5(h_{i,j+1} - h_{i,j-1})$, and $\frac{\partial p}{\partial x}, \frac{\partial q}{\partial x}, \frac{\partial p}{\partial y}, \frac{\partial q}{\partial y}$ having been calculated using the indirect finite differencing as shown in Section 3.1. The calculation of the gradients of the *x* and *y* components of the current velocity is performed in the same way as for the depths shown here.

3.3. Propagation using the "PHIDIAS method"

WW3 uses a flux method for geographic propagation, in conjunction with either the first order upwind explicit scheme, or the higher order "ULTIMATE QUICKEST" scheme, which is detailed in Leonard (1991). To generalize these methods for curvilinear coordinates, we follow the suggestion given in the PHIDIAS documentation, which is to use the Jacobian given above to convert the entire problem between the normal, curving space and a straightened space. Unlike the PHIDIAS suggestion, however, this conversion is performed only inside the propagation routine, rather than integrating the entire model in straightened space, which would require transformation of model source terms, etc. A simple, three step process is used every time the propagation subroutine is called (i.e. for every time step and every spectral component):

1) Convert dependent variable (wave action density) to straightened space: $\widetilde{N} = \sqrt{G}N$
2) Propagate wave action density using flux method via subroutine calls for each (of two) grid axes.
3) Convert wave action density back to normal, curved space: $N = \widetilde{N} / \sqrt{G}$

In the pre-existing WW3, the subroutine that executes the flux method for *x* and *y* assumes that the axes are orthogonal and that the grid spacing is uniform. The primary advantage of the PHIDIAS approach is that this fluxing subroutine *is not modified*. The fluxing is being performed for *p* and *q*, but this is transparent to subroutine itself. In the straightened space, the axes are orthogonal, with $\Delta p = \Delta q == 1$, as noted above.

The propagation velocities need to be provided to the fluxing subroutine in the straightened, index space. These velocities are:

$$C_p = \frac{\partial p}{\partial t} = \frac{\partial x}{\partial t}\frac{\partial p}{\partial x} + \frac{\partial y}{\partial t}\frac{\partial p}{\partial y} = C_x \frac{\partial p}{\partial x} + C_y \frac{\partial p}{\partial y} \tag{3.3-1}$$

$$C_q = \frac{\partial q}{\partial t} = \frac{\partial x}{\partial t}\frac{\partial q}{\partial x} + \frac{\partial y}{\partial t}\frac{\partial q}{\partial y} = C_x \frac{\partial q}{\partial x} + C_y \frac{\partial q}{\partial y} \tag{3.3-2}$$

Though this method differs slightly from the description given in the PHIDIAS documentation, we refer to it herein as the "PHIDIAS method" for propagation on a curvilinear grid.

3.4. Anti-GSE measures

WW3 provides two options for dealing with the Garden Sprinkler Effect (GSE). One involves the addition of a diffusion scheme to the governing equation, following Booij and Holthuijsen (1987). The other method involves directionally appropriate spatial averaging, following Tolman (2002b). Both methods are now implemented in WW3 for curvilinear coordinates. These calculations are performed in curved (normal) space.

3.4.1. Diffusion method

The diffusion occurs separate from all other operations, within the propagation subroutine. The scheme is implemented in curvilinear coordinates in a manner similar to that used in the curvilinear SWAN code (undocumented, but see code of public versions 40.11 and all later versions). The update step is:

$$N_{i,j} = N_{xx} + N_{yy} + N_{xy} \tag{3.4-1}$$

where

$$N_{xx} = D_{xx}\Delta t\left[\left(\frac{\partial p}{\partial x}\right)^2 Q_{xx} + 2\frac{\partial p}{\partial x}\frac{\partial q}{\partial x}Q_{xy} + \left(\frac{\partial q}{\partial x}\right)^2 Q_{yy}\right] \tag{3.4-2}$$

$$N_{yy} = D_{yy}\Delta t\left[\left(\frac{\partial p}{\partial y}\right)^2 Q_{xx} + 2\frac{\partial p}{\partial y}\frac{\partial q}{\partial y}Q_{xy} + \left(\frac{\partial q}{\partial y}\right)^2 Q_{yy}\right] \tag{3.4-3}$$

$$N_{xy} = 2D_{xy}\Delta t\left[\frac{\partial p}{\partial x}\frac{\partial p}{\partial y}Q_{xx} + \frac{\partial q}{\partial x}\frac{\partial q}{\partial y}Q_{yy} + \left(\frac{\partial p}{\partial x}\frac{\partial q}{\partial y} + \frac{\partial q}{\partial x}\frac{\partial p}{\partial y}\right)Q_{xy}\right] \tag{3.4-4}$$

The diffusion coefficients *D*, are taken directly from Booij and Holthuijsen (1987), and are not changed during the generalization of WW3 for curvilinear coordinates:

$$D_{xx} = (D_{ss}\cos^2(\theta) + D_{nn}\sin^2(\theta)) \tag{3.4-5}$$

$$D_{yy} = (D_{ss}\sin^2(\theta) + D_{nn}\cos^2(\theta)) \tag{3.4-6}$$

$$D_{xy} = (D_{ss} - D_{nn})\cos(\theta)\sin(\theta) \tag{3.4-7}$$

Here, the subscript of *D* indicates the direction of the diffusion, in tensor notation, with *s* indicating "in the direction of propagation" and *n* indicating "normal to the propagation direction".

The *Q* terms are simply the components of a particular finite difference scheme for diffusion in two dimensions (equivalent to the scheme given in Fletcher (1991, Volume 2, p. 57, eq. 12.34):

$$Q_{xx} = \frac{\partial^2 N}{\partial p^2} = N_{i+1,j} - 2N_{i,j} + N_{i-1,j} \tag{3.4-8}$$

$$Q_{yy} = \frac{\partial^2 N}{\partial q^2} = N_{i,j+1} - 2N_{i,j} + N_{i,j-1} \tag{3.4-9}$$

$$Q_{xy} = \frac{\partial^2 N}{\partial p \partial q} = 0.25(N_{i+1,j+1} - N_{i-1,j+1} - N_{i+1,j-1} + N_{i-1,j-1}) \tag{3.4-10}$$

These Q terms are also not changed in the curvilinear WW3 code; thus, the same finite difference scheme is used as before.

For the special, less general, case of a rectilinear grid, the curvilinear solution reduces to:

$$N_{i,j}^{rect} = D_{xx}\Delta t\left[\left(\frac{1}{\Delta x}\right)^2 Q_{xx}\right] + D_{yy}\Delta t\left[\left(\frac{1}{\Delta y}\right)^2 Q_{yy}\right] + 2D_{xy}\Delta t\left[\left(\frac{1}{\Delta x}\frac{1}{\Delta y}\right)Q_{xy}\right] \tag{3.4-11}$$

3.4.2. Averaging method

Normal and parallel width in the pre-existing WW3 is:

$$s_x = F_{cg,c}\frac{1}{\Delta x}\cos(\theta) \tag{3.4-12}$$

$$s_y = F_{cg,c}\frac{1}{\Delta y}\sin(\theta) \tag{3.4-13}$$

$$n_x = -F_{\theta,c}\frac{1}{\Delta x}\sin(\theta) \tag{3.4-14}$$

$$n_y = F_{\theta,c}\frac{1}{\Delta y}\cos(\theta) \tag{3.4-15}$$

with

$$F_{cg,c} = 0.5\alpha_g\alpha_s\left(\gamma - \frac{1}{\gamma}\right)\frac{C_g\Delta t}{n_{\Delta t}} \tag{3.4-16}$$

$$F_{\theta,c} = \alpha_g\alpha_n\Delta\theta C_g\Delta t \tag{3.4-17}$$

A global tuning parameters is set in the code, $\alpha_g = 1$. α_n is the averaging area for correcting the GSE associated with directional binning; it is user-specified, 1.5 by default. α_s is the averaging area for correcting the GSE associated with frequency (group velocity) binning, and is also 1.5 by default. γ is the frequency bin increment, $f_{i+1} = \gamma f_i$. The parameter $n_{\Delta t}$ is related to the dynamic time stepping in WW3, relating the global time step to the sub-step: $\Delta t_g = n_{\Delta t}\Delta t$. Division by $n_{\Delta t}$ in the pre-existing WW3 (eq. 3.4-16) may be unintentional, since it apparently does not follow Tolman (2002b), does not appear in the $F_{\theta,c}$ calculation, and is not intuitive, but we retain this feature in the curvilinear version, for consistency during verification, with probable revision later.

In the curvilinear WW3, it is changed to:

$$s_x = F_{cg,c}\frac{\partial p}{\partial x}\cos(\theta) + F_{cg,c}\frac{\partial p}{\partial y}\sin(\theta) \tag{3.4-18}$$

$$s_y = F_{cg,c}\frac{\partial q}{\partial x}\cos(\theta) + F_{cg,c}\frac{\partial q}{\partial y}\sin(\theta) \quad (3.4\text{-}19)$$

$$n_x = -F_{th,c}\frac{\partial p}{\partial x}\sin(\theta) + F_{th,c}\frac{\partial p}{\partial y}\cos(\theta) \quad (3.4\text{-}20)$$

$$n_y = -F_{th,c}\frac{\partial q}{\partial x}\sin(\theta) + F_{th,c}\frac{\partial q}{\partial y}\cos(\theta) \quad (3.4\text{-}21)$$

The strength of the averaging scheme is dependent on grid resolution. Since grid resolution is non-uniform for curvilinear grids, this will mean that the strength of the averaging will also be non-uniform. This is not a desirable effect. A potential future upgrade would be to add an additional term/factor that balances the effect of the spatial variation of grid resolution.

4. Verification

In this section, verification test cases are presented. Herein, the propagation schemes are referenced as:

1) Propagation scheme 1, which is the first order upwind explicit scheme. Here, the *x* and *y* propagation occurs in a single step.
2) Propagation scheme 2, which is the ULTIMATE QUICKEST scheme with the additional diffusion term included to address the Garden Sprinkler Effect problem.
3) Propagation scheme 3, which is the ULTIMATE QUICKEST scheme with a spatial averaging procedure included to address the Garden Sprinkler Effect problem.

With schemes 2 and 3, the *x* and *y* propagation occurs in two separate steps, specifically, two separate calls to the ULTIMATE QUICKEST fluxing subroutine. The GSE corrections are also performed in separate steps. For example, with propagation scheme 2, the sequence is:

1. propagate in *x* (or *y*)
2. propagate in *y* (or *x*)
3. add diffusion

Parentheses here indicate that the order is reversed every other time step.

4.1. One-dimensional propagation

4.1.1. Test case tp1.1

Summary: Full global propagation of Gaussian spike along the equator (one-dimensional), a test case included in public release WW3 v2.22.
Settings:

- Δt_g =3600 s , Δt_{xy} =3600 s
- duration= 24 days, 19680601 000000 to 19680625 000000
- n_x = 360 n_y = 3 number of sea points=360
- Δx = 1.0° Δy = 1.0°

- x range: -180° to 179°
- y range: -1° to 1°
- n_f = 3 f_1 = 0.0368 Hz, γ_f =1.1
- n_θ = 4
- deep water
- input spectrum: H_s=2.5 m, f_p =0.0407 Hz , θ_m =270.0°
- Garden Sprinkler Corrections disabled (note that with this setting, propagation schemes 2 and 3 should produce identical results)

Six (6) cases tested:
- rectilinear, pseudo-curvilinear
- propagation schemes 1,2,3

Baseline: v3.13 with propagation schemes 1,2,3
Result: exact match.

4.1.2. Test case tp1.2

Summary: Partial propagation of Gaussian spike along meridian (one-dimensional), a test case included in public release WW3 v2.22.
Settings:
- Δt_g =3600 s , Δt_{xy} =3600 s
- duration= 6 days, 19680601 000000 to 19680607 000000
- n_x = 3 , n_y = 123 , number of sea points=121
- Δx = 1.0° , Δy = 1.0°
- x range: -1° to 1°
- y range: -61° to 61°
- n_f = 3, f_1 = 0.0368, γ_f =1.1
- n_θ = 4
- deep water
- input spectrum: H_s=2.5 m fp=0.0407 Hz , Dir=180.0 °
- Garden Sprinkler Corrections disabled

Three (3) cases tested:
- rectilinear only
- propagation schemes 1,2,3

Baseline: v3.13 with propagation schemes 1,2,3
Result: exact match.

4.1.3. Test case tp1.3

Summary: Test for monochromatic shoaling (one-dimensional). This is a test case included in public release WW3 v2.22.

Settings:
- n_θ = 4

- n_f = 3 , f_1 = 0.0800 , γ_f =1.25
- Δt_g =1200 s , Δt_{xy} =1200 s
- n_x = 43 , n_y = 3 , number of sea points= 42
- Δx = 15 km , Δy = 15 km
- X range (km) : -15.00 615.00
- Y range (km) : -15.00 15.00
- bathymetry: sloping beach
- input spectrum: H_s=1.0 m f_p =0.1 Hz , θ_m =270.0°
- duration=2 days, 19680606 000000 to 19680608 000000

Three (3) cases tested (rectilinear only):
(1,2)= propagation schemes 1,2 without GSE correction
(3)= propagation scheme 3 with $\alpha_s = \alpha_n$ = 1.50
Baseline: equivalent v3.13 simulations

Result:
Cases (1) and (2) match exactly.
Case (3) shows very minor differences.

4.1.4. Test case tp1.4

Summary: Test of spectral refraction (one-dimensional propagation, *x*-direction). This is a test case included in public release WW3 v2.22.

Settings:

- n_θ = 24
- n_f = 3 , f_1 = 0.0800 , γ_f =1.25
- Δt_g =300 s , Δt_{xy} =300 s
- n_x = 13 , n_y = 3 , number of sea points= 11
- Δx = 5 km Δy = 5 km
- X range (km) : -5.00 55.00
- Y range (km) : -5.00 5.00
- bathymetry: sloping beach
- input spectrum: H_s=1.0 m f_p =0.1 Hz , θ_m =300.0°
- duration=12 hours, 19680606 000000 to 19680606 120000

Three (3) cases tested (rectilinear only):
Cases (1,2,3)= propagation schemes 1,2,3
Baseline: equivalent v3.13 simulations

Result:
Case (1) matches exactly.

Cases (2,3) shows insignificant differences.

4.1.5. Test case tp1.5

Summary: Test of spectral refraction (one-dimensional propagation, y-direction). This is a test case included in public release WW3 v2.22.

Settings:

- n_θ = 24
- n_f = 3 , f_1 = 0.0800 , γ_f =1.25
- Δt_g =300 s , Δt_{xy} =300 s
- n_x = 3 , n_y = 13 , number of sea points= 11
- Δx = 5 km , Δy = 5 km
- X range (km) : -5.00 5.00
- Y range (km) : -5.00 55.00
- bathymetry: sloping beach
- input spectrum: H_s=1.0 m f_p =0.1 Hz , θ_m =330.0°
- duration=12 hours , 19680606 000000 to 19680606 120000

Three (3) cases tested (rectilinear only):
Cases (1,2,3)= propagation schemes 1,2,3
Baseline: equivalent v3.13 simulations

Result:
Case (1) matches exactly.
Cases (2,3) shows insignificant differences.

4.1.6. Test case tp1.6

Summary: Test with wave blocking (one-dimensional propagation). This is a test case included in public release WW3 v2.22.

Settings:

- n_θ = 8
- n_f = 15 , f_1 = 0.1863 , γ_f =1.1
- Δt_g =600 s , Δt_{xy} =600 s
- n_x = 22 , n_y = 3 , number of sea points= 20
- Δx = 3 km Δy = 3 km
- X range (km) : -3.00 60.00
- Y range (km) : -3.00 3.00
- deep water
- input spectrum: H_s=1.0 m f_p =0.3 Hz , θ_m =270.0°
- duration=1 day , 19680606 000000 to 19680607 000000

One (1) case tested:

- rectilinear only
- propagation scheme 1 only

Baseline: equivalent v3.13 simulations
Result: exact match.

4.1.7. Test case tp1.6e

Summary: A new test case, based on tp1.6: 1-d, opposing current that becomes stronger in the down-wave direction. The difference from tp1.6 is that the currents are not so strong that blocking occurs.

Configuration, other than current field, is identical to tp1.6

Three (3) cases tested:

- rectilinear only
- propagation schemes 1,2,3

Baseline: equivalent v3.13 simulations
Result: exact match.

The results for test cases tp1.6 and tp1.6e are shown in Figure 4-1. Results using new code match that from v3.13 exactly, so only one model result is plotted for each of the two test cases.

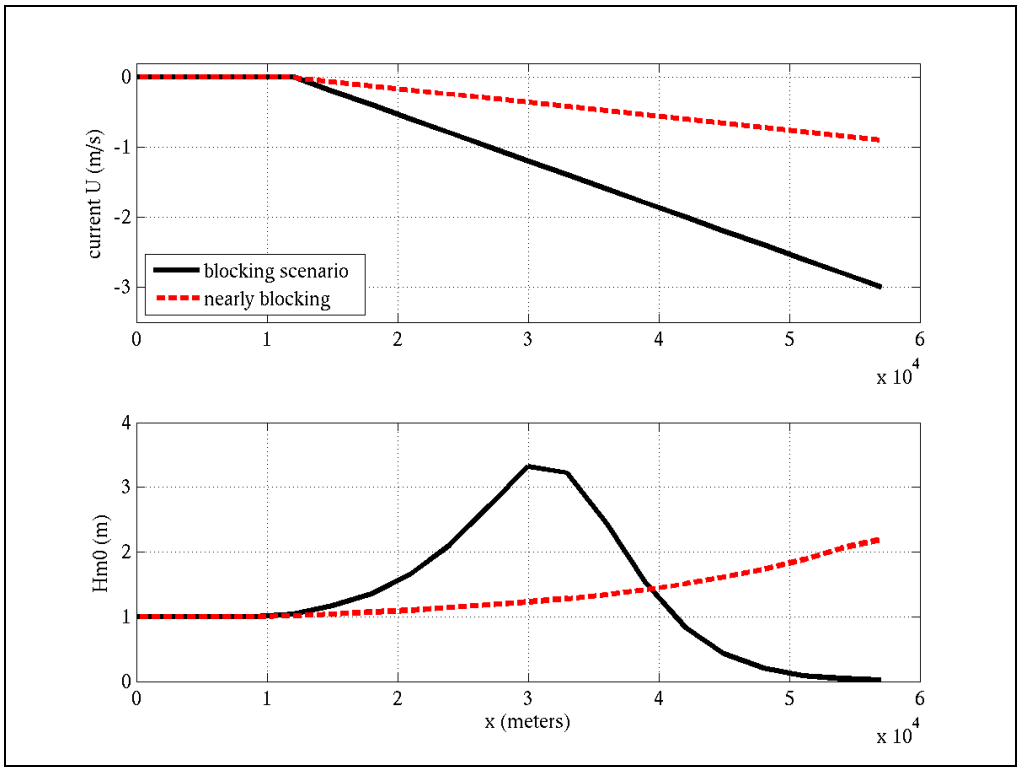

Figure 4-1 Results from tp1.6 (blocking scenario) and tp1.6e (nearly blocking).

4.2. Two-dimensional propagation

4.2.1. Test case tp2.1

Summary: Propagation of Gaussian spike under angle with grid (two-dimensional, meters grid), a test case included in public release WW3 v2.22.

Settings:

- n_θ = 24
- n_f = 3 f_1 = 0.0368 γ_f =1.1
- Δt_g =360 s , Δt_{xy} =360 s
- n_x = 43 , n_y = 43 , number of sea points= 973
- Δx = 10 km , Δy = 10 km
- X range (km) : -60.00 360.00
- Y range (km) : -60.00 360.00
- deep water with some land points in grid
- input spectrum: H_s=2.5 m f_p =0.04 Hz , θ_m =225.0°
- duration=4 hrs 48 min, 19680606 000000 to 19680606 044800

Three (3) cases tested:

- rectilinear only
- propagation schemes 1,2,3 without GSE correction

Baseline: equivalent v3.13 simulations
Result: exact match.

4.2.2. Test case tp2.1b

Summary: Two-dimensional propagation of a single Gaussian spike under angle with a Cartesian (meters-meters) rectilinear grid, with optional extension to multiple spikes/angles.

Settings:

- n_θ = 12
- n_f = 3 , f_1 = 0.0368 , γ_f =1.1
- Δt_g =360 s , Δt_{xy} =360 s
- n_x = 273 , n_y = 274 , number of sea points= 73712
- Δx = 16 km , Δy = 16 km
- X range (km) : 0.00 4352.00
- Y range (km) : 0.00 4368.00
- deep water
- input spectrum: H_s=2.5 m f_p =0.04 Hz , θ_m =210.0°
- duration=1 day , 19680606 000000 to 19680607 000000
- Garden Sprinkler Corrections disabled

Six (6) cases tested

- rectilinear and pseudo-curvilinear
- Cases (1,2,3)= propagation schemes 1,2,3

Baseline: equivalent v3.13 simulations

Result:
Case (1) matches exactly.
Cases (2,3) shows insignificant differences.

Figure 4-2 shows the result for test case tp2.1b with propagation scheme 1 after 24 hours of propagation. Since there is no GSE correction for this scheme, the reduction in amplitude of the signal seen in the figure is entirely due to unintentional numerical diffusion. This first order upwind explicit scheme, employed as the default scheme in most implementations of the WAM model (WAMDI Group 1988), is notorious for having diffusion characteristics that are dependent on the direction of propagation. This is evident here, with the four signals propagating along the grid axes being less strongly diffused. Inconsistency in the nature of the strong shape distortion is also evident.

Figure 4-3 shows the result for test case tp2.1b with propagation scheme 2. In this example, the diffusion is nearly zero, with hardly noticeable amplitude reduction or shape distortion.

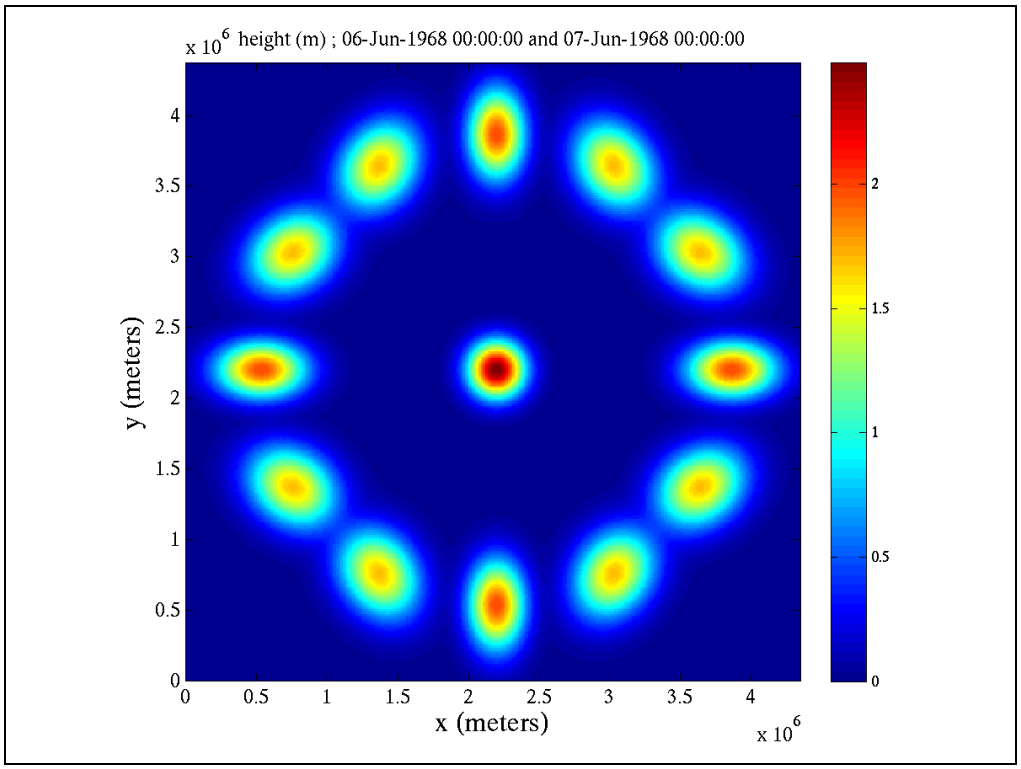

Figure 4-2 Test case 2.1b PR1 (first order scheme) used. H_{m0} (m) plotted.

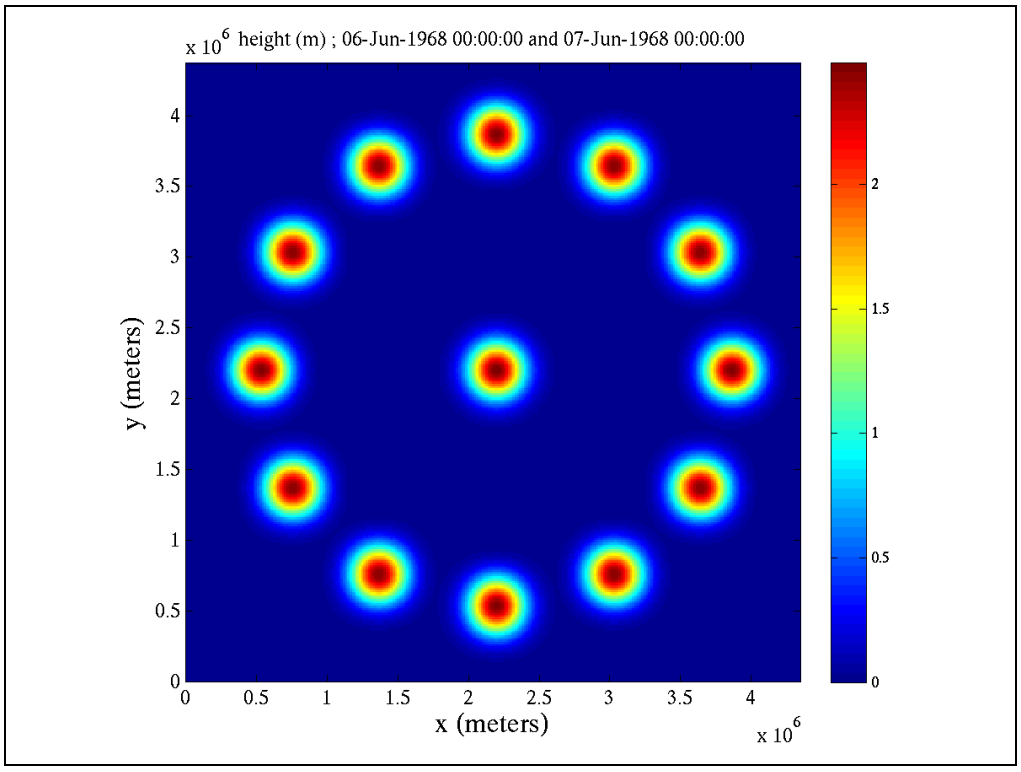

Figure 4-3 Test case 2.1b, PR2 (higher order scheme) used, no GSE controls (i.e. T_s =0) . H_{m0} (m) plotted.

4.2.3. Test case tp2.1c

Summary: Similar to 2.1b {two-dimensional propagation of a single Gaussian spike under angle with a Cartesian (meters-meters) grid, with optional extension to multiple spikes/angles}, but on a true curvilinear grid. Grid is the top left quadrant of a polar grid.

Settings:

- n_θ = 12
- n_f = 3 , f_1 = 0.0368 , γ_f =1.1
- Δt_g =360 s , Δt_{xy} =180 s (smaller for propagation scheme with GSE controls active)
- n_i = 226 , n_j = 331 , number of sea points= 73696
- Δx , Δy variable
- X range (km) : 1040.39 to 7000.00
- Y range (km) : 2000.00 to 7959.61
- deep water
- input spectrum: H_s=2.5 m f_p =0.04 Hz , θ_m =210.0°
- duration=1 day , 19680606 000000 to 19680607 000000

Verification is less straightforward in this case, since it is not possible to apply version 3.13 to this true curvilinear case. Verification methods:

1. Visualization of results for eight (8) scenarios: with propagation scheme 2 (UQ with diffusion), T_s = 0,7200,14400,86400 ; propagation scheme 3 (UQ with averaging), α_s = 0,1.50, α_n = 0,1.50. This verification is qualitative, and is only meant to detect gross errors. Result: no problems detected.
2. Visual comparison of results without GSE correction using the three propagation schemes (schemes 2 and 3 should be identical in this case) versus comparable cases on rectilinear grid, case 2.1b. This verification is qualitative, and is only meant to detect gross errors. Result: no problems detected. Note: a quantitative comparison could be made by regridding to a common grid, but these could not be expected to be identical, since numerical error (diffusion, dispersion) are necessarily different when propagated on different grids. Therefore, the outcome would be not much more useful than that from the qualitative comparison.
3. Mass balance checking. The case is designed such that energy does not enter or leave the domain during the simulation, and source terms are zero. The model domain is integrated to ensure that total energy does not change. Since velocity is uniform in this test case, the integration is effectively a check on energy flux conservation. Results:
   - Propagation scheme 1, volume change over duration of simulation is 0.02%
   - Propagation schemes 2 and 3, with no GSE correction: 0.01%
   - Propagation scheme 2, with T_s = 14400 s : 0.09%
   - Propagation scheme 3, $\alpha_s = \alpha_n$ = 1.50: 0.31%

Figure 4-4 shows the result for test case tp2.1c with propagation scheme 1. Again, the scheme shows strong numerical diffusion. The tendency to favor a particular propagation direction is less evident, since none of the signals are propagating exactly along the grid lines, unlike the rectilinear example (tp2.1b). Inconsistent manner of shape distortion is, however, still apparent.

Figure 4-5 shows the result for test case tp2.1c with propagation scheme 2, without GSE controls (i.e. T_s =0). Here, in contrast to Figure 4-3, there is noticeable numerical diffusion, with a modest reduction in amplitude visible. There is also some distortion in shape, with the low energy contours of most of the 12 signals taking a squared-off shape. However, the diffusion and distortion is still far better than that of the first order scheme on either grid type and the manner of distortion is less dependent on propagation direction. There is less amplitude reduction for the signals propagating toward the lower right. This is due to the higher resolution in this portion of the grid, which reduces diffusion.

Figure 4-6 shows the result for test case tp2.1c with propagation scheme 2, with T_s =4 hours. A time step size Δt_{xy} =90 s is used here. Here, the GSE control is active, but is only 25% of the strength that should ideally be applied. (Since the “final state” being evaluated is after 24 hours of propagation, the optimal diffusion strength T_s is 24 hours.).

Even with the diffusion strength at only 25%, the "squaring off" of the signals is hardly, if at all, noticeable. Again, the unintentional numerical diffusion is slightly less for the signals propagating toward the lower right, due to inhomogeneous spatial resolution.

Figure 4-7 shows the result for test case tp2.1c with propagation scheme 2, with T_s =24 hours. Here, the GSE control is active, and is applied at the optimal strength: T_s is equal to the simulation duration. The result is a continuous circular geographic distribution of energy, a very nice result qualitatively similar to what one would expect with, e.g., $\Delta\theta = 0.001°$, for which GSE would be unnoticeable. It should be noted here that the propagation time step was rather restrictive in this case, since, all else being equal, a larger T_s requires a smaller time step size; Δt_{xy} =10 s is used here, which may not be feasible for a typical operational model.

Figure 4-8 shows the result for test case tp2.1c with propagation scheme 3 with default GSE-control settings, $\alpha_s = \alpha_n = 1.50$. Visually, strength of the GSE control here appears approximately similar to that of propagation scheme 2, with T_s =4 hours. Scheme 3, however, has the advantage of not influencing the choice of time step size, so it will often be considerably faster than scheme 2. One small defect is evident, though: the amplitudes of the signals propagating toward the lower right are larger. We know from Figure 4-6 that only a minority of this can be attributed to the reduction of unintentional numerical diffusion by increased resolution. Most of the difference must then be due to the averaging method, which, as presently designed, does not take into account the spatial variation of the geographic resolution possible with curvilinear grids. Specifically, where the grid cell size is larger, the effect of the averaging is greater. This could potentially be addressed by scaling α_g such that it is increased (decreased) where the local grid resolution, $H_p \times H_q$ is small (large), relative to some global mean, e.g. $\alpha_g = \left(\frac{\left[H_p H_q \right]_{mean}}{\left[H_p H_q \right]_{local}} \right)^{\phi}$, where ϕ is a control on the sensitivity, yet to be determined.

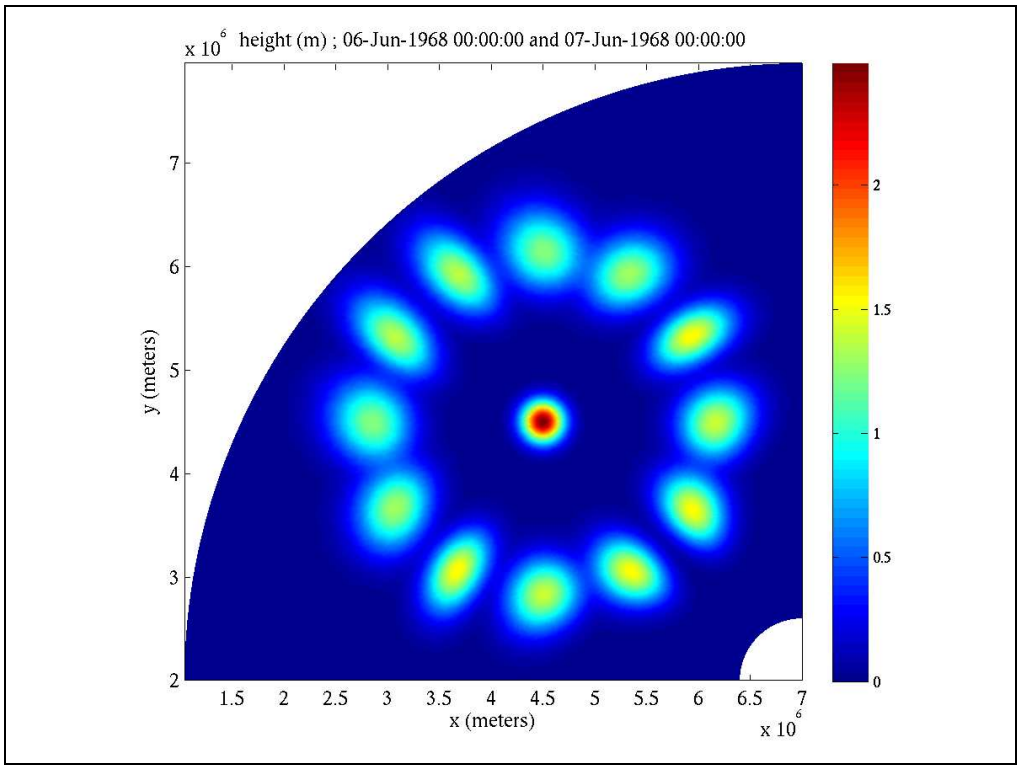

Figure 4-4 Test case 2.1c, PR1 (first order scheme) used. H_{m0} (m) plotted.

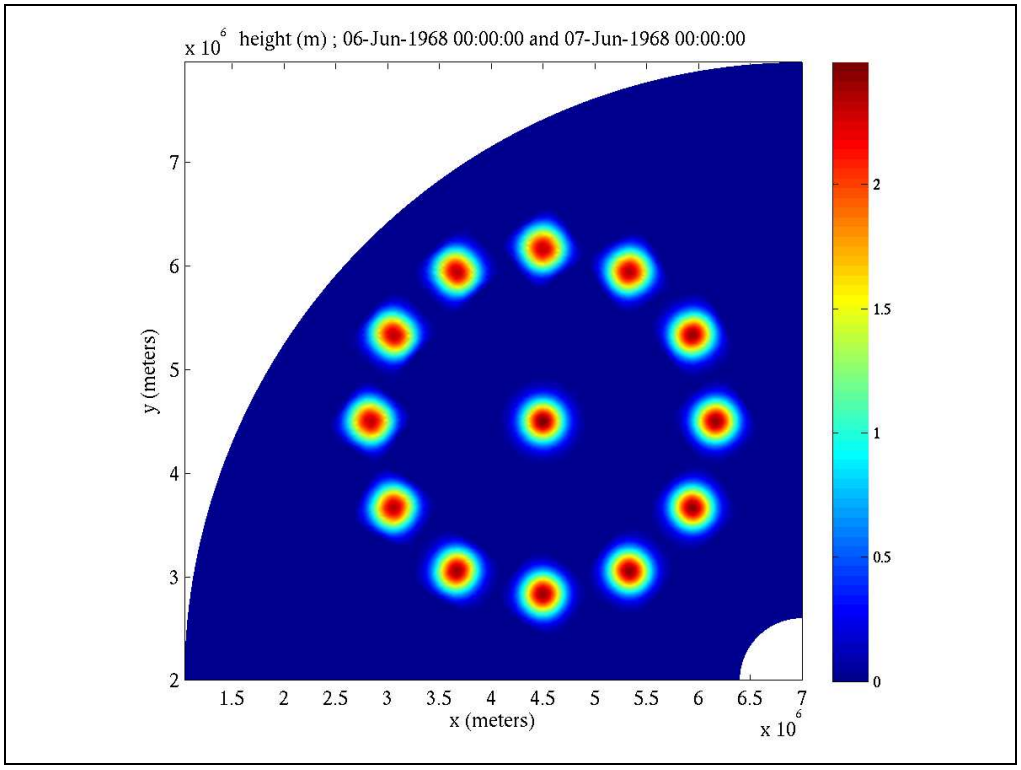

Figure 4-5 Test case 2.1c, PR2, no GSE controls (i.e. T_s =0) . H_{m0} (m) plotted.

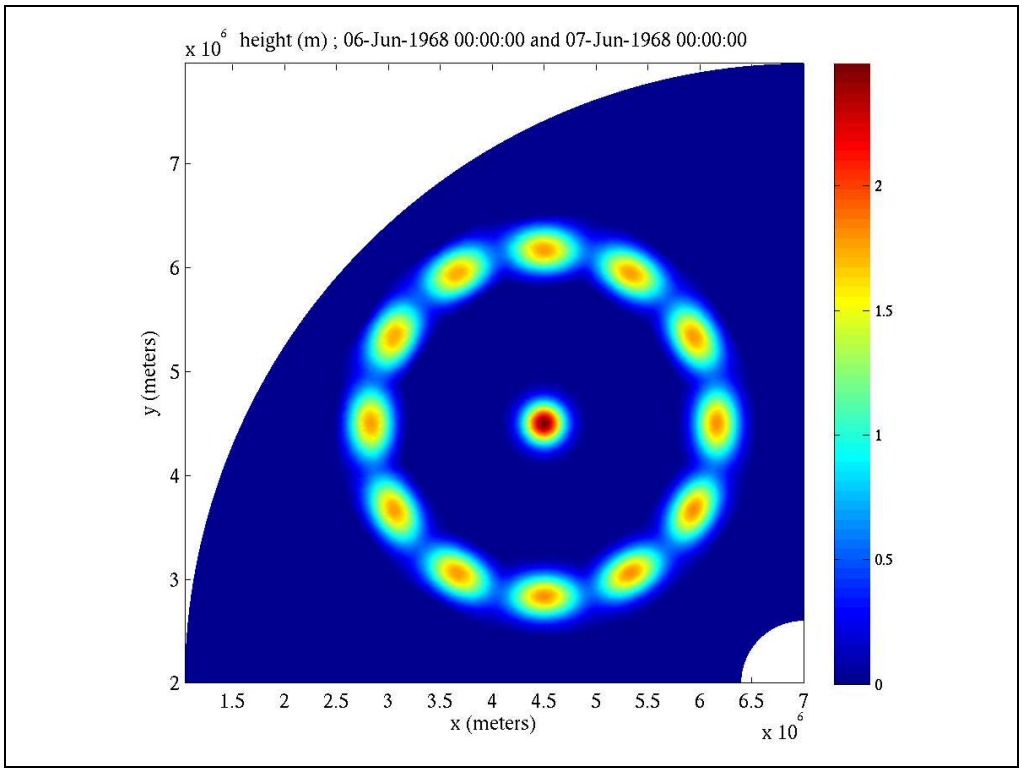

Figure 4-6 Test case 2.1c, PR2, GSE: T_s =4 hours. H_{m0} (m) plotted.

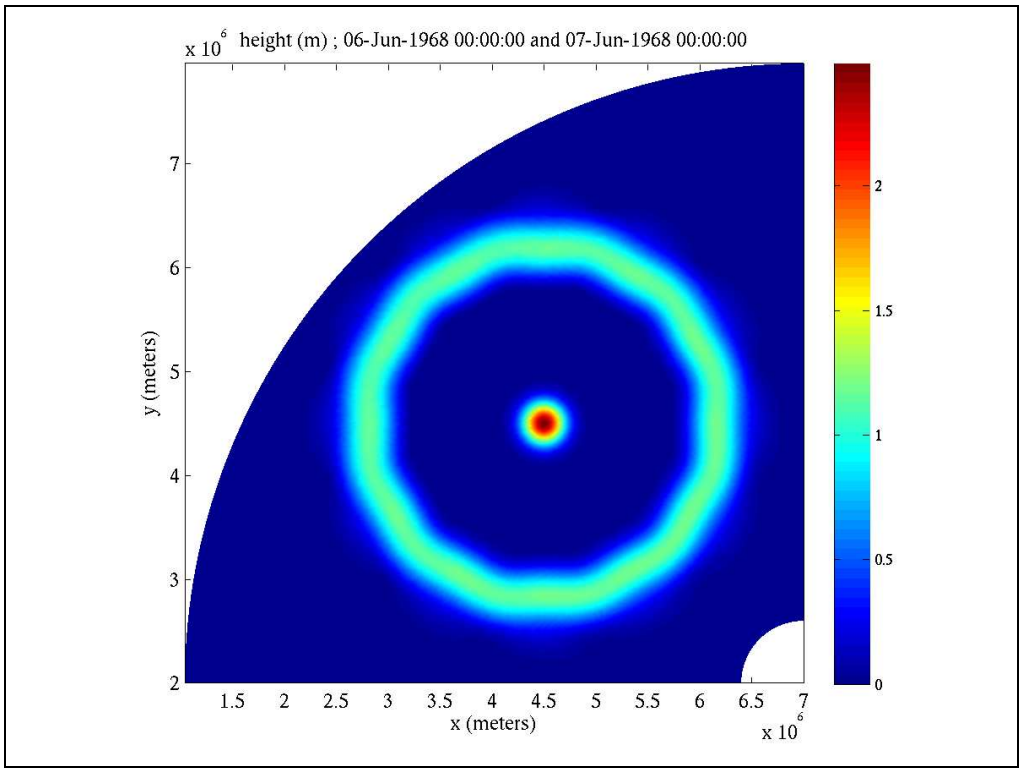

Figure 4-7 Test case 2.1c, PR2, GSE: T_s =24 hours. H_{m0} (m) plotted.

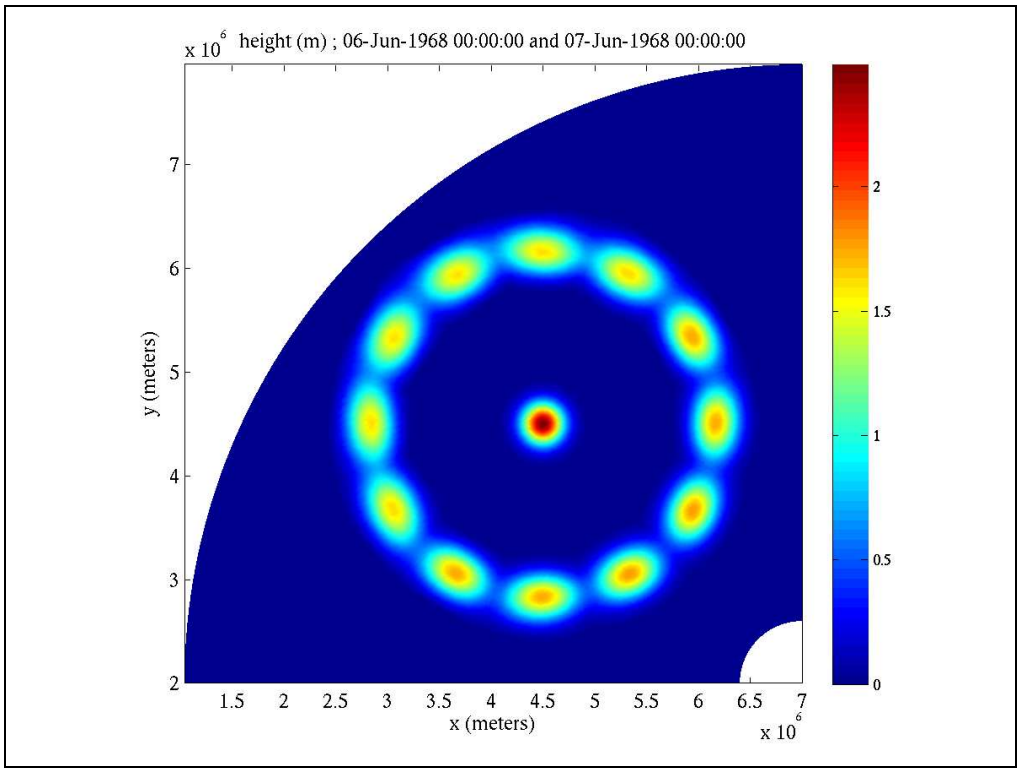

Figure 4-8 Test case 2.1c, PR3 w/default GSE settings, $\alpha_s = \alpha_n = 1.50$. H_{m0} (m) plotted.

4.2.4. Test case tp2.2

Summary: Propagation of Gaussian spike over half the globe (two-dimensional), a test case included in public release WW3 v2.22.

Settings:

- Δt_g =2200 s , Δt_{xy} =2200 s ,
- duration= 12 days , 19680606 000000 to 19680618 000000
- n_x = 193 , n_y = 93 , number of sea points= 17381
- Δx = 1.0 ° Δy = 1.0 °
- Longitude range (°.) : -6.00 186.00
- Latitude range (°.) : -46.00 46.00
- n_f = 3 f_1 = 0.0368 γ_f =1.1
- n_θ = 24
- deep water
- input spectrum: H_s=2.5 m f_p =0.04 Hz , θ_m =270.0°

Five (5) cases tested:

- rectilinear only
- propagation schemes 1,2,3 without GSE correction

- propagation scheme 2 with T_s = 345600. s
- propagation scheme 3 with $\alpha_s = \alpha_n$ = 1.50

Baseline: equivalent v3.13 simulations
Result: exact match.

4.2.5. Test case tp2.3

Summary: Garden Sprinkler Effect tests (two-dimensional propagation of Gaussian spike, meters grid), a test case included in public release WW3 v2.22.

Settings:

- n_θ = 24
- n_f = 15 , f_1 = 0.0466 , γ_f =1.1
- Δt_g =3600 s , Δt_{xy} =3600 s
- n_x = 48 , n_y = 38 , number of sea points= 1656
- Δx = 100 km Δy = 100 km
- X range (km) : -600.00 4100.00
- Y range (km) : -600.00 3100.00
- deep water
- input spectrum: H_s=2.5 m f_p =0.1 Hz , θ_m =240.0°

Four (4) cases tested:

- rectilinear only
- propagation schemes 1,2 without GSE correction
- propagation scheme 2 with T_s = 345600. s
- propagation scheme 3 with $\alpha_s = \alpha_n$ = 1.50

Baseline: equivalent v3.13 simulations
Result: exact match.

4.2.6. Test case tp2.4

Summary: Two-dimensional propagation of a single Gaussian spike under angle with a spherical (latitude-longitude) rectilinear grid, with optional extension to multiple spikes/angles.

Settings:

- n_θ = 12
- n_f = 3 , f_1 = 0.0368 , γ_f =1.1
- Δt_g =2200 s , Δt_{xy} =1100 s
- n_x =225, n_y =106, Number of sea points = 14548
- $\Delta x = \Delta y$ = 0.35°
- Longitude range : 183.40° 262.82°
- Latitude range : 25.10° 62.33°

- depth grid: bathymetry from NOAA's GEODAS database; region corresponds to northeast Pacific Ocean. Due to scale, a vast majority of grid points are either on land or deep water.
- input spectrum: H_s=2.5 m f_p =0.0.04 Hz , θ_m =210.0°
- duration=1 day, 20080522 000000 to 20080523 000000

Three (3) cases tested:

- rectilinear only
- propagation schemes 1,2 without GSE correction
- propagation scheme 2 with T_s = 48 hours

Baseline: equivalent v3.13 simulations
Result: exact match in deep water; near-exact match near coastline

Figure 4-9 shows the result for test case tp2.4 with propagation scheme 2, without GSE controls (i.e. T_s =0), after 19 hours of propagation. Note that the end distribution is not quite ellipsoid (broader at the top), due to plotting with Mercator projection. The increased diffusion compared to tp2.1b is due to the relatively coarse resolution, i.e. the spatial variation of each signal is described with fewer grid points, compared to tp2.1b, which has fives times the number of sea points.

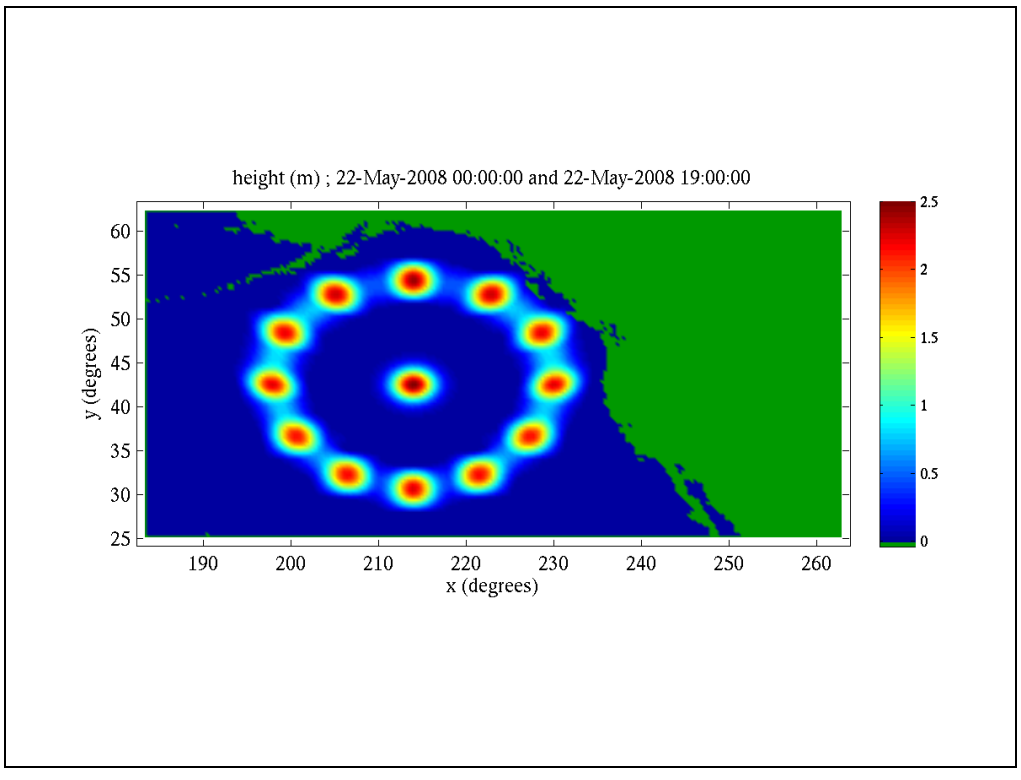

Figure 4-9 Test case tp2.4 after 19 hours, PR2: T_s =0. H_{m0} (m) plotted.

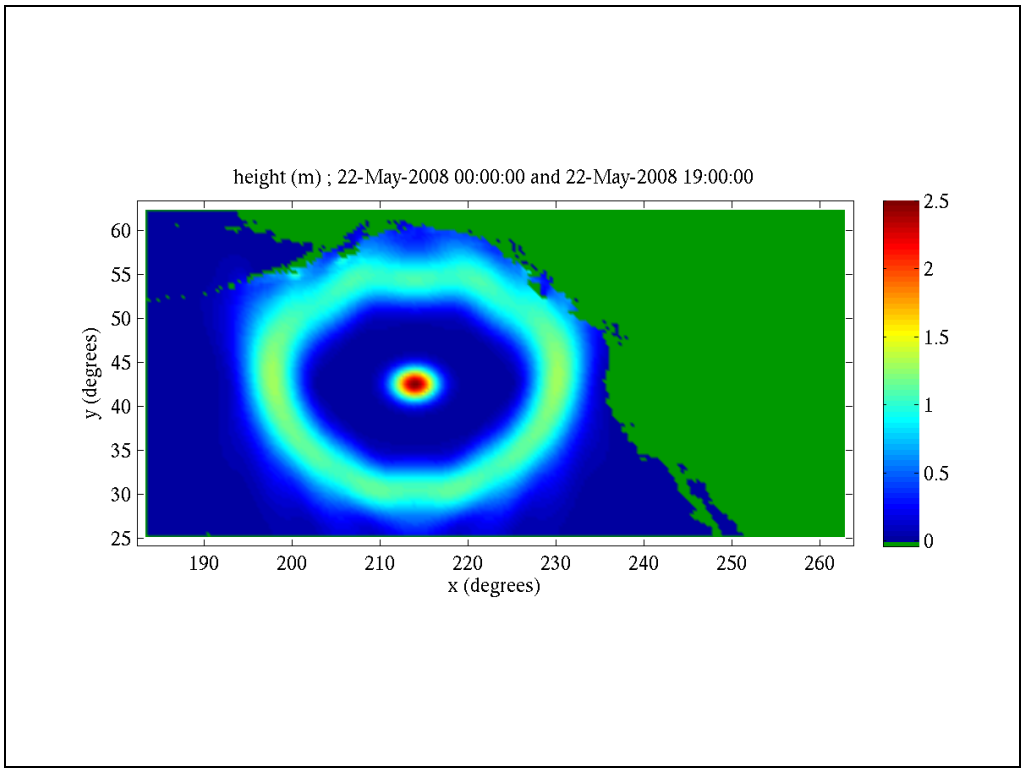

Figure 4-10 Test case tp2.4 after 19 hours, PR2: T_s =48 hours. H_{m0} (m) plotted.

4.2.7. Test case tp2.4c

Summary: This is like tp2.4 {two-dimensional propagation of a single Gaussian spike under angle with a spherical (latitude-longitude) grid, with optional extension to multiple spikes/angles.}, except that it is on a true curvilinear grid. Relative to our meters-grid test case 2.1c, this one has fewer grid points (i.e. less well-resolved signals), has less variation in grid spacing and is more orthogonal (i.e. closer to rectilinear).

Settings:

- n_θ = 12
- n_f = 3 f_1 = 0.0368 γ_f =1.1
- Δt_g =2200 s , Δt_{xy} =1100 s
- n_i, n_j = 169 , 139
- number of sea points= 16117
- $\Delta x, \Delta y$ variable
- Longitude range : 183.49° to 262.51°
- Latitude range : 25.11° to 62.10°
- depths: like tp2.4, the region is the northeast Pacific Ocean, with depths taken from the GEODAS database. However, for tp2.4c, the grid is true curvilinear, on a Lambert Conformal projection. The grid is identical to that used for the FNMOC (Fleet Numerical Meteorology and Oceanography Center) COAMPS

(Coupled Ocean/Atmosphere Mesoscale Prediction System, Hodur 1997) EPAC (East Pacific) grid.

- input spectrum, duration same as tp2.4

Three (3) cases tested (curvilinear only):

- propagation scheme 1
- propagation schemes 2,3 without GSE correction
- propagation scheme 2 with T_s = 48 hours

Verification method (qualitative only): visual comparison with results from tp2.4.

Figure 4-11 shows the result for test case tp2.4c with propagation scheme 2, without GSE controls (i.e. T_s =0). Encouragingly, the result looks very similar to that of Figure 4-9.

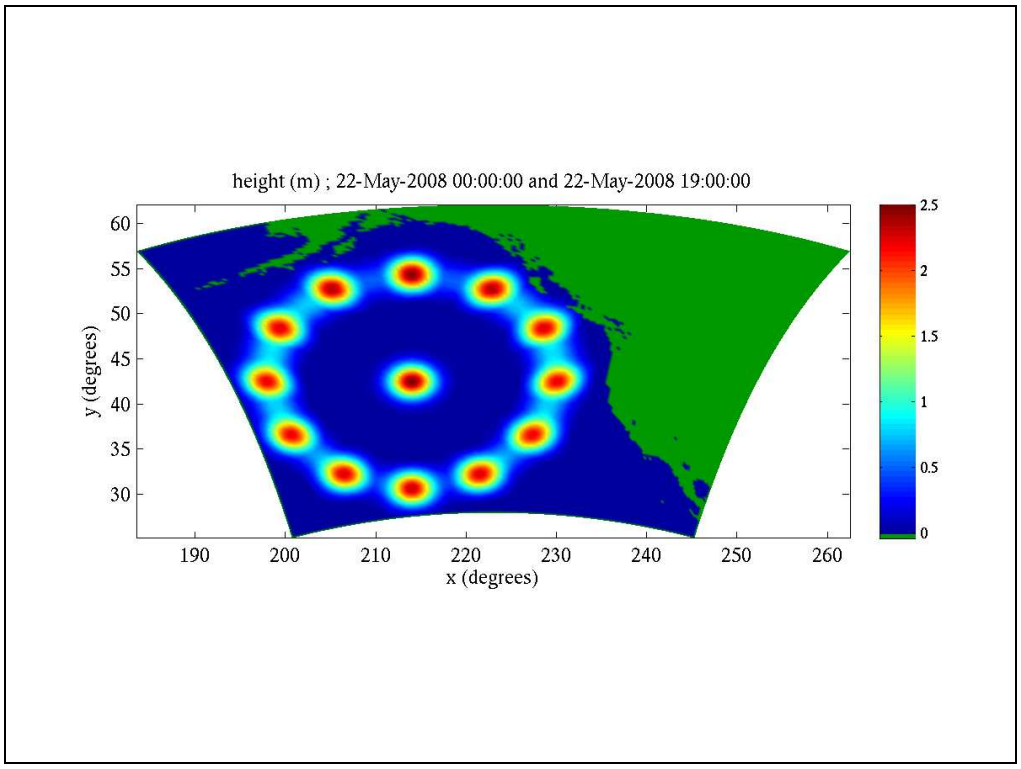

Figure 4-11 Test case tp2.4c after 19 hours. H_{m0} (m) plotted.

4.2.8. Test case tp2.5c

Summary: This is like tp2.4c {two-dimensional propagation of a single Gaussian spike under angle with a spherical (latitude-longitude) curvilinear grid, with optional extension to multiple spikes/angles.}, but this is for a hypothetical ice-free Arctic up to 88°N. The latter is noteworthy, as modeling of such a region with a rectilinear grid would be highly inefficient[1]. The grid is a polar stereographic grid, used for a COAMPS Arctic model, and was provided by FNMOC. The number of sea points, and therefore computation time, is similar to that of tp2.1c.

Settings:

- n_θ = 12, 36
- n_f = 3 f_1 = 0.0368 γ_f =1.1
- Δt_g =600 s , Δt_{xy} =600 s
- n_i, n_j = 361 , 361
- number of sea points= 62274
- $\Delta x, \Delta y$ variable
- Longitude range : ±180° (all longitudes)
- Latitude range : minimum: 54° to 65°N; maximum: North Pole.
- depths: The region is the Arctic Ocean, with depths taken from the GEODAS database. The grid is true curvilinear, a stereographic projection.
- ice: The ice representation feature of WW3 is not used here. Rather, this is meant to be a hypothetical scenario for an ice-free Arctic. Land is artificially applied north of 88°N, and the grid does extend all the way to the pole, so this can be imagined as polar ice. However, our motivation for blocking waves north of 88°N is actually unrelated to ice, but is due to a singularity at the North Pole, see below.
- input spectrum: JONSWAP spectrum, with spatial distribution of a Gaussian spike centered at 150°W, 83°N.
- duration: 12 or 24 hours

Three (3) cases tested (curvilinear only):

- propagation schemes 2,3 without GSE correction
- propagation scheme 2 with T_s = 2500 s
- propagation scheme 3 with default averaging: $\alpha_s = \alpha_n$ = 1.50

Verification method (qualitative only): As mentioned above, it is impractical to apply a rectilinear grid to this region. Therefore, verification is simply inspection for obvious, visible problems.

[1] For example, a $\frac{1}{5}°$ rectilinear grid extending to 88°N would require a propagation time step of approximately 30 s, or 20 time steps for each one used in the curvilinear grid here, due to the former conforming to the converging meridians, and would use about twice as many grid points, with very high aspect ratio grid cells near the pole, and high Δx resolution there essentially wasted.

To demonstrate that the tp2.5c grid is not, in fact, rectilinear, Figure 4-12 shows the points of the grid, plotted on a Mercator projection. Figure 4-13 shows the result for test case tp2.5c with propagation scheme 3, without GSE controls (i.e. $\alpha_s = \alpha_n = 0$) and with 36 directional bins, after 17 hours of propagating an isotropic swell field. The results look very reasonable. One nice feature is that hardly any Garden Sprinkler effect is visible, even without GSE controls active, suggesting for this case that using this moderately high directional resolution is sufficient to counter the problem. Distortion of the swell signal north of Siberia is evident; this is a physical effect, resulting from interaction with the relatively broad continental shelf there (depth contours shown).

Figure 4-14 shows the result for test case tp2.5c with propagation scheme 3, without GSE controls (i.e. $\alpha_s = \alpha_n = 0$) similar to the prior example, but with 12 directional bins, and after only 8 hours of propagating the isotropic swell field. With this very coarse directional resolution—a typical operational model uses 24 bins—and with GSE controls disabled, the individual directional components are visible here (i.e. large GSE). The most remarkable feature, however, is the very inhomogeneous distortion of each signal, with the directional components from 150°, 180°, and 210° (i.e. the three signals propagating toward a northerly direction) markedly focused. Though this problem might at first appear to be due to some problem of the geographic propagation scheme or the curvilinear implementation, it is actually due to the method of defining wave directions in the specification of the spectral grid. Consider, for example, a single swell field, broadly distributed in geographic space, but with all energy contained in the directional bin 180°. In this case, all the energy, everywhere in the swell field, is propagating toward the North Pole, and so as it nears the North Pole, there is a visible convergence of all this energy, and at the North Pole, a singularity. To some extent, this can be blamed on a somewhat unreasonable initial condition, but nevertheless it does imply a likely problem for wave modeling in the Arctic. The problem can be corrected by changing the model to allow the user to define an alternate reference point for defining spectral directions (the farther away, the better), but the method details are not obvious.

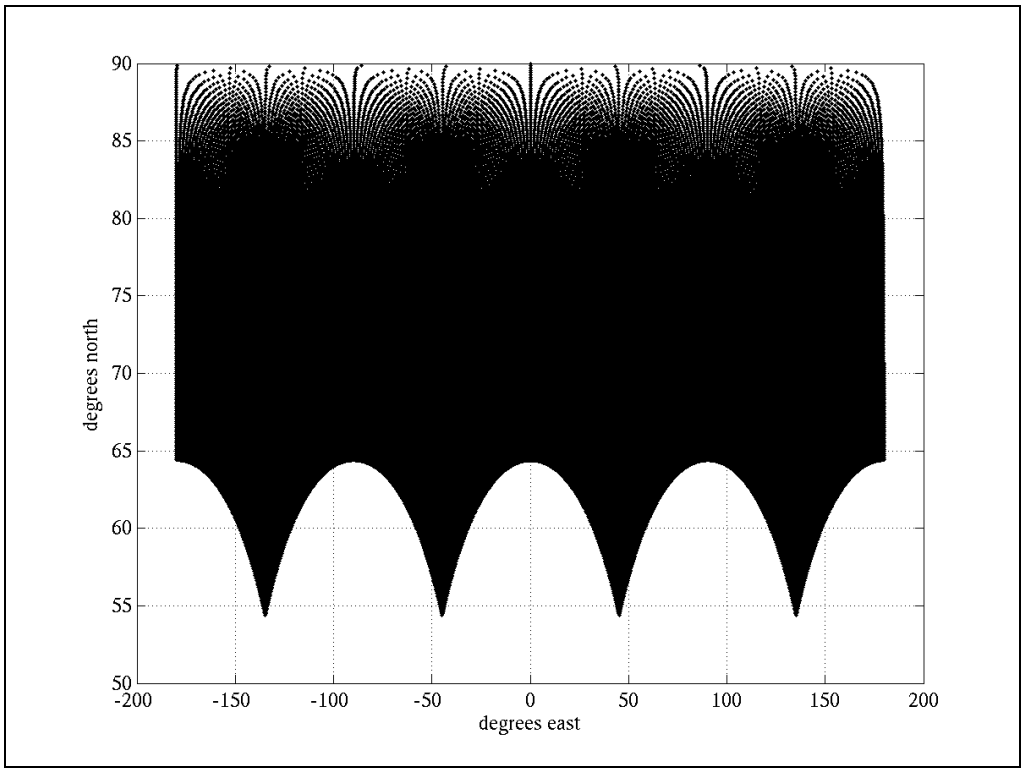

Figure 4-12 Grid points for test case tp2.5c

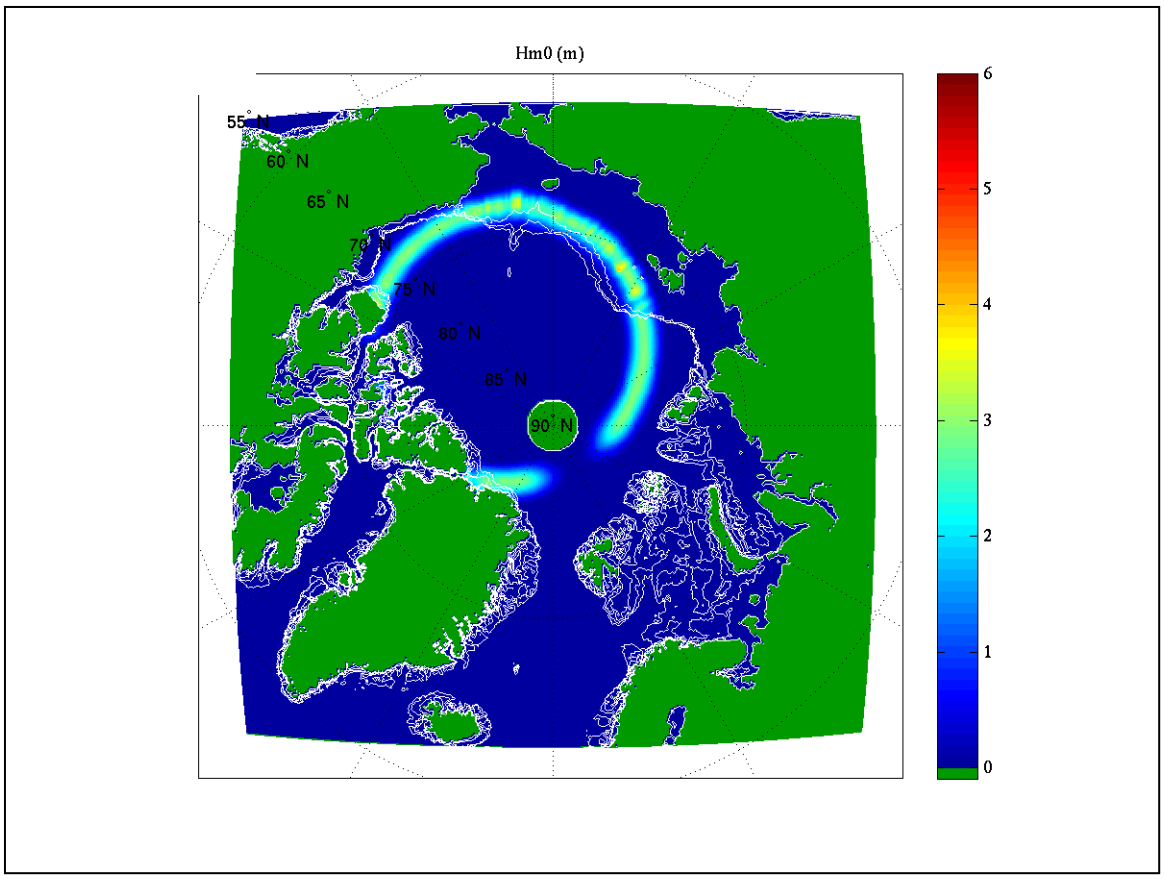

Figure 4-13 Test case tp2.5c after 17 hours, without GSE controls, and with 36 directional bins. White lines indicate contours for 0, 100, 200, 300 m. H_{m0} (m) plotted with orthographic projection (2D) from above the North Pole.

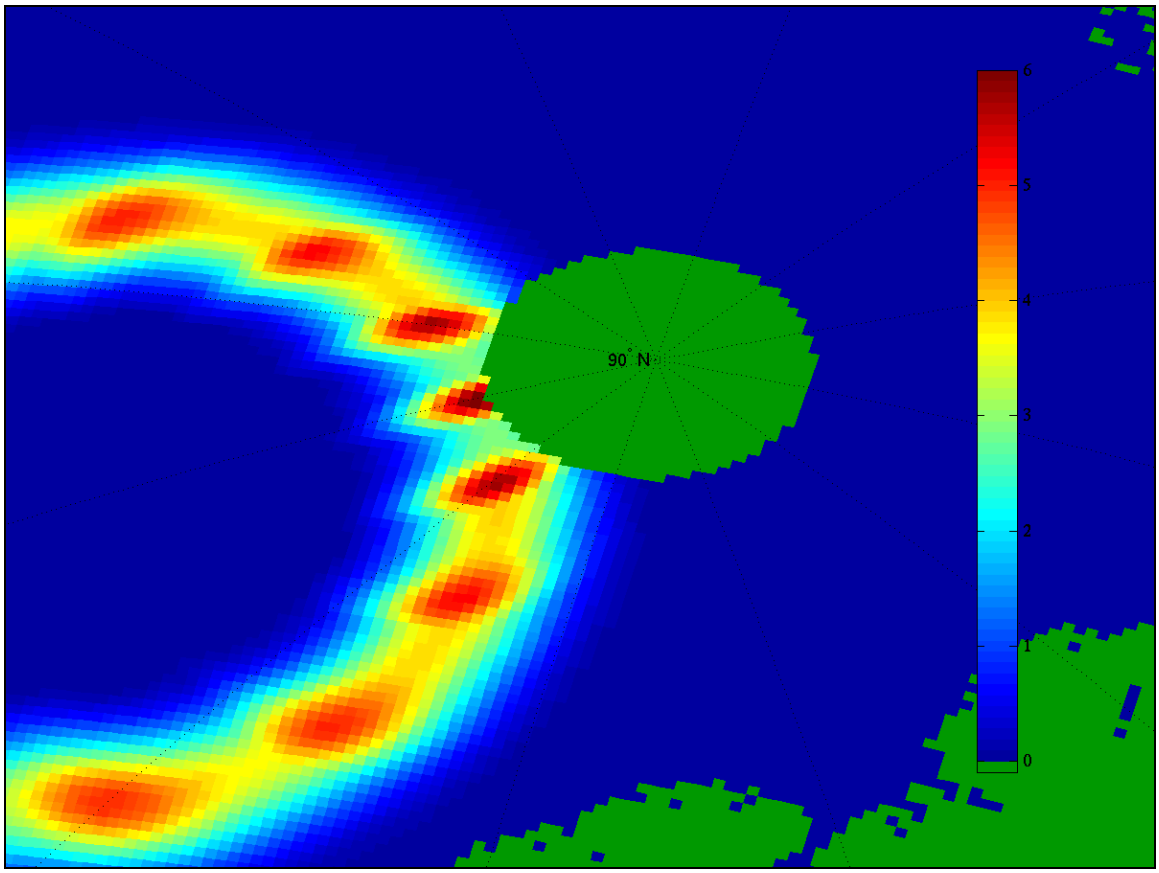

Figure 4-14 H_{m0} (m) plotted for test case tp2.5c after 8 hours, without GSE controls, and with 12 directional bins. The globe is plotted as a 3D image.

4.3. Two-dimensional propagation with source terms

4.3.1. Test case tps1

Summary: Case for simultaneously testing propagation and source terms. Grid is rectilinear and spherical (latitude-longitude), corresponding to northeast Pacific Ocean (identical to 2.4). Wind field is a fictitious extra-tropical cyclone. The input wind field is also rectilinear, but is not at the same resolution as the computational grid.

Settings:

- $n_\theta = 36$
- $n_f = 25$, $f_1 = 0.0418$, $\gamma_f = 1.1$
- $\Delta t_g = 450$ s , $\Delta t_{xy} = 450$ s
- n_x , n_y ,number of sea points, same as tp2.4
- $\Delta x = \Delta y = 0.35°$
- depth grid, computational grid: same as tp2.4 (northeast Pacific Ocean)
- initial condition, JONSWAP, using fetch-limited approximation
- duration=3 days, 20080422 000000 to 20080425 000000
- no GSE corrections

- Default Tolman and Chalikov (1996) physics used
- Wind field longitude range 183.4° to 263.0°
- Wind field latitude range 25.1° to 62.5°
- Wind field n_x =399, n_y = 188, $\Delta x = \Delta y$ =0.2°

One case tested: rectilinear, with propagation scheme 2.
Verification: visual comparison to equivalent v3.13 simulation
Result: no problems detected.

Figure 4-15 shows the result for test case tps1 with propagation scheme 2, without GSE controls (i.e. T_s =0).

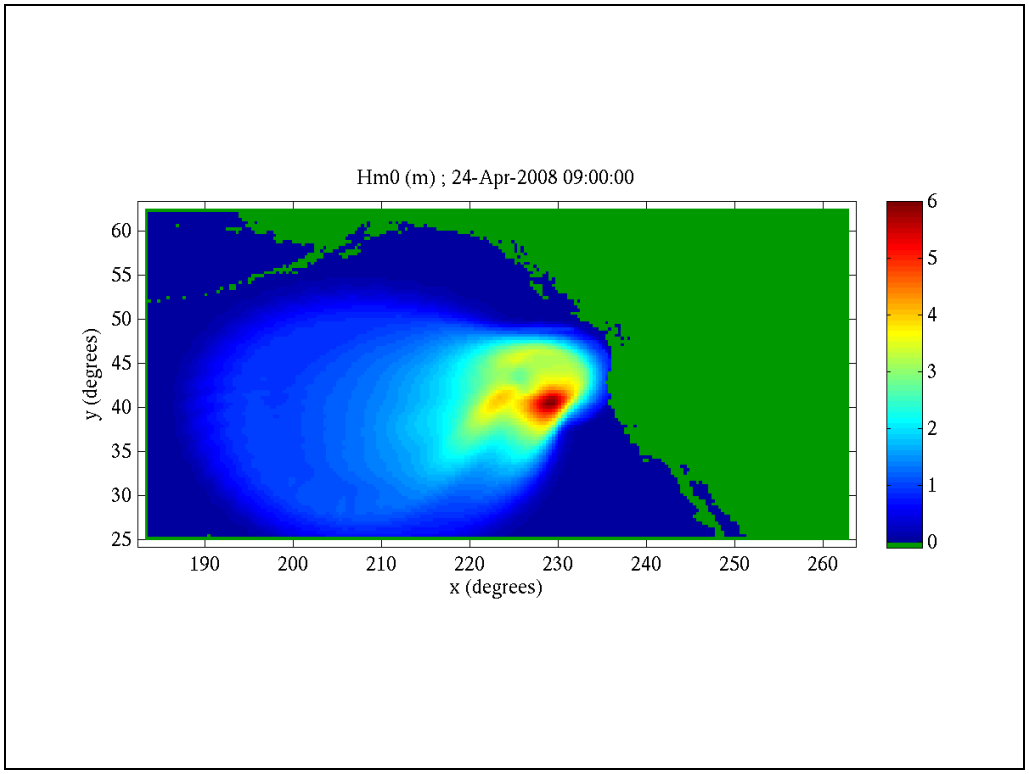

Figure 4-15 Test case tps1 after 2 days, 9 hours. . H_{m0} (m) plotted.

4.3.2. Test case tps1c

Summary: Like tps1, this is a case for simultaneously testing propagation and source terms on a spherical (latitude-longitude) grid, corresponding to northeast Pacific Ocean, with fictitious extra-tropical cyclone for wind field. The wind field input is rectilinear and is identical to the input field used for tps1. Unlike tps1, tps1c is on a true curvilinear grid, a Lambert Conformal projection (identical to 2.4c).

Settings:

- n_θ , n_f , f_1, γ_f , Δt_g , Δt_{xy} same as tps1

- computational grid: curvilinear (Lambert Conformal projection), same as 2.4c, corresponding to FNMOC COAMPS grid for northeast Pacific Ocean.
- depths: same as 2.4c, taken from NOAA GEODAS database.
- initial condition, JONSWAP, using fetch-limited approximation, same as tps1
- no GSE corrections
- Default Tolman and Chalikov physics used
- Wind file: rectilinear, same as tps1

One case tested, with propagation scheme 2.

Verification: visual comparison with tps1
Result: no problems detected

Since the source/sink terms are handled in a separate step from the propagation, we do not expect any problems associated with wave generation/decay on the curvilinear coordinates. This test case is designed to verify that. Figure 4-16 shows the result for test case tps1c with propagation scheme 2, without GSE controls (i.e. T_s =0). The desired outcome is observed: again, the result looks very similar to its rectilinear counterpart (Figure 4-15).

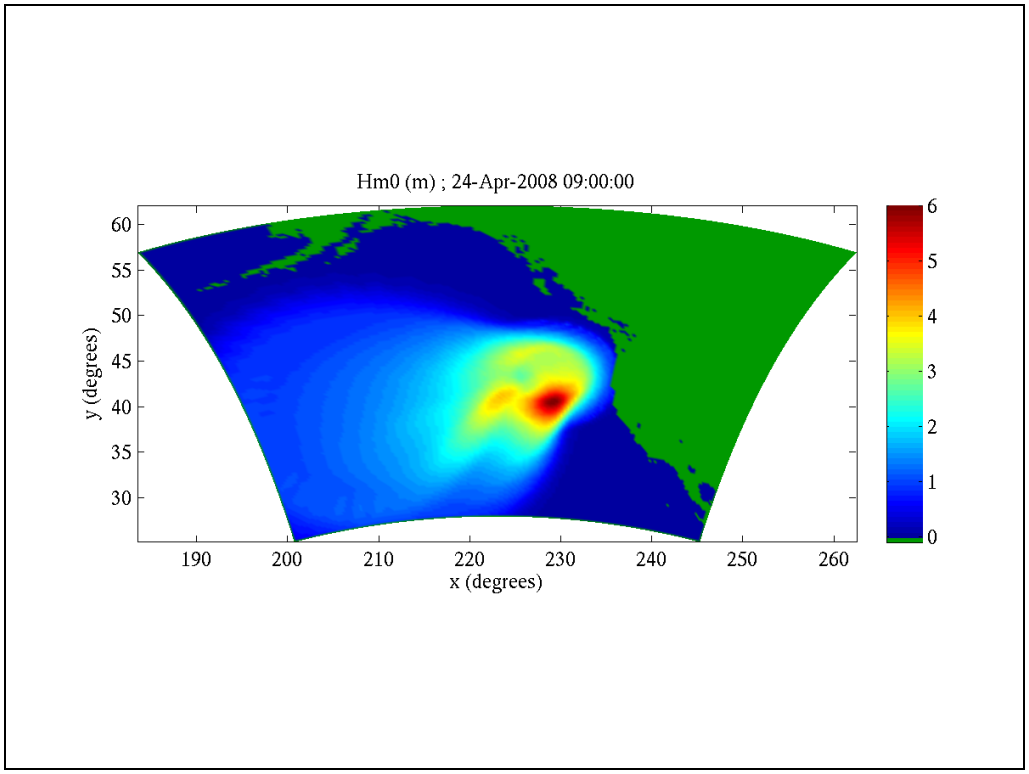

Figure 4-16 Test case tps1c after 2 days, 9 hours. . H_{m0} (m) plotted.

4.4. Wind field processing

4.4.1. Test case tw1

Summary: Test case for wind field pre-processing ("ww3_prep") program. This one is for rectilinear, spherical (latitude-longitude) computational grid with rectilinear wind input on similar grid. Location is the Gulf of Mexico. Wind event is Hurricane Ivan.

Settings:

- n_θ, n_f, Δt,etc. : N/A (no wave spectrum calculations)
- input spectrum: N/A
- n_x = 161 , n_y = 131 , number of sea points= 15386
- $\Delta x = \Delta y$ =0.1°
- Longitude range : 262.00° to 278.00° (Gulf of Mexico)
- Latitude range : 18.00° to 31.00°
- depths taken from NRL DBDB2 (2' resolution, coarsened to 0.1°)
- wind input file:
  - created from FNMOC COAMPS "Central America" grid.
  - n_x = 81 , n_y = 66
  - $\Delta x = \Delta y$ =0.2°
  - 0000 UTC 15 Sept 2004 – 0000 UTC 16 Sept 2004
  - Longitude, Latitude range , same as computational grid
  - duration : 20040915 000000 to 20040916 000000

Verification by reading WW3 binary containing re-gridded wind fields and plotting against similar fields with v3.13 binary. The new code was tested using both rectilinear and pseudo-curvilinear grid specification.

Result: negligible differences.

4.4.2. Test case tw2

Summary: Like tw1, except with higher computation grid resolution. This is a more severe test of the wind field pre-processing ("ww3_prep") program than tw1.

Settings:

- n_θ, n_f, Δt,etc. : N/A (no wave spectrum calculations)
- input spectrum: N/A
- n_x = 151 , n_y = 181 , number of sea points= 23160
- $\Delta x = \Delta y$ =2'
- Longitude range : 271.00° to 276.00°
- Latitude range : 25.00° to 31.00°(north central Gulf of Mexico)
- depths taken from NRL DBDB2 (2' resolution)
- wind input file: same as tw1 (rectilinear, $\Delta x = \Delta y$ =0.2°)
- duration : 20040915 000000 to 20040916 000000

Verification by reading WW3 binary containing re-gridded wind fields and plotting against similar fields with v3.13 binary. The new code was tested using both rectilinear and pseudo-curvilinear grid specification.

Result: negligible differences.

Figure 4-17 shows one of the wind fields for tw2, after being interpolated onto the computational grid. [The result from only one of the two models is shown, since they are essentially identical.]

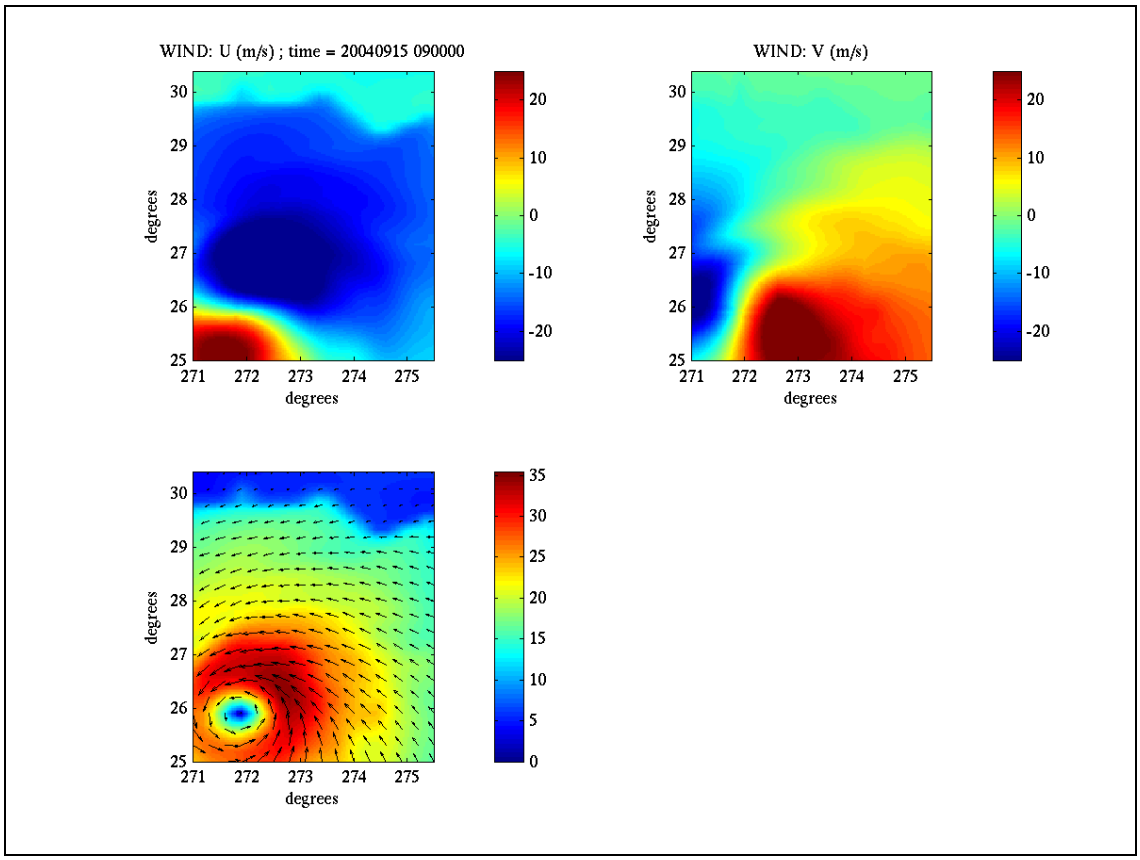

Figure 4-17 Test case tw2. These plots are created from the WW3 binary which contains wind fields interpolated by WW3 onto its computational grid. *x*- and *y*-components of 10 m wind velocity (upper panels) and 10 m wind speed with vectors (lower panel).

4.4.3. Test case tw3b

Summary: Third test case for wind field pre-processing ("ww3_prep") program. Computational grid is rectilinear and Cartesian (meters). Wind input is also on a rectilinear grid. Thus, this test case is fairly similar to tw2, the difference being that this one (tw3b) is on a meters grid and uses an artificial wind vortex.

Settings:

- n_θ, n_f, Δt,etc. : N/A (no wave spectrum calculations)
- input spectrum: N/A
- computational grid:
  - rectilinear
  - X range : 1000 km to 6970 km
  - Y range : 2000 km to 7980 km
  - Δx =30 km , Δy =20 km
  - n_x = 200 , n_y = 300 , number of sea points= 59004
- deep water
- wind input:
  - artificial vortex
  - rectilinear specification
  - X range : 1000 km to 7000 km
  - Y range : 2000 km to 8000 km
  - Δx =62.5 km , Δy =40 km
  - n_x = 97 , n_y = 151
  - duration: 20080422 000000 to 20080423 000000

Verification by reading WW3 binary containing re-gridded wind fields and plotting against similar fields with v3.13 binary.

Result: negligible differences.

Figure 4-18 shows one of the wind fields for tw2, after being interpolated onto the computational grid. [The result from only one of the two models is shown, since they are essentially identical.]

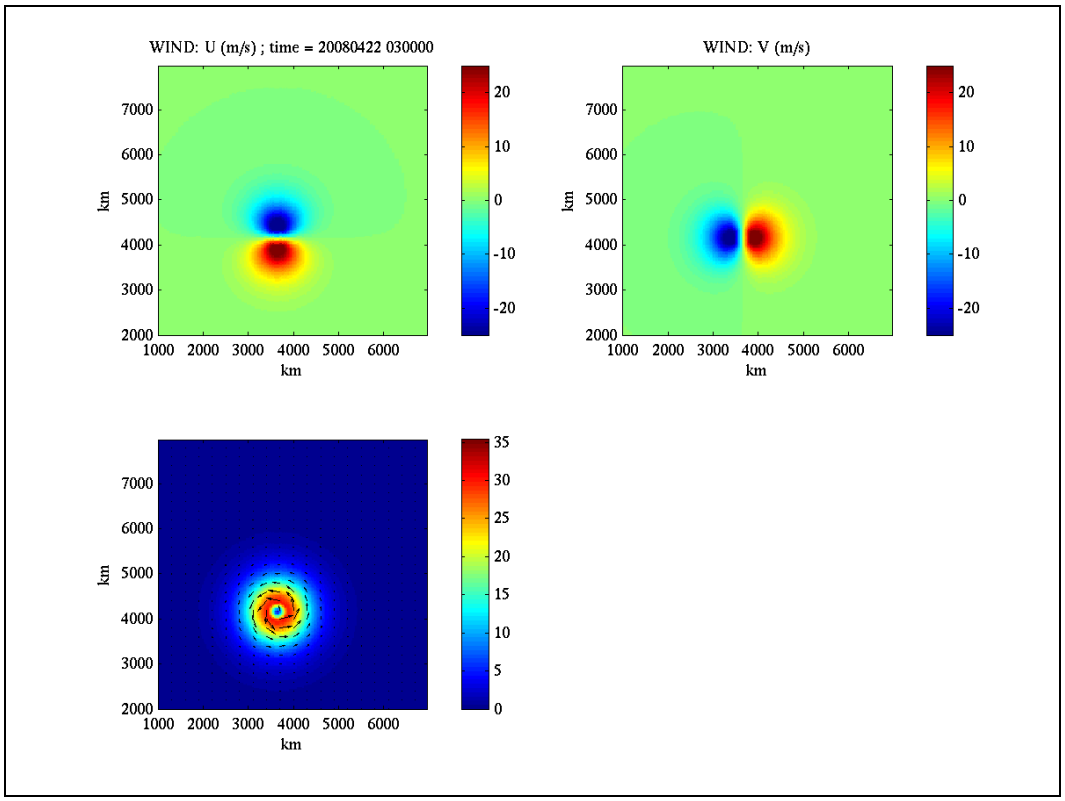

Figure 4-18 Test case tw3b. These plots are created from the WW3 binary which contains wind fields interpolated by WW3 onto its computational grid. *x*- and *y*-components of 10 m wind velocity (upper panels) and 10 m wind speed with vectors (lower panel). Time is three hours after the start of the input wind field.

4.4.4. Test case tw3c

Summary: Fourth test case for wind field pre-processing ("ww3_prep") program. Here, the computational grid is curvilinear (upper left quadrant of a polar grid) and Cartesian (meters), identical to the grid of tp2.1c. The wind input file is identical to that of tw3b: describing a fictitious moving vortex, defined on a rectilinear grid.

Settings:

- Computational grid: curvilinear, meters identical to tp2.1c (n_x = 226 , n_y = 331, number of sea points= 73696)
- n_θ , n_f , Δt ,etc. : N/A (no wave spectrum calculations)
- input spectrum: N/A
- X range (km) : 1040.39 to 7000.00
- Y range (km) : 2000.00 to 7959.61
- deep water
- wind input file, artificial vortex, rectilinear specification, see tw3b description

Verification is performed by taking the rectilinear input grid and the curvilinear output grid (created using the "dry run…no wave computation" option available in WW3),

regridding both to a common relatively fine (Δx =3125 m , Δy =2000 m) rectilinear grid using built-in Matlab functions, then comparing quantitatively.

Result: Some differences, but very localized.

Figure 4-19 shows the comparison on the fine rectilinear grid for the *x* component of the wind speed. The entire computational grid is not shown; instead, a sub-region is shown, *x* range 3500 km to 4500 km and *y* range 4000 km to 5000 km. This corresponds to the region of the grid where differences are most noticeable, near the center of the vortex. The problem area is very localized, near the center of the vortex, and it is possible that the Matlab interpolation functions are responsible for some of the differences.

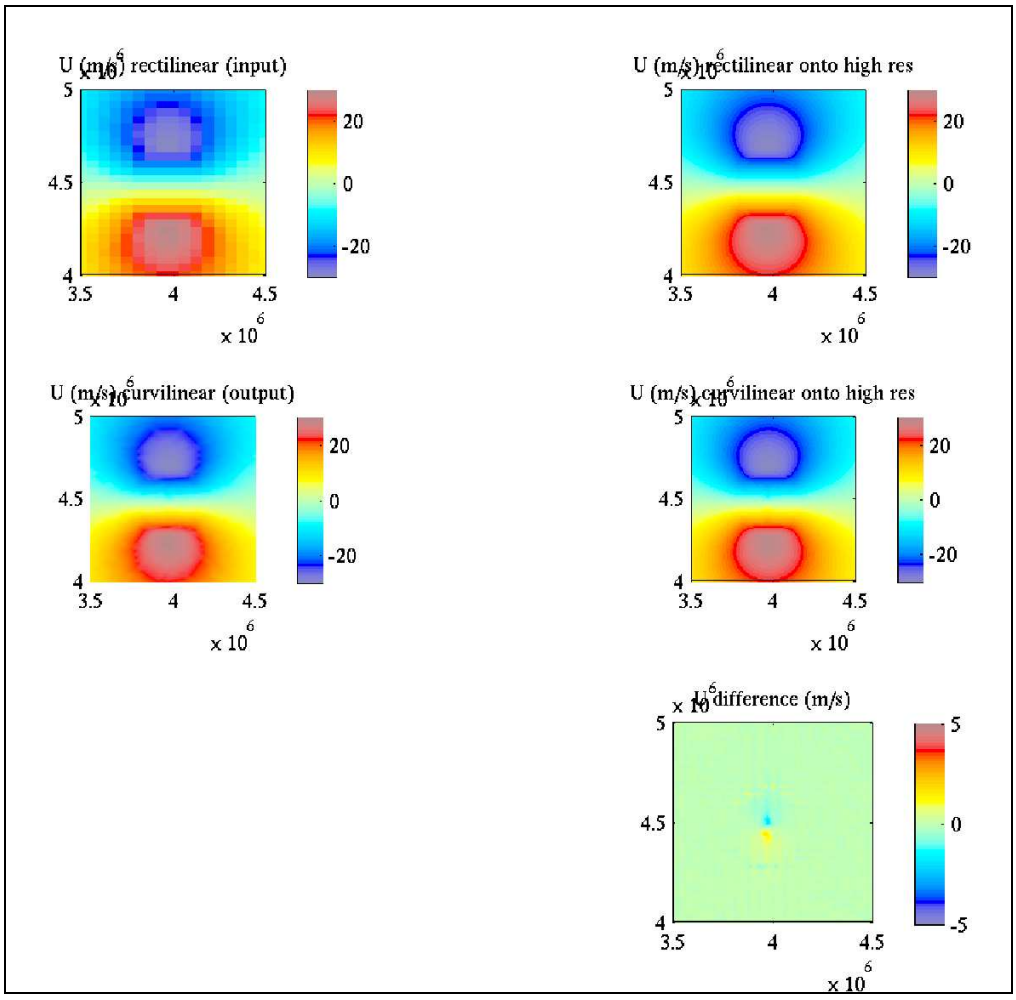

Figure 4-19 Comparison of tw3c input (top) versus output (center), with the difference shown (lower right). U component is shown, and graphics are zoomed in, i.e. only a subset of the actual domain is shown. Time is 12 hours after the start of the input wind field.

5. Discussion of alternate method: local rotation method

5.1. Method Description

The "PHIDIAS method" implemented as described above, was not the first method applied. One method was implemented prior. That other method is described in this section.

The NCOM (Navy Coastal Ocean Model) documentation (Martin, 2000) notes three basic difficulties that must be addressed when implementing a curvilinear system for an ocean model:

1. The horizontal grid spacing Δx, Δy is non-uniform and must be stored as arrays.
2. The fluxes between grid cells must account for the changing size of the grid cells.
3. Correction terms are needed to account for exchange between *u* and *v* momentum due to horizontal transport along curving grid coordinates.

Item (1) is a relatively obvious coding change, though of course there are several other variables derived from Δx, Δy which must therefore also be made spatially non-uniform. Item (2) is very simple in the case of both NCOM and WW3, since both use flux methods: the flux "out of" a given grid cell is necessarily equivalent to the flux "into" its neighbor along that cell boundary, so conservation is maintained regardless of any curving of the grid.

For item (3), the WW3 approach must differ somewhat from that of NCOM, since the directionality is handled not by separate tracking of the vector described by *u* and *v*, but rather via the spectral density as function of propagation direction, θ. However, a fundamentally similar procedure can be used. Specifically, during the propagation stage, the *propagation direction is adjusted based on the local rotation of the grid axis*. This method was implemented and is referred to here as the *local rotation method*.

The flux method used in WW3 is a two step process. First, the fluxes are calculated for all cell boundaries. Second, at the "propagation stage", the fluxes are applied to update the spectral density, which is defined at the cell centers. With the pre-existing WW3, giving the *x*-propagation as an example, the non-dimensional quantity $\mu_x = \dfrac{C\cos(\theta)\Delta t}{\Delta x}$ is employed at the flux calculation stage. (Here, *C* represents the combined propagation speed, which includes the wave group velocity, currents, and grid motion). At the propagation stage, μ_x is not used. To summarize, the procedure in WW3 v3.13 is:

1. Flux calculation stage: for all values of *i*,
   - calculate $\mu_x^{(i+0.5)}$ from pre-calculated values of $\mu_x(x,y) : \mu_x^{(i+0.5)} = 0.5(\mu_x^{(i)} + \mu_x^{(i+1)})$
   - calculate density at cell boundaries, e.g. $N_b^{(i+0.5)}$ using ULTIMATE QUICKEST scheme
   - calculate flux at cell boundaries, e.g. $F^{(i+0.5)} = \mu_x^{(i+0.5)} N_b^{(i+0.5)}$
2. Propagation stage: for all values of *i*,

- update local density using fluxes at the two boundaries: $N^{(i,t)} = N^{(i,t-1)} + F^{(i-0.5)} - F^{(i+0.5)}$

This approach assumes that the parameter $\nu_x = \frac{\Delta t \cos(\theta)}{\Delta x} = \frac{\mu_x}{|C|}$ is spatially uniform and therefore this pre-existing approach does not work in curvilinear coordinates, because at the propagation stage, ν must be treated as ν_i, as opposed to some combination of $\nu_{i-0.5}$ and $\nu_{i+0.5}$. The latter approach implies that the cell density update is using two inconsistent definitions of ν for the left and right cell boundaries, which naturally leads to violation of energy flux conservation. On the other hand, the value for *C* must be allowed to differ between the left and right cell boundaries; otherwise, the model will not be able to produce shoaling/de-shoaling due to spatial variations in depths and currents.

With the local rotation method, the grid cell rotation must be defined as uniform for any given cell. Most importantly, the left cell boundary and the right cell boundary cannot have two different rotations. The revised procedure is therefore:

1. Flux calculation stage: for all values of *i*,
   - calculate $\mu_x^{(i+0.5)}$ from pre-calculated variables: $\mu_x^{(i+0.5)} = 0.5(|C|^{(i)}\nu^{(i)} + |C|^{(i+1)}\nu^{(i+1)})$
   - calculate density at cell boundaries, e.g. $N_b^{(i+0.5)}$ using ULTIMATE QUICKEST scheme (unchanged from v3.13)
   - store flux-related variable for use in propagation stage: $(N_b|C|)^{(i+0.5)} = 0.5(|C|^{(i)} + |C|^{(i+1)})N_b^{(i+0.5)}$
2. Propagation stage: for all values of *i*,
   - update local density using fluxes at the two boundaries using cell-uniform value of ν: $N^{(i,t)} = N^{(i,t-1)} + \nu^{(i)}[(N_b|C|)^{(i-0.5)} - (N_b|C|)^{(i+0.5)}]$

One might imagine a scenario in which wave energy, traveling in a straight line through a curving grid cell, enters through the left boundary and then exits through some combination of the right and top (or bottom) boundary (Figure 5-1a). In fact, this does not occur, as will be explained now. As noted above, WW3 uses a split calculation for independent treatment of *x* and *y* propagation. Thus, for energy entering through the left (low-*x*) boundary of a cell produces three possible outcomes, either: a) it exits through the right (high-*x*) boundary, b) it produces a change to the density in the cell, or c) some combination of (a) and (b). It is not possible for some of this energy to exit through, for example, the top (high-*y*) boundary. This holds true for the model modified to use the local rotation method, since it also uses split calculations. Thus the individual cells of a curvilinear grid should be imagined as consisting of straight lines, even if the grid itself is severely curved (Figure 5-1b). Thus, it is easy to understand that a grid should be designed to be as orthogonal as possible, to reduce numerical error. In fact, this is probably true for any numerical model on a curvilinear grid.

Note that with the PHIDIAS approach, all propagation and fluxing is performed in "straightened space", so all cell boundaries have identical rotation, and this issue is moot.

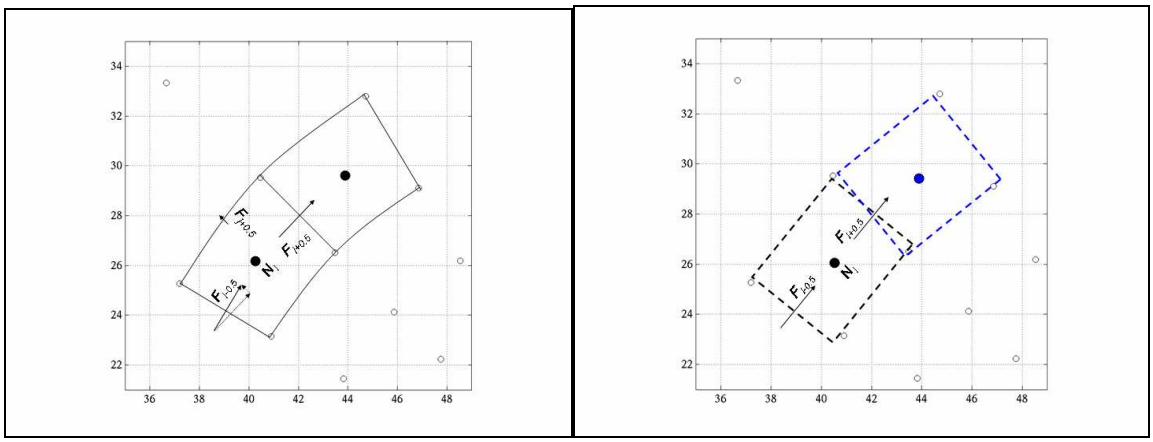

Figure 5-1 Schematic of curvilinear grid cells. Left panel (a): hypothetical redistribution of fluxes between *x* and *y* directions. Right panel (b): actual implementation (cell composed of straight lines, so no redistribution).

5.2. Results using local rotation method and contrast with PHIDIAS method

Results using the local rotation method are not presented here, for the simple reason that the results are practically indistinguishable from those using the PHIDIAS method. Thus, both methods have the modest defect of tending to square off the energy contours in cases of severely curving grids, for example, as seen in Figure 4-5. In fact, the local rotation method was the method first implemented, and experiments with the PHIDIAS method were originally considered as potential means to improve this defect. Of course, both methods have a number of positive characteristics also; these are summarized in Section 6.

With the similarity of results, the choice between the two methods must be made based on computational run times and code simplicity. The run times with the two methods indicate insignificant difference. The PHIDIAS method was chosen over the local rotation method because it requires fewer lines of code modified from v3.13. This is primarily due to the need to modify the flux/propagation routine when using the local rotation method, as described above, whereas with the PHIDIAS method, the original subroutine for flux/propagation can be applied without any modification.

6. Closing remarks

The extension of WW3 for propagation to curvilinear coordinate grid systems, in its present form, has the following negative characteristics:

1. When using the averaging technique for controlling the Garden Sprinkler Effect (GSE), the strength of the averaging is directly related to the local resolution. In some cases, this will lead to some minor inconsistencies in the correction (e.g. Figure 4-8). Addressing of this behavior should be fairly straightforward (Section 4.2.3), but would subsequently require additional testing.

2. When propagating wave energy features with strong horizontal curvature $\frac{\partial^2 E}{\partial \bar{x}^2}$, on a grid with severe curvature, with GSE corrections improperly disabled, some distortion of the energy contour will occur (e.g. see squaring of low-energy contours in Figure 4-5).

The extended code has these positive characteristics:

1. properly conserving energy flux,
2. matching results of v3.13 for rectilinear cases,
3. providing propagation that
   - is as accurate as the original model for mildly curving grids (e.g. Figure 4-11)
   - is much more accurate than an analogous implementation of the first order scheme on a rectilinear grid, even for severely curving grids (Figure 4-2 versus Figure 4-5)
   - has no visible defects, provided that GSE corrections are activated by the user, especially when proper correction strength is used (Figure 4-7), but also even when relatively weak correction strength is used (Figure 4-6) .

There are two other items left as potential further work:

1. The modifications to the code described herein, do not conflict with the pre-existing multi-grid feature described in Tolman (2007). This has been confirmed by running the NCEP-provided multi-grid test cases with the new code. However, those test cases utilize rectilinear grids, and the multi-grid code has not been extended to accommodate curvilinear grids.

2. As mentioned in the presentation of test case tp2.5c, there exists a singularity at the North Pole due to the specification of spectral direction components relative to the North Pole. The problem can be corrected by changing the model to allow the user to define an alternate reference point for defining spectral directions (the farther away, the better), but the method details are not obvious.

7. Acknowledgements

This work was supported by NOAA/NCEP EMC. We thank Dr. Hendrik Tolman (NCEP) for advice during the development process. PHIDIAS methodology was provided by Dr. Gerbrant van Vledder (TU Delft). COAMPS grid definitions were provided by Paul Wittmann (FNMOC). We gratefully acknowledge their contributions.

8. References

Booij, N., L.H. Holthuijsen, and R. Padilla-Hernandez, (1997). Numerical wave propagation on a curvilinear grid. *In the Proc. of the 3rd Int. Sym. on Ocean Wave Meas. and Analysis*, WAVES '97, ASCE, 286-294.

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

Booij, N., R.C. Ris, and L.H. Holthuijsen, 1999. A third-generation wave model for coastal regions, Part 1: Model description and validation. *J. Geophys. Res.* **104** (C4), 7649-7666.

Fletcher, C.A.J., 1991. Computational *Techniques for Fluid Dynamics*, Part II. Springer, 493 pp.

Hodur, R.M., 1997. The Naval Research Laboratory's Coupled Ocean/Atmospheric Mesoscale Prediction System (COAMPS). *Mon. Wea. Rev.*, **125**, 1414-1430.

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 Univ. Press, 532 pp.

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

Martin, P.J., 2000: *A Description of the Navy Coastal Ocean Model Version 1.0*. NRL Technical Report NRL/FR/7322-00-9962, 42 pp.

the SWAN team, 2007: SWAN Technical Documentation: SWAN Cycle III version 40.51A. Delft University of Technology. 99 pp.

Tolman, H.L., 2002a: *User Manual and System Documentation of WAVEWATCH-III Version 2.22.* NCEP Technical Note. 133 pp. [http://polar.ncep.noaa.gov/waves/wavewatch/wavewatch.html]

Tolman, H.L. 2002b: Alleviating the Garden Sprinkler Effect in wind wave models. *Ocean Modeling*, **4**, 269-289.

Tolman, H.L. 2007: Development of a multi-grid version of WAVEWATCH III. NCEP Technical Note. 112 pp.

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

WAMDI Group, 1988. The WAM model—A third generation ocean wave prediction model. *J. Phys. Oceanogr.*, **18**, 1775-1810.

Appendix: Modifications to user input method

In the pre-existing WW3, the user provides the bathymetry on the computational grid. This convention is retained in the curvilinear version. Thus, the user defines the curvilinear grid dimensions n_i and n_j, then reads in three matrixes, $x_{i,j}$, $y_{i,j}$, and the water depth $h_{i,j}$. Dissimilar grids for the input fields, such as winds, do not pose a problem; this is handled via the new grid search utility used in the field preprocessor executable as demonstrated in the test cases, e.g. tw3c.

The new model required changes to the way in which the user-created grid instruction file *ww3_grid.inp* is designed. As an example, for test case tp2.1b, the *ww3_grid.inp* file for the rectilinear case has the following lines where the input grid and depths are specified:

```
  'RECT' F F
  273  274
    1.6E+4   1.6E+4  1.
     0.E3    0.E3  1.
  -5. 5.75  10  -2500. 4 1 '(....)' 'UNIT' 'input'
   74802*1
$ etc.
```

The *ww3_grid.inp* file for the pseudo-curvilinear case has instead:

```
  'CURV' F F
  273  274
```