MODIFIED BOLTZMANN INTEGRAL FOR PRACTICAL WAVE MODELLING

by

Adhi Susilo

Submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

Major Subject: Engineering Mathematics

at

DALHOUSIE UNIVERSITY

Halifax, Nova Scotia May, 2007

Your file Votre référence
ISBN: 978-0-494-35795-8
Our file Notre référence
ISBN: 978-0-494-35795-8

DALHOUSIE UNIVERSITY

To comply with the Canadian Privacy Act the National Library of Canada has requested that the following pages be removed from this copy of the thesis:

Preliminary Pages
- Examiners Signature Page
- Dalhousie Library Copyright Agreement

Appendices
- Copyright Releases (if applicable)

O, God
Thy sea is so great
and my boat is so small

(The Breton Fisherman's Prayer)

Contents

List of Tables

List of Figures

Nomenclature

α	JONSWAP Phillips parameter
$\delta(\ldots)$	delta function
η	water surface elevation
γ	JONSWAP peak enhancement factor
λ	DIA parameter
λ	incremental factor
λ	longitude co-ordinate
ω_i	frequency
ϕ	latitude co-ordinate
ϕ	velocity potential
ρ	density or mass per unit volume
σ	angular frequency
σ	wave angular frequency
$\sigma_{a,b}$	JONSWAP peak width parameter
θ	directional spreading

ε wave phase

$\mathbf{C}_g$ group velocity, a vector

$\mathbf{k}_i$ wave number vector

$\mathbf{U}$ wind velocity, a vector

A wave amplitude

C coupling coefficient

C modified numerical constant

C phase velocity

C' numerical constant

C_g group velocity

d water depth

$D(f,\theta)$ directional spreading function

$D(N)$ density function

E total energy

$E(f,\theta)$ energy spectrum

F energy densities

F external force per unit mass

F fetch length

f frequency, Hz

$F(f)$ one-dimensional wave frequency spectrum

$F(f,\theta)$ directional wave frequency spectrum

$F(k,\theta)$ directional wave number spectrum

F^+ energy densities

F^- energy densities

F_d dominant factor

f_p peak frequency

g gravitational acceleration

$G(\mathbf{k})$ geometri function

H significant wave height

$H_{1/3}$ significant wave height

H_{rms} root-mean-square wave height

H_s significant wave height

k wave number

KE kinetic energy

L wave length

m mass

$N(k,\theta)$ directional action density spectrum

N_i action density

P pressure

PE potential energy

R finite depth scaling factor

R radius of the earth

$s(\sigma)$ spectral energy density

S_{bot} bottom interaction source term

S_{ds} dissipation source term

S_{inp} wind input source term

S_{nl4} nonlinear quadruplet interactions source term

S_{tot} total source term

T significant wave period

T wave period

t time

t wind duration

$T(\ldots)$ Webb's transfer function

$T_{1/3}$ significant wave period

U estimated wind velocity

u velocity component along the x-axis

w velocity component along the z-axis

x horizontal length co-ordinate

z vertical length co-ordinate

Acknowledgements

I would like to thank my supervisor Professor Matiur Rahman for his continuous support, guidance, and supervision throughout my research for the completion of my PhD degree in Engineering Mathematics Department at Dalhousie University.

I am gratefully indebted to my co-supervisor Dr Will Perrie of Fisheries and Oceans Canada, Bedford Institute for Oceanography (BIO) for his guidance, encouragement, and most of all financial support through a research grant from the Natural Science and Engineering Research Council Canada, and a great assistance for my shelter and other needs since I came to Canada. My sincere appreciation goes for his valuable criticism and advice, which has tremendously helped me in completing the work on this research. With this research I have also been introduced to a new interesting subject in the fluid dynamics field, the nonlinear wave-wave interactions.

I also would like to thank my thesis committee members, Professor W. Phillips and Professor G. Satish from Dalhousie University for their comments, corrections and suggestions on this thesis.

I especially thank my colleagues at Fisheries and Oceans Canada, Bedford Institute of Oceanography, especially for Bash Toulany, for helping me better understand wave models and the UNIX operating system and also Fumin Xu for granting me access to use their data.

I would like to dedicate many thanks to my parents for their moral support and their prayers that were sent from the other side of the world. Special thanks are also to my wife, Ariesta, for her understanding and her side-by-side support, since our wedding day. A lot of thanks are also dedicated to my little son, who gives me big

inspiration, spirit and joy.

Finally, I would like to thank to God, my strength and my refuge. Praise Him for his greatness. *'God is the one who began this work in you, and I am certain that he won't stop before it is complete'* (Phillippians 1:6).

Abstract

A fast accurate computation of nonlinear wave-wave interactions is needed for practical wave modeling. For all modern wave models such as WAM, SWAN and WAVEWATCH III, wave-wave interactions are computed using the Discrete Interaction Approximation (DIA). It is the centre of these models. While the DIA is fast, it is well know that it is not accurate. An accurate solution was proposed by Webb, Tracy and Resio (WRT) some years ago. Unfortunately, the WRT algorithm is still much slower than the DIA, and cannot be used operationally.

Based on Tracy and Resio's work, this thesis develops a new method which approximates the dominant contributions to the calculation of the total nonlinear wave-wave interactions. This method is denoted the Advanced Dominant Interaction (AvDI) method. This method saves computational time by choosing sets of interacting waves that account for the most important portion of the nonlinear wave transfer, so that it isn't necessary to integrate the contributions from the entire spectrum of possible wave-wave interactions. Taking account of the dominant transfer along wavenumbers and angles of interactions, the AvDI method can reduce the required computation time and space sufficiently that it can be implemented for practical wave modelling.

The AvDI method was implemented in an operational wave model and tested for ideal wave spectral cases and for a real storm. WAVEWATCH III wave model was chosen as a test-bed for implementation of the AvDI method. Fetch- and duration-limited growth tests were used in the ideal cases. Comparisons with waves observed during hurricane Juan constitute the real storm test. Compared to DIA, results show that AvDI method can be used as an alternative competitive method to compute the nonlinear wave-wave interactions.

Chapter 1

Introduction

1.1 Background

Waves are generated by the wind, formed as small wavelets due to variations of air pressure over the sea surface. The wind pushes against the sides of the wavelets, transferring energy more rapidly and causes the wave growth to accelerate. As the waves continue to grow, a roughened sea state is formed. Waves become higher and steeper until reaching a limiting point, after that individual waves break to form white caps. The 'fully-developed' limit to wave growth occurs approximately when the ratio of wave height to wave length (wave steepness) is 1 to 7. Waves continue to travel until reaching a boundary, or entering shallow water, where they are influenced by the bottom and modified, possibly with a loss of energy due to friction. If the wind direction changes, new waves are formed and travel in this new direction.

There are three factors that influence the generation of wind waves, i.e.

- wind velocity,
- fetch distance that wind blows across the water, and
- time or duration over which the wind blows.

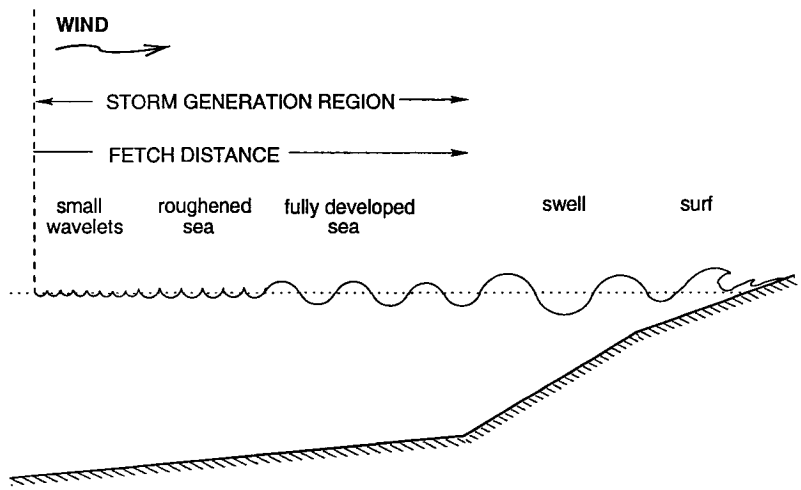


Figure 1.1: Wind wave growth (after Earle & Bishop, 1984).

Figure 1.1 depicts wave growth from ripples to full development, propagation as swell, dissipation and breaking on distant beaches as surf.

The correlation of wind speeds and ocean wave heights has been known for many years. It dates back at least as far as 1805, when Sir Francis Beaufort introduced a chart, the Beaufort scale, for this purpose. An interesting website which presents more information about the Beaufort scale can be found from Wikipedia contributors (2001). Initially the function of such a chart was to help sail boat captains to determine whether to add or take in sail, based on observations of the sea.

After sails were replaced with engines, people still used the Beaufort scale to estimate the sea conditions, for a given wind speed. In modern forecasting the Beaufort scale has been replaced, but it is still useful to provide an intuitive understanding of the ocean conditions in the absence of instrumental measurements. A modern version of the Beaufort scale is given in Table A.1, Appendix A. The table shows the relation between wind speed and wave height, as well as providing a visualization of matching sea surface conditions.

The modern investigation of ocean surface waves started with a pioneering study

by Sverdrup and Munk (1947). A discussion of their work also can be found in Kinsman (1965). Sverdrup and Munk were the first to develop the wave-forecast technique, expressed in terms of significant wave height. Their work and Bretschneider's work produced a wave model, commonly called the SMB (Sverdrup, Munk and Bretschneider) model. The SMB model predicts the significant wave height $H_{1/3}$ and significant wave period $T_{1/3}$ from known storm conditions, i.e. wind velocity U, fetch distance F and storm duration t. Predictions are made empirically using graphs of all the available data in terms of the dimensionless ratios gF/U^2, gt/U, gH/U^2 and gT/U, where g is the gravitational acceleration, U is the estimated wind velocity, F is the fetch length, t is the wind duration, T is the significant wave period and H is the significant wave height. A detailed formulation and discussion of this model is given by Komar (1976). However, since the SMB method only used local data, it was most effective for local forecasts.

More than half a century has passed since then and the study of ocean surface waves has greatly advanced. The current numerical wave models, supported by many fundamental studies, enable us to compute ocean surface waves on a global scale with sufficient accuracy for practical purposes. More details about the development of the study of ocean surface waves are presented by Komen et al. (1994) and Mitsuyasu (2002). The latter paper gives a summary of this development which can be seen in Table B.1, Appendix B. However, the physical processes controlling the energy balance of ocean surface waves are still not completely understood.

Modern wave modelling demands accurate results. An accurate computation of nonlinear wave-wave interactions is important. Good results are needed for accurate weather forecasting and offshore engineering projects in coastal waters of Canada's three oceans; Atlantic, Arctic and Pacific.

1.2 The Nonlinear Term

Modern wave models are often based on the energy balance equation. Wind-generated waves are described by the energy spectrum, $E(f,\theta)$, which obeys an energy conservation relation, whereby energy is input to the spectrum by wind and removed by wave-breaking dissipation,

$$\frac{dE(f,\theta)}{dt} = S_{tot} \, . \tag{1.1}$$

In deep water, S_{tot} can be represented as:

$$S_{tot} = S_{inp} + S_{ds} + S_{nl4} \tag{1.2}$$

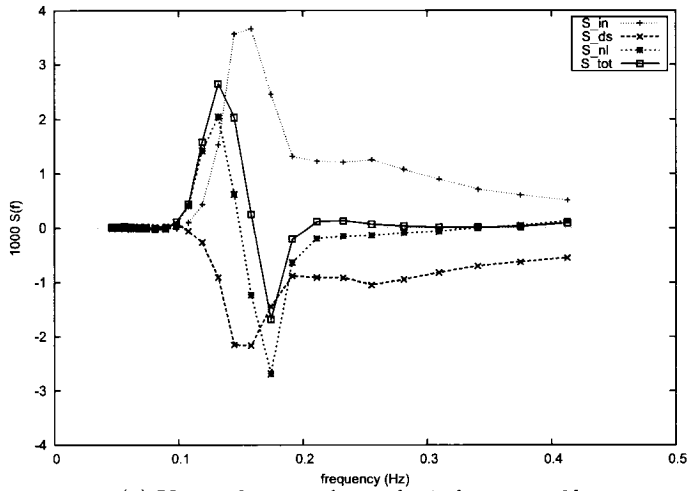
where S_{inp} is the energy input by wind, S_{ds} is the energy dissipation by white-capping and wave breaking, and S_{nl4} represents the nonlinear quadruplet wave-wave interactions.

The nonlinear interactions play an important role in the evolution of wind waves, representing a mechanism for shifting wave energy to lower and higher frequencies within the spectrum. Phillips (1960), using high order analysis, concluded that there are interactions between spectral components which cause transfer energy between the components. Figure 1.2 depicts the behaviour of nonlinear transfer for young and old seas.

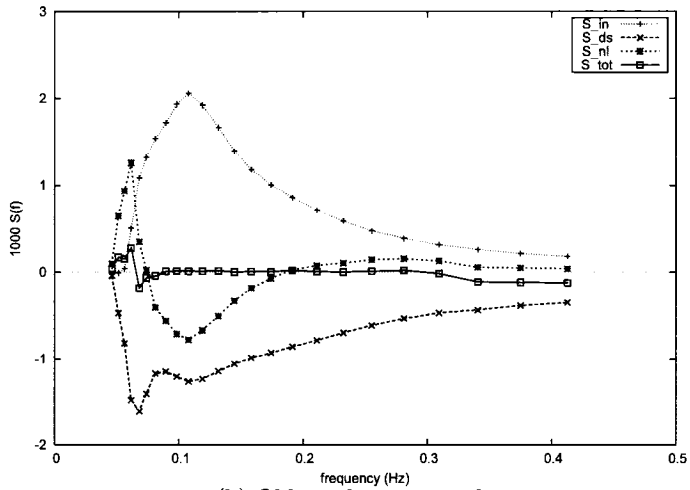
In shallow water, the interactions may involve three waves in triad interactions, by which two waves interact nonlinearly and transfer energy to a third component. Hasselmann (1962) reported that this kind of interaction is insignificant in deep water.

In deep water, nonlinear interactions describe the resonant exchange of energy, momentum, and action between four spectral components with wave number vectors $\mathbf{k}_1$, $\mathbf{k}_2$, $\mathbf{k}_3$, and $\mathbf{k}_4$ and corresponding frequencies ω_1, ω_2, ω_3, and ω_4 respectively. This set of four waves is called a quadruplet. This thesis will discuss this type of interaction only.

The basic equation describing S_{nl4} is the Boltzmann integral or kinetic equation,



(a) Young duration-limited wind sea, $t = 3h$.



(b) Old wind sea, $t = 96h$.

Figure 1.2: The energy balance at different sea states (after Komen et al., 1994).

proposed by Hasselmann (1962). He developed a perturbation method for the nonlinear resonant interaction of waves and found that energy exchange could happen when the resonant conditions are satisfied, i.e.

$$\mathbf{k}_1 + \mathbf{k}_2 = \mathbf{k}_3 + \mathbf{k}_4 \tag{1.3}$$

$$\omega_1 + \omega_2 = \omega_3 + \omega_4 \tag{1.4}$$

He concluded that for wind-generated seas, a dominant energy flux is expected from high to low wave-numbers in the initial period of development, when most of the energy is concentrated in the high wave-number region. As the peak migrates to lower wave-numbers, an increasing tendency for energy flux in the reverse direction develops, from low to high wave-numbers.

In 1978, Hasselmann's equation for nonlinear wave-wave interactions was studied by Webb numerically. He inferred that the integral consists of two terms, the pumping term and the diffusive term. Webb also introduced a transfer function $T(\mathbf{k}_1, \mathbf{k}_3)$ to simplify the triple integrals. However, the integral still required extensive in terms of required computing time.

Hasselmann and Hasselmann (1981) developed the pioneering systematic computation of S_{nl4}. They introduced symmetrical variables that avoid repeated calculations. They could speed up the computation relative to previous computations by the order of a factor 30-100.

Tracy and Resio (1982), continuing Webb's work, developed a simpler and more efficient integration process by utilization of a geometrically spaced polar grid over the spectral region. This special grid allows the loci and the coefficients inside the integrand to scale by various multiples of the geometric scaling factors. Only basic coefficients are computed. All other related computed values are products of the basic coefficients times scale factors.

Webb's transformation simplified the Hasselmann equation and Tracy and Resio's method increased computational efficiency significantly. The combination of these methods is called the WTR (*Webb Tracy-Resio*) method. Further review on this

method will be presented in Chapter 3.

As mentioned earlier a fast accurate technique for computing the nonlinear term is an important requirement for systematic studies of the nonlinear interactions or for operational applications. However, efficient computation of the nonlinear term is hampered by the complexity of the functional form and its computation is several orders of magnitude more expensive than all other terms in Eq.(1.2).

Hasselmann and Hasselmann (1985) used a symmetrical treatment, and continued to reduce the computing time. Hasselmann et al. (1985) developed four different parameterizations to approximate the exact solution of the nonlinear interaction expression. The *Discrete Interaction Approximation*, DIA, their fourth parameterization method, gives the best results. Therefore, they proposed the discrete-interaction parameterization to represent the full Boltzmann integral, based on two elementary interactions only. Chapter 3 will discuss this method in more detail.

The DIA, which dramatically increases the computational speed, enabled the development of third generation wave prediction models such as WAM and SWAN. However, DIA has a number of shortcomings (Van Vledder, 2000):

- DIA compares poorly with full integrations of S_{nl4} for many types of spectra.
- DIA's estimated spectral width is too large compared with measurements and full integrations of S_{nl4}.
- DIA produces too much transfer towards higher frequencies (Figure 3.6, Komen et al., 1984), which has impact on the tuning that has to be imposed on the source terms for wind input and dissipation.

Since DIA's development, improvements to the nonlinear term computation method have been attempted, consisting of three major approaches:

- more efficient calculation of the full S_{nl4} expression
- enhancement of DIA, and
- new approaches.

Resio and Perrie (1991) used selected scaling properties and symmetries of the nonlinear energy transfer integral to construct an integration grid to evaluate the Boltzmann complete integral. Lin and Perrie (1999) developed a formulation to represent the complete integral which is called the *Reduced Integration Approximation*, RIA. Their work is based on an analysis of the main resonance domain which reduces the six -dimensional integration to a quasi-line integral.

The previous paragraph shows examples of some works that have been done to compute the full integral in a more efficient manner. The next paragraph is going to show some examples of approaches which try to improve the DIA method.

The main idea of these approaches is to replace the full nonlinear integral S_{nl4} in the kinetic equation that describes a continuum of four-wave resonant interactions with a sum of a relatively small number of terms for a particular set of resonant quadruplets. Extending the DIA term with another elementary interaction term, Hashimoto et al. (2002) proposed the *Multiple Discrete Interaction Approximation*, MDIA. Polnikov and Farina (2002) found several simple configurations which are more efficient than the original DIA. They called their method the *Fast Discrete Interaction Approximation.* The original DIA has constant parameters throughout the spectrum. Tolman (2004) suggested that these parameters should be allowed to vary in spectral frequency f. This method is called the *Variable Discrete Interaction Approximation*, VDIA. It less accurate but cheaper than the MDIA.

A new approach was developed to improve the computational efficiency of the nonlinear term. This method solves the nonlinear interaction problem from a different point of view. Krasnopolsky, et.al. (2002) developed a neural network methodology to compute the nonlinear interactions. They stated that the nonlinear interactions constitute a continuous mapping from an initial input spectrum through an interaction process to an output spectrum. A neural network that is a generic and accurate tool for modelling complicated input-output relationship is trained using given inputs and outputs. Derived coefficients from the training are used as proper parameterizations for the mapping. This method can improve the performance by ten times and

higher, but representative training data must be constructed first.

A full six-dimensional integration is expensive for practical wave model forecasts. The present study attempts to speed up the calculation of the full S_{nl4} expression using the WTR formulation so it can be used for practical purposes. It is believed that the key to speeding up this integration is to reduce the number of integrations. As will be discussed in detail below, that the integral has two components; one component is a function of time and the other is not. This thesis suggests setting up the spectral grid first and keeps the value of the independent component at initial time for the next step in the computation. We then find a set of (k_1, k_3) that gives maximum transfer. The complete integral is replaced by representative integrals from the sets of contributions that produce maximum transfer, multiplied by an empirical scaling magnification factor.

This procedure will reduce the computation time needed for the whole integration. This method can be two orders of magnitude slower than DIA, but it still takes account of the loci of resonance. As an ultimate test, this formulation is installed in a wave model for practical storm wave simulations.

1.3 Objectives

The main objective of this thesis is to develop a new method to improve the computation of the nonlinear wave-wave interactions. Its purpose is to reduce the computation time while yielding reasonable results, so that the new formulation can be integrated into a wave model for practical storm wave simulations.

Secondly, a set of programs is built, based on the new method, and installed into an existing modern wave model (WAVEWATCH III). The new program becomes subprogram of the wave model and is modified to suit the main WAVEWATCH III program.

Thirdly, tests are conducted to document the performance of the new wave model, containing the new method for the nonlinear term. There are three test cases. The

first test concerns a defined input or 'snapshot' spectrum. This test must be done before installing the new program into the wave model. After this test is successful, the new program is ready to be installed in the wave model. The second test case is to implement the modified wave model into a square box sea, and perform fetch-limited growth experiments as prescribed in the SWAMP project. The third test case implements the new model for a real storm. We study hurricane Juan which made landfall in Halifax in September 2003, during the construction of this thesis, as a category 2 storm.

1.4 Overview

This thesis consists of seven chapters. The background of the project is presented in Chapter 1. Chapter 2 discusses wave theories in general, followed by discussion of nonlinear wave-wave interactions in Chapter 3, which also covers the basic methodologies of previous works on the nonlinear term. Chapter 4 is dedicated to presenting the main work of the thesis, i.e. development of the new method to compute the nonlinear transfer term. The next chapter, Chapter 5 discusses different kinds of wave models and the process of installation of the new method in a wave model. Results of the application of the new method for simulation of storm waves during hurricane Juan are presented in Chapter 6. Finally, conclusions and recommendation are presented in Chapter 7.

Chapter 2

Surface Wave Formulations

2.1 Linear Waves

In the ocean, waves have differing characteristics, from simple sinusoidal waves to complex C-choidal or Trochoidal shapes, with different amplitudes, frequencies, and directional distributions. These diversities create difficulties in the analysis of wave behaviours. Waves also break and loose energy due to friction and turbulence. To assist in the analysis of wave characteristics, we are going to analyse waves in their simplest form using a number of assumptions as follows:

- waves have small amplitude
- water is incompressible
- flow is irrotational and inviscid
- two-dimensional flow (the component of acceleration along the y-axis is assumed to be smaller than the other two components).

The theory that was developed using these assumptions is referred as small amplitude theory or linear theory or Airy wave theory.

We start with the continuity equation for incompressible two-dimensional flow, which is stated as

$$\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} = 0 \,. \tag{2.1}$$

For an irrotational flow, a velocity potential ϕ exists, that gives

$$u = -\frac{\partial \phi}{\partial x} \quad \text{and} \quad w = -\frac{\partial \phi}{\partial z} \,. \tag{2.2}$$

Substituting Eq.(2.2) into Eq.(2.1) yields

$$\triangle \phi = 0 \tag{2.3}$$

which is Laplace's equation, where

$$\triangle = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial z^2} \,. \tag{2.4}$$

Equation (2.3) is a partial differential equation which must be solved, subject to boundary conditions. Before we solve Eq.(2.3), we recall the equations of motion in the x and z directions, which are given as

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + w\frac{\partial u}{\partial z} = -\frac{1}{\rho}\frac{\partial P}{\partial x} + F_1(x) \tag{2.5}$$

$$\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + w\frac{\partial w}{\partial z} = -\frac{1}{\rho}\frac{\partial P}{\partial z} + F_2(z) \,. \tag{2.6}$$

Assuming that the only external force acting on that system is the gravitational force per unit mass, then we may write $F_1(x) = 0$ and $F_2(z) = -g$. The gravitational force is also derivable from a potential field, so we can write $-g = -\frac{\partial(gz)}{\partial z}$. With these assumptions and using Eq.(2.2), the equations of motion, Eq.(2.5) and Eq.(2.6), can

be stated as

$$-\frac{\partial^2\phi}{\partial t\partial x}+u\frac{\partial u}{\partial x}+w\frac{\partial u}{\partial z}=-\frac{1}{\rho}\frac{\partial P}{\partial x} \tag{2.7}$$

$$-\frac{\partial^2\phi}{\partial t\partial z}+u\frac{\partial w}{\partial x}+w\frac{\partial w}{\partial z}=-\frac{1}{\rho}\frac{\partial P}{\partial z}-\frac{\partial(gz)}{\partial z}\,. \tag{2.8}$$

Integrating Eq.(2.7) and Eq.(2.8) with respect to x and z respectively yields

$$-\frac{\partial\phi}{\partial t}+\frac{1}{2}(u^2+w^2)+\frac{P}{\rho}=Q_1(z,t) \tag{2.9}$$

$$-\frac{\partial\phi}{\partial t}+\frac{1}{2}(u^2+w^2)+\frac{P}{\rho}+gz=Q_2(x,t)\,. \tag{2.10}$$

Subtracting Eq.(2.9) from Eq.(2.10) gives

$$gz=Q_2(x,t)-Q_1(z,t) \tag{2.11}$$

which suggests that Q_2 is a function of t alone. Thus $Q_2=Q_2(t)$ and hence $Q_1(z,t)=Q_2(t)-gz$. Therefore, Eq.(2.9) and Eq.(2.10) reduce to the single equation

$$-\frac{\partial\phi}{\partial t}+\frac{1}{2}(u^2+w^2)+\frac{P}{\rho}+gz=Q_2(t) \tag{2.12}$$

which is Bernoulli's equation in two-dimensional flow.

For the steady state case, $\frac{\partial\phi}{\partial t}=0$, Eq.(2.12) reduces to

$$\frac{1}{2}(u^2+w^2)+\frac{P}{\rho}+gz=\text{constant}\,. \tag{2.13}$$

Without loss of generality, $Q_2(t)$ can be combined with the velocity potential $\phi(x,z,t)$, so that Eq.(2.13) becomes

$$-\frac{\partial\phi}{\partial t}+\frac{1}{2}(u^2+w^2)+\frac{P}{\rho}+gz=0\,. \tag{2.14}$$

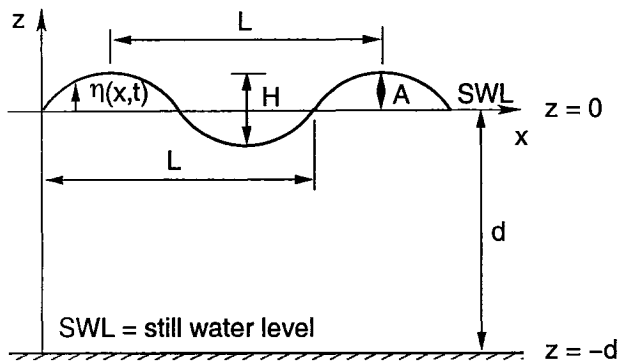


Figure 2.1: Schematic diagram for linear wave theory.

The small amplitude wave theory is based upon the further assumption that all motions are infinitely small, which enables us to neglect the square of the velocity components, and yields

$$-\frac{\partial \phi}{\partial t} + \frac{P}{\rho} + gz = 0 \tag{2.15}$$

which is the general equation of motion applied in the development of the small amplitude theory of water waves.

The Laplace's equation, Eq.(2.3), is the differential equation for water waves. Variables and coordinates are defined in Fig. 2.1. The equation is to be satisfied in the region $-d \leq z \leq \eta$ and $-\infty < x < \infty$ subject to the boundary conditions

$$w = -\frac{\partial \phi}{\partial z} = 0 \quad \text{on} \quad z = -d \qquad \text{and} \tag{2.16}$$

$$\eta = \frac{1}{g}\left(\frac{\partial \phi}{\partial t}\right) \quad \text{on} \quad z = 0\,. \tag{2.17}$$

The method of separation of variables is a powerful tool for solving Laplace's equation. More details can be found in Rahman (1991). An alternative solution is obtained by assuming a progressive wave solution of the form $\phi \propto e^{i(kx-\sigma t)}$. Thus we

may write

$$\phi = Z(z)R_e\{e^{i(kx-\sigma t)}\} \tag{2.18}$$

where R_e represents the real part of the complex solution. Therefore, we can rewrite Eq.(2.18) as

$$\phi = Z(z)\cos(kx - \sigma t)\ . \tag{2.19}$$

Applying this solution in Laplace's equation, we get

$$Z'' - k^2 Z = 0 \tag{2.20}$$

which has the solution,

$$Z = Ae^{kz} + Be^{-kz}\ . \tag{2.21}$$

Therefore

$$\phi = (Ae^{kz} + Be^{-kz})\cos(kx - \sigma t). \tag{2.22}$$

The boundary conditions to be satisfied by Eq.(2.22) are Eq.(2.16) and Eq.(2.17). Using Eq.(2.16), we obtain

$$\begin{aligned} Ae^{-kd} + Be^{kd} &= 0 \\ A &= Be^{2kd}. \end{aligned}$$

Then Eq.(2.22) becomes

$$\begin{aligned} \phi &= (Be^{2kd}e^{kz} + Be^{-kz})\cos(kx - \sigma t) \\ &= Be^{kd}\{e^{k(d+z)} + e^{-k(d+z)}\}\cos(kx - \sigma t)\ . \end{aligned} \tag{2.23}$$

Applying Eq.(2.17), we get

$$\eta = \frac{B\sigma}{g}e^{kd}\{e^{kd} + e^{-kd}\}\sin(kx - \sigma t)\ . \tag{2.24}$$

Defining

$$A = \frac{B\sigma}{g} e^{kd}(e^{kd} + e^{-kd}) \tag{2.25}$$

where A is the amplitude of the wave, then

$$\eta = A\sin(kx - \sigma t)\ . \tag{2.26}$$

Consequently, we can write Eq.(2.23) as

$$\begin{aligned}\phi &= \frac{Ag}{\sigma}\frac{e^{k(d+z)} + e^{-k(d+z)}}{e^{kd} + e^{-kd}}\cos(kx - \sigma t) \\ &= \frac{Ag}{\sigma}\frac{\cosh k(d+z)}{\cosh kd}\cos(kx - \sigma t).\end{aligned} \tag{2.27}$$

With this velocity potential, the associated physical quantities can be written,

- wave elevation

$$\begin{aligned}\eta &= \frac{1}{g}\left(\frac{\partial\phi}{\partial t}\right)_{z=0} \\ &= A\sin(kx - \sigma t)\end{aligned} \tag{2.28}$$

- horizontal velocity component

$$\begin{aligned}u &= -\frac{\partial\phi}{\partial x} \\ &= \frac{Agk}{\sigma}\frac{\cosh k(d+z)}{\cosh kd}\sin(kx - \sigma t)\end{aligned} \tag{2.29}$$

- vertical velocity component

$$\begin{aligned}w &= -\frac{\partial\phi}{\partial z} \\ &= -\frac{Agk}{\sigma}\frac{\sinh k(d+z)}{\cosh kd}\cos(kx - \sigma t)\end{aligned} \tag{2.30}$$

- pressure

$$P = \rho\frac{\partial\phi}{\partial t} - \rho gz$$
$$= \rho Ag\frac{\cosh k(d+z)}{\cosh kd}\sin(kx - \sigma t) - \rho gz. \tag{2.31}$$

A descriptive book about waves, with minimum mathematical description, can be found in Bascom (1964).

2.2 Group Velocity

If we consider two waves moving in the positive x-direction, having equal amplitudes and phase, thus

$$\eta_T = A\sin(k_1x - \sigma_1t) + A\sin(k_2x - \sigma_2t) \tag{2.32}$$
$$= 2A\cos[\frac{1}{2}(k_1 - k_2)x - \frac{1}{2}(\sigma_1 - \sigma_2)t]$$
$$\times \sin[\frac{1}{2}(k_1 + k_2)x - \frac{1}{2}(\sigma_1 + \sigma_2)t]. \tag{2.33}$$

To determine nodal points which are located by finding the zeros of the cosine factor in Eq.(2.33), we may write,

$$\cos\theta = 0$$
$$\frac{1}{2}(k_1 - k_2)x - \frac{1}{2}(\sigma_1 - \sigma_2)t = (2n+1)\frac{\pi}{2} \qquad n = 0, 1, 2, 3, \ldots \tag{2.34}$$

Thus

$$x_{node} = \left(\frac{\sigma_1 - \sigma_2}{k_1 - k_2}\right)t + \frac{(2n+1)\pi}{k_1 - k_2}. \tag{2.35}$$

As can be seen, the position of all nodes is not stationary, rather it is function of time. At $t = 0$ there will be nodes at

$$x_{node} = \frac{(2n+1)\pi}{k_1 - k_2} \qquad n = 0, 1, 2, 3, \ldots$$

The speed of propagation of these nodes is called the group velocity, which is given by

$$C_g = \frac{d}{dt}(x_{node}) = \frac{\sigma_1 - \sigma_2}{k_1 - k_2}. \tag{2.36}$$

As σ_1 approaches σ_2, we can rewrite Eq.(2.36) as $C_g = \frac{d\sigma}{dk}$.

In terms of wave phase velocity C, C_g can be stated as

$$\begin{aligned}
C_g &= \frac{d(kC)}{dk} \\
&= C + k\frac{dC}{dk} \qquad \text{and} \\
C &= \sqrt{\frac{g}{k}}(\tanh kd)^{1/2} \\
\frac{dC}{dk} &= \sqrt{g}(-\frac{1}{2})k^{-3/2}(\tanh kd)^{1/2} \\
&\quad + \sqrt{\frac{g}{k}}(\tanh kd)^{-1/2}\, d\, \frac{1}{2}\text{sech}^{\,2} kd \\
k\frac{dC}{dk} &= -\frac{1}{2}\sqrt{\frac{g}{k}}(\tanh kd)^{1/2} + \frac{1}{2}\sqrt{\frac{g}{k}}(\tanh kd)^{1/2} kd \frac{\text{sech}^{\,2} kd}{\tanh kd} \\
&= -\frac{1}{2}C + \frac{C}{2} kd \frac{1}{\sinh kd\, \cosh kd} \\
&= -\frac{1}{2}C + \frac{C}{2}\frac{2kd}{\sinh 2kd} \qquad \text{therefore} \\
C_g &= C + k\frac{dC}{dk} \\
&= C - \frac{1}{2}C + \frac{C}{2}\frac{2kd}{\sinh 2kd} \\
&= \frac{C}{2}\left(1 + \frac{2kd}{\sinh 2kd}\right).
\end{aligned} \tag{2.37}$$

For shallow water, $kd \to 0$ and $Cg \approx C$. For deep water $kd \to \infty$ and $Cg \approx \frac{1}{2}C$.

2.3 Wave Energy

The total energy of a progressive wave is the summation of its potential energy and kinetic energy. The potential energy is the height of the column of water when a wave form is present minus the state when there is no wave. We shall first compute the potential energy when there is a wave form, PE_1. The potential energy of a column of water with $d+\eta$ height, dx length, and one unit width is

$$\begin{aligned} \Delta(PE_1) &= (\text{height to centre of gravity}) \times g\Delta m \\ d(PE_1) &= \left(\frac{d+\eta}{2}\right) g\rho(d+\eta)\, dx \\ &= \frac{(d+\eta)^2}{2}\rho g\, dx. \end{aligned} \tag{2.38}$$

Using $\eta = A\sin(kx - \sigma t)$, the average potential energy per unit surface area can be denoted

$$\begin{aligned} PE_1 &= \frac{\rho g}{2LT}\int_t^{t+T}\int_x^{x+L}(d+\eta)^2\, dxdt \\ &= \frac{\rho g d^2}{2} + \frac{\rho g A^2}{4}. \end{aligned} \tag{2.39}$$

The potential energy in the absence of a wave can be stated as

$$\begin{aligned} PE_2 &= \frac{\rho g}{2LT}\int_t^{t+T}\int_x^{x+L} d^2\, dxdt \\ &= \frac{\rho g d^2}{2}. \end{aligned} \tag{2.40}$$

Thus, the average of total potential energy is

$$\begin{aligned} PE &= PE_1 - PE_2 \\ &= \frac{\rho g A^2}{4}. \end{aligned} \tag{2.41}$$

The kinetic energy of a small element of water with dz height, dx length, and one unit width, and with velocity components u and w is

$$\begin{aligned} d(KE) &= \frac{1}{2}(u^2 + w^2)\, dm \\ &= \frac{1}{2}(u^2 + w^2)\rho\, dzdx. \end{aligned} \tag{2.42}$$

Thus, the average of kinetic energy is

$$KE = \frac{\rho}{2LT} \int_t^{t+T} \int_x^{x+L} \int_{-d}^{\eta=0} (u^2 + w^2)\rho\, dzdxdt. \tag{2.43}$$

Using the velocity components compatible with the progressive wave $\eta = A\sin(kx - \sigma t)$, we get

$$KE = \frac{\rho g A^2}{4}. \tag{2.44}$$

Therefore, the total energy is given by

$$E = PE + KE = \frac{\rho g A^2}{2}. \tag{2.45}$$

2.4 Dispersion Relation

For small amplitude waves, the vertical velocity component w is equal to the rate of change of the water surface at any point. Thus

$$w = \frac{d\eta}{dt} = \frac{\partial \eta}{\partial t} + u\frac{\partial \eta}{\partial x}.$$

Neglecting the second order term on the right, we get

$$w \approx \frac{\partial \eta}{\partial t} \qquad \text{on} \qquad z = 0.$$

Since $w = -\frac{\partial \phi}{\partial z}$, hence

$$\begin{aligned} \frac{\partial \eta}{\partial t} &= -\frac{\partial \phi}{\partial z} \\ \frac{\partial \eta}{\partial t} + \frac{\partial \phi}{\partial z} &= 0\,. \end{aligned} \tag{2.46}$$

Since the dynamic boundary condition is given by

$$\eta = \frac{1}{g}\left(\frac{\partial \phi}{\partial t}\right)_{z=0} \tag{2.47}$$

then we can rewrite Eq.(2.46) as

$$\frac{\partial^2 \phi}{\partial t^2} + g\frac{\partial \phi}{\partial z} = 0 \qquad \text{at} \qquad z = 0. \tag{2.48}$$

Considering a progressive wave with a velocity potential given by

$$\phi = \frac{Ag}{\sigma}\frac{\cosh k(d+z)}{\cosh kd}\cos(kx - \sigma t)$$

we obtain,

$$\frac{\partial^2\phi}{\partial t^2} = -Ag\sigma\frac{\cosh k(d+z)}{\cosh kd}\cos(kx-\sigma t) \tag{2.49}$$

$$g\frac{\partial\phi}{\partial z} = \frac{Ag^2k}{\sigma}\frac{\sinh k(d+z)}{\cosh kd}\cos(kx-\sigma t). \tag{2.50}$$

Substituting these values into Eq.(2.48) at $z = 0$, yields

$$\sigma^2 = gk\tanh(kd). \tag{2.51}$$

This relation is well known as the dispersion relation. Since $\sigma = kC$, Eq.(2.51) may be written as

$$C^2 = \frac{g}{k}\tanh(kd). \tag{2.52}$$

With $\sigma = \frac{2\pi}{T}$ and $k = \frac{2\pi}{L}$, the dispersion relation may be rearranged to give

$$L = \frac{gT^2}{2\pi}\tanh(\frac{2\pi h}{L}). \tag{2.53}$$

2.5 Wave Height

We have discussed regular waves whose properties are the same from one cycle to the next. In the real ocean waves are not regular, in the sense that the ocean surface is composed of waves moving in different directions, with different amplitudes, frequencies and phases. In characterizing ocean waves, wave height is important.

There are several wave height definitions, but $H_{1/3}$ (or H_s), significant wave height and, H_{max}, maximum wave height are the most popular wave height definitions. If there are N wave heights measured at a certain point, and we arrange them from the largest to the smallest, assigning numbers 1 to N, then $H_{1/3}$ or H_s is defined as the

average of the first $N/3$ highest waves. Mathematically this can be stated as

$$H_s = H_{1/3} = \frac{3}{N}\sum_{i=1}^{N/3} H_i. \tag{2.54}$$

The root-mean-square (rms) wave height is defined as

$$H_{rms} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} H_i^2}. \tag{2.55}$$

The relation between H_{max} and H_{rms} is given by (Longuet-Higgins, 1952)

$$H_{max} = \left[\sqrt{\ln N} + \frac{0.2886}{\sqrt{\ln N}}\right]. \tag{2.56}$$

2.6 Wave Spectra

A pure sinusoidal wave (Figure 2.2a) can be stated as

$$\eta = A\cos(\sigma t - kx) \qquad \text{or} \qquad \eta = A\cos(\sigma t - \varepsilon)\,. \tag{2.57}$$

The actual ocean consists of many sinusoidal waves with many frequencies σ_n, amplitudes A_n, and phases ε_n. Superimposing all these waves gives

$$\eta = \sum_{n=0}^{\infty} A_n \cos(\sigma_n t - \varepsilon_n) \tag{2.58}$$

This represents a random wave, and its plot can be seen in Fig. 2.2b. If the wave is composed of the sum of sines and cosines and has a fundamental period T, it can be expressed as

$$\eta = \sum_{n=0}^{\infty}(a_n \cos n\sigma t + b_n \sin n\sigma t) \tag{2.59}$$

where $\sigma = 2\pi/T$. This wave is called an irregular wave, as shown on Fig. 2.2c.

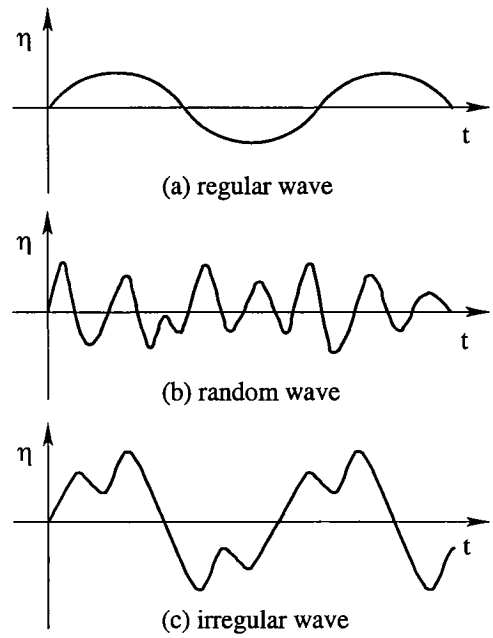


Figure 2.2: Sample of various wave profiles as functions of time (after Rahman, 1995).

Plotting η against t is not really helpful because knowing η at any time is not necessary. It is more useful to plot the amplitude A_n against σ_n. This plot is called a discrete amplitude spectrum (Fig. 2.3a). However, if $n \rightarrow \infty$, a continuous amplitude spectrum is achieved, as shown on Fig. 2.3b. Another spectrum is found if we plot $A_n^2/2$ against σ_n, which results in a discrete energy spectrum (Fig. 2.3c). When $n \rightarrow \infty$, a continuous energy spectrum yields Fig. 2.3d. This curve is commonly called the energy density spectrum, because the area under this curve represents the total energy of the wave. Figure 2.4 illustrates wave spectra which are associated with types of surface waves, the nature of their displacing forces, wavelengths, frequencies, and the relative amounts of energy. This kind of graph can be used to help harvesting energy from the sea. A brief discussion about wave energy conversion can be found in Brooke (2003).

Fourier series can be defined if $\eta(t)$ is a continuous function defined over $d \leq t \leq$

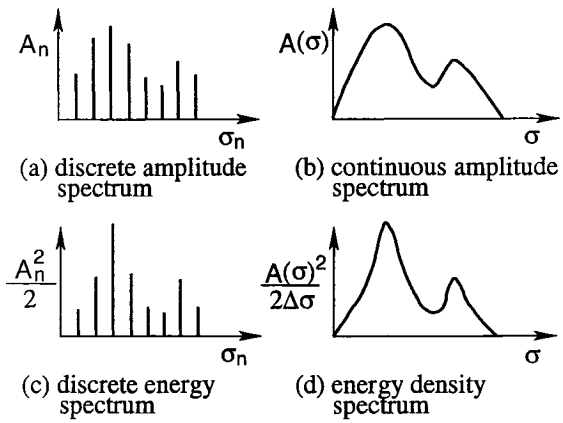


Figure 2.3: Plots of various spectra (after Rahman, 1995).

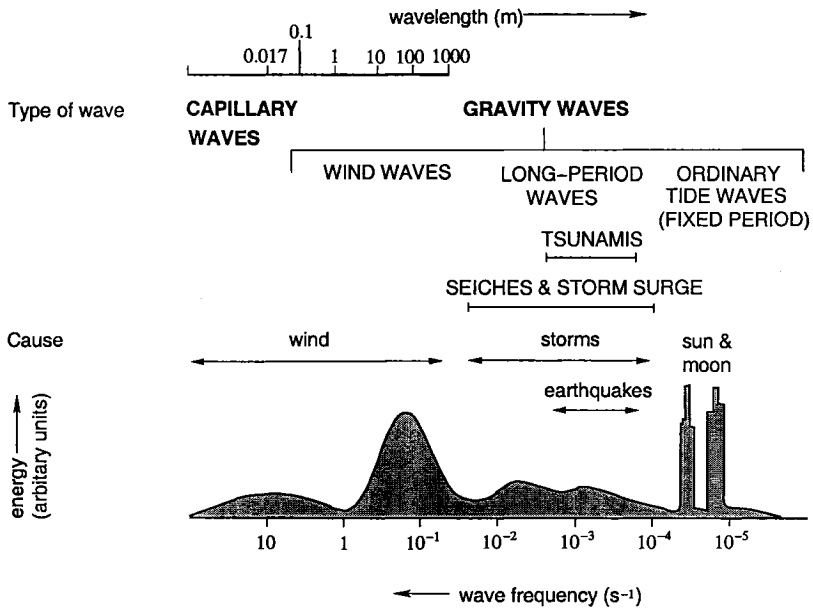


Figure 2.4: Wave spectra and types of surface waves (after Brown, et al., 1989).

$d + 2p$ where $2p$ is the period of function $\eta(t)$. Then $\eta(t)$ can be expressed as

$$\eta(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \cos \frac{n\pi t}{p} + b_n \sin \frac{n\pi t}{p} \right) \tag{2.60}$$

where

$$a_n = \frac{1}{p} \int_d^{d+2p} \eta(t) \cos \frac{n\pi t}{p} \, dt \quad n = 0, 1, 2, \ldots \tag{2.61}$$

$$b_n = \frac{1}{p} \int_d^{d+2p} \eta(t) \sin \frac{n\pi t}{p} \, dt \quad n = 1, 2, 3, \ldots \tag{2.62}$$

If the Fourier series contains a finite number of components, the infinite series can be replaced by N number.

$$\langle E \rangle = \frac{1}{2} \rho g \langle \eta^2(t) \rangle \tag{2.63}$$

where $\langle \eta^2(t) \rangle$ is the mean square value of $\eta(t)$ over a wave cycle, which can be written as

$$\begin{aligned}
\langle \eta^2(t) \rangle &= \frac{1}{2p} \int_d^{d+2p} \eta(t) \, \eta(t) \, dt \\
&= \frac{1}{2p} \int_d^{d+2p} \eta(t) \left[\frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \cos \frac{n\pi t}{p} + b_n \sin \frac{n\pi t}{p} \right) \right] dt \\
&= \frac{a_0}{2} \left(\frac{1}{2p} \int_d^{d+2p} \eta(t) \, dt \right) \\
&\quad + \sum_{n=1}^{\infty} a_n \left(\frac{1}{2p} \int_d^{d+2p} \eta(t) \cos \frac{n\pi t}{p} \, dt \right) \\
&\quad + \sum_{n=1}^{\infty} b_n \left(\frac{1}{2p} \int_d^{d+2p} \eta(t) \sin \frac{n\pi t}{p} \, dt \right).
\end{aligned}$$

Substituting Eq.(2.61) and (2.62) in the above expression, we get

$$\langle \eta^2(t) \rangle = \frac{a_0^2}{4} + \frac{1}{2} \sum_{n=1}^{\infty} (a_n^2 + b_n^2) \, . \tag{2.64}$$

This relation is known as Parseval's identity. Then, Eq.(2.63) can be written as

$$\langle E \rangle = \frac{1}{2}\rho g \left[\frac{a_0^2}{4} + \frac{1}{2}\sum_{n=1}^{\infty}(a_n^2 + b_n^2) \right] . \tag{2.65}$$

The periodic elevation function of period $2p$ can be defined in complex Fourier series as (see Rahman 1995 for detail),

$$\begin{aligned} \eta_p(t) &= \sum_{n=-\infty}^{\infty} c_n e^{in\pi t/p} \qquad \text{and} \\ c_n &= \frac{1}{2p}\int_{-p}^{p} \eta_p(t) e^{-in\pi t/p}\, dt \\ &= \frac{1}{2p}\int_{-p}^{p} \eta_p(\tau) e^{-in\pi\tau/p}\, d\tau . \end{aligned}$$

Substituting the second expression for c_n into $\eta_p(t)$ gives

$$\begin{aligned} \eta_p(t) &= \sum_{n=-\infty}^{\infty} \left(\frac{1}{2p}\int_{-p}^{p} \eta_p(\tau) e^{-in\pi\tau/p}\, d\tau \right) e^{in\pi t/p} \qquad \text{or} \\ &= \sum_{n=-\infty}^{\infty} \left(\frac{1}{2\pi}\int_{-p}^{p} \eta_p(\tau) e^{-in\pi\tau/p}\, d\tau \right) e^{in\pi t/p} \left(\frac{\pi}{p} \right) . \end{aligned} \tag{2.66}$$

Furthermore, by defining, $\sigma_n = n\pi/p$, $\sigma_{n+1} = (n+1)\pi/p$, and $\Delta\sigma = \sigma_{n+1} - \sigma_n = \pi/p$, we can rewrite Eq.(2.66) with these definitions

$$\eta_p(t) = \sum_{n=-\infty}^{\infty} \left(\frac{1}{2\pi}\int_{-p}^{p} \eta_p(\tau) e^{-i\sigma_n\tau}\, d\tau \right) e^{i\sigma_n t}\, \Delta\sigma .$$

As p goes to infinity, the periodic function $\eta_p(t)$ becomes $\eta(t)$ and $\Delta\sigma \to 0$. Then the

previous expression asymptotes to the continuous random wave elevations integral

$$\eta(t) = \int_{-\infty}^{\infty} \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} \eta(\tau) \mathrm{e}^{-\mathrm{i}\sigma\tau}\, d\tau \right) \mathrm{e}^{\mathrm{i}\sigma t}\, d\sigma \tag{2.67}$$

$$= \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \eta(\tau) \mathrm{e}^{-\mathrm{i}\sigma(\tau - t)}\, d\tau\, d\sigma\ . \tag{2.68}$$

In terms of cosine and sine, remembering the fact that $\int_{-\infty}^{\infty} \sin\sigma(\tau - t)\, d\tau = 0$ and $\int_{-\infty}^{\infty} \cos\sigma(\tau - t)\, d\tau = 2\int_{0}^{\infty} \cos\sigma(\tau - t)\, d\tau$, therefore we infer that

$$\eta(t) = \frac{1}{\pi} \int_{0}^{\infty} \int_{-\infty}^{\infty} \eta(\tau) \cos\sigma(\tau - t)\, d\tau\, d\sigma$$

or

$$\eta(t) = \frac{1}{\pi} \int_{0}^{\infty} [a(\sigma)\cos\sigma t + b(\sigma)\sin\sigma t]\, d\sigma \tag{2.69}$$

where

$$a(\sigma) = \int_{-\infty}^{\infty} \eta(\tau)\cos\sigma\tau\, d\tau \qquad \text{and} \qquad b(\sigma) \int_{-\infty}^{\infty} \eta(\tau)\sin\sigma\tau\, d\tau\ .$$

The total energy of a wave per unit surface area, E, in the wave record between infinite time limits is given by

$$E = \frac{1}{2}\rho g \int_{-\infty}^{\infty} \eta^2(t)\, dt. \tag{2.70}$$

Substituting Eq.(2.69) into Eq.(2.70) gives

$$\begin{aligned} E &= \frac{1}{2}\rho g \int_{-\infty}^{\infty} \eta(t)\, \eta(t)\, dt \\ &= \frac{1}{2}\rho g \int_{-\infty}^{\infty} \eta(t) \left[\frac{1}{\pi}\int_0^{\infty} [a(\sigma)\cos\sigma t + b(\sigma)\sin\sigma t]\, d\sigma\right] dt \\ &= \frac{\rho g}{2\pi}\int_0^{\infty}\left[a(\sigma)\int_{-\infty}^{\infty}\eta(t)\cos\sigma t\, dt + b(\sigma)\int_{-\infty}^{\infty}\eta(t)\sin\sigma t\, dt\right] d\sigma \\ &= \frac{\rho g}{2\pi}\int_0^{\infty}[a^2(\sigma)+b^2(\sigma)]\, d\sigma \\ &= \frac{\rho g}{2\pi}\int_0^{\infty} A^2(\sigma)\, d\sigma \end{aligned} \tag{2.71}$$

where $A^2(\sigma) = a^2(\sigma) + b^2(\sigma)$. Comparing Eq.(2.70) and Eq.(2.71), we find that

$$\int_{-\infty}^{\infty}\eta^2(t)\, dt = \frac{1}{\pi}\int_0^{\infty} A^2(\sigma)\, d\sigma \tag{2.72}$$

which is Parseval's identity for a continuous nonperiodic random wave function $\eta(t)$. This relation shows the concept of wave energy spectrum.

If $\langle\eta^2(t)\rangle$ is the mean square value of $\eta(t)$ over a specified record length T then

$$\langle\eta^2(t)\rangle = \lim_{T\to\infty}\left[\frac{1}{T}\int_{-T/2}^{T/2}\eta^2(t)\, dt\right] . \tag{2.73}$$

The mean total energy can be stated as

$$\begin{aligned}
\langle E \rangle &= \frac{1}{2}\rho g \lim_{T\to\infty}\left[\frac{1}{T}\int_{-T/2}^{T/2}\eta^2(t)\,dt\right] \\
&= \lim_{T\to\infty}\frac{\rho g}{2T}\int_{-T/2}^{T/2}\eta(t)\,\eta(t)\,dt \\
&= \lim_{T\to\infty}\frac{\rho g}{2T}\int_{-T/2}^{T/2}\eta(t)\left[\frac{1}{\pi}\int_0^\infty [a(\sigma)\cos\sigma t + b(\sigma)\sin\sigma t]\,d\sigma\right]dt \\
&= \lim_{T\to\infty}\frac{\rho g}{2\pi T}\int_0^\infty\left[a(\sigma)\int_{-T/2}^{T/2}\eta(t)\cos\sigma t\,dt\right. \\
&\qquad\left. +b(\sigma)\int_{-T/2}^{T/2}\eta(t)\sin\sigma t\,dt\right]d\sigma\,.
\end{aligned} \tag{2.74}$$

Considering that T is large but finite and integrals $\int_{-T/2}^{T/2}\eta(t)\cos\sigma t\,dt$ and $\int_{-T/2}^{T/2}\eta(t)\sin\sigma t\,dt$ are approximated as $a(\sigma)$ and $b(\sigma)$ respectively, then

$$\begin{aligned}
\langle E \rangle &= \frac{\rho g}{2\pi T}\int_0^\infty [a^2(\sigma)+b^2(\sigma)]d\sigma \\
&= \frac{\rho g}{2}\int_0^\infty \frac{A^2(\sigma)}{\pi T}d\sigma \\
&= \frac{\rho g}{2}\int_0^\infty s(\sigma)\,d\sigma
\end{aligned} \tag{2.75}$$

where $s(\sigma) = \frac{A^2(\sigma)}{\pi T}$, which is the spectral energy density as a function of frequency. The total energy is obtained for the area under that curve.

There are two commonly known methods, i.e. the autocorrelation method and the Fast Fourier Transform (FFT) method, for calculating the spectral energy density. Further discussion of those methods is given by Rahman (1995). Several mathematical spectrum models have also been proposed. Two well known models are the Pierson-Moskowitz (1964) spectrum and the Joint North Sea Wave Project (JONSWAP) spectrum, by Hasselmann et al. (1973). In common form, the JONSWAP spectrum

is stated as

$$F(f) = \underbrace{\alpha g^2 (2\pi)^{-4} f^{-5} \exp\left[-\frac{5}{4}\left(\frac{f}{f_p}\right)^{-4}\right]}_{\text{Pierson-Moskowitz spectrum}} \gamma^{\exp\left[-\frac{(f-f_p)^2}{2\sigma^2 f_p^2}\right]} \tag{2.76}$$

where α, γ, and σ are JONSWAP shape parameters; $\sigma = \sigma_a$ for $f < f_p$ and $\sigma = \sigma_b$ for $f \geq f_p$; and f_p is the peak frequency.

2.6.1 Directional wave spectra

Wave energy at a point has an angular distribution as well as a frequency distribution. This angular distribution is known as the *directional spreading*. The spectral energy density which represents both frequency distribution and angular spreading is called the *directional spectra*. The directional spectra is needed in order to give wave predictions. Collins et al. (1981) suggested that a model without directional spectra overpredicts significant height by 20 percent during refraction.

The directional spectra can be commonly parameterized as the product of the one-dimensional sprectrum times a directional spreading factor (Longuet-Higgins et al. 1963),

$$F(f,\theta) = F(f)\, D(f,\theta) \tag{2.77}$$

where the directional spreading factor $D(f,\theta)$ must satisfy the following condition

$$\int D(f,\theta)\, d\theta = 1\,. \tag{2.78}$$

Longuet-Higgins et al. (1963) defined the directional spreading factor as

$$D(f,\theta) = C(s) \cos^{2s}\left[\frac{\theta - \theta_m(f)}{2}\right] \tag{2.79}$$

where $C(s)$ is a normalization factor needed to satisfy Eq.(2.78) and $\theta_m(f)$ is the

mean wave direction at frequency f. Many different directional spreading factors have been proposed, for example Mitsuyasu (1981). Some of them are summarized by Young (1999).

2.7 Energy Balance

In this section the energy balance equation, which forms the core of a wave model, will be discussed briefly. Starting with the basic spectrum, which is the wave number-direction spectrum $F(k,\theta)$, it is possible to define the wave action density spectrum $N(k,\theta) = F(k,\theta)/\sigma$. $F(k,\theta)$ has invariance characteristics with respect to the physics of wave growth and decay for different water depths, and in general the action density spectrum is conserved.

The wave propagation is defined as

$$\frac{DN}{Dt} = \frac{S_{tot}}{\sigma} \tag{2.80}$$

where D/Dt represents the total derivative and S_{tot} represents the total effect of source and sink terms, defined by

$$S_{tot} = S_{in} + S_{nl4} + S_{ds} + S_{bot} \, . \tag{2.81}$$

These are the general source terms that consist of wind input, nonlinear wave-wave interaction, dissipation , and wave-bottom interaction (found in shallow water cases) respectively.

In the modern third generation operational wave model, WAVEWATCH, the balance equation for the spectrum $N(k,\theta;\mathbf{x},t)$ is defined as

$$\frac{\partial N}{\partial t} + \nabla \bullet \dot{\mathbf{x}} N + \frac{\partial}{\partial k}\dot{k}N + \frac{\partial}{\partial \theta}\dot{\theta}N = \frac{S_{tot}}{\sigma} \tag{2.82}$$

and

$$\dot{\mathbf{x}} = \mathbf{C}_g + \mathbf{U} \tag{2.83}$$

$$\dot{k} = -\frac{\partial\sigma}{\partial d}\frac{\partial d}{\partial s} - \mathbf{k} \bullet \frac{\partial \mathbf{U}}{\partial s} \tag{2.84}$$

$$\dot{\theta} = -\frac{1}{k}\left[\frac{\partial\sigma}{\partial d}\frac{\partial d}{\partial m} - \mathbf{k} \bullet \frac{\partial \mathbf{U}}{\partial m}\right] \tag{2.85}$$

where $\mathbf{C}_g$ is given by C_g and θ, s is a coordinate in the direction θ and m is a coordinate perpendicular to s. Details are given by Tolman and Chalikov (1996) and Tolman (2002).

Equation (2.82) is valid for a Cartesian grid. It must be transferred into a spherical grid for large scale or global applications. Rewriting Eq.(2.82) in longitude λ and latitude ϕ grid implies

$$\frac{\partial N}{\partial t} + \frac{1}{\cos\phi}\frac{\partial}{\partial\phi}\dot{\phi}N\cos\theta + \frac{\partial}{\partial\lambda}\dot{\lambda}N + \frac{\partial}{\partial k}\dot{k}N + \frac{\partial}{\partial\theta}\dot{\theta}_g N = \frac{S_{tot}}{\sigma} \tag{2.86}$$

and

$$\dot{\phi} = \frac{C_g\cos\theta + U_\phi}{R} \tag{2.87}$$

$$\dot{\lambda} = \frac{C_g\sin\theta + U_\lambda}{R\cos\phi} \tag{2.88}$$

$$\dot{\theta}_g = \dot{\theta} - \frac{C_g\tan\phi\cos\theta}{R} \tag{2.89}$$

where R is the radius of the earth and $U_{\phi,\lambda}$ is current component. This model is using a meteorological convention for θ where $\theta = 0$ corresponds to waves travelling from west to east.

Detailed information about energy balance and source terms in WAVEWATCH III is given by Tolman (1999, 2002) and Komen et al. (1994). The nonlinear wave-wave interactions will be discussed in the next chapter.

Chapter 3

Nonlinear Wave-wave Interactions

The ocean can be defined as the superposition of waves, with interactions among wave components which give transfer of energy among the different wave components. Triad interactions involve three waves. In this case, two waves interact nonlinearly and transfer energy to a third wave. However, triad interactions are not significant in deep water. For deep water, sets of four waves or *quadruplet interactions* constitute the significant exchange the energy, when resonance conditions are fulfilled.

Many methods have been proposed to compute the non-linear wave-wave interactions. However, only two common methods will be discussed in detail here, the WTR (*Webb, Tracy and Resio*) method and DIA (*Discrete Interaction Approximation*) method. The WTR method represents an 'exact' solution of the nonlinear transfer equation, while the DIA method is an approximation method which represents a practical method to solve the problem.

3.1 Exact Solution

Hasselmann (1962) used a perturbation method to evaluate the energy flux in a finite-depth gravity wave spectrum resulting from weak non-linear couplings between the spectral components. His fifth-order analysis gave a fourth-order effect comparable

in magnitude to the energy input by wind and the dissipative processes in wind-generated seas. See Hasselmann (1962) for a detail development of this method.

This section will emphasize the work of Webb (1978) and Tracy and Resio (1982). Basing their work on Hasselmann (1962), they created a simple algorithm that can be used to numerically compute nonlinear wave-wave interactions. This section presents a summary of Tracy and Resio (1982). However, the formulations are derived for deep water only. Finite depth formulations can be seen in Resio et al. (2001).

The basic equation describing the non-linear quadruplet wave-wave interactions, as proposed by Hasselmann (1962), is known as the Boltzmann integral. The equation describes the rate of change of action density, S_{nl4}, of a particular wave number due to resonant interactions among quadruplets of wave numbers, i.e.

$$\begin{aligned}\frac{dN_1}{dt} = \iiint & C(\mathbf{k}_1,\mathbf{k}_2,\mathbf{k}_3,\mathbf{k}_4)[N_1N_3(N_4-N_2)+N_2N_4(N_3-N_1)] \\ & \delta(\mathbf{k}_1+\mathbf{k}_2-\mathbf{k}_3-\mathbf{k}_4)\delta(\omega_1+\omega_2-\omega_3-\omega_4)d\mathbf{k}_2d\mathbf{k}_3d\mathbf{k}_4\ .\end{aligned} \tag{3.1}$$

Equation (3.1) describes the rate of change of N_1 at wave number $\mathbf{k}_1$ due to all quadruplet interactions involving $\mathbf{k}_1$. Webb (1978) introduced a transfer function $T(\mathbf{k}_1,\mathbf{k}_3)$

$$\frac{dN_1}{dt} = \int T(\mathbf{k}_1,\mathbf{k}_3)\ d\mathbf{k}_3 \tag{3.2}$$

where

$$\begin{aligned}T(\mathbf{k}_1,\mathbf{k}_3) = 2\iint & C(\mathbf{k}_1,\mathbf{k}_2,\mathbf{k}_3,\mathbf{k}_4)[N_1N_3(N_4-N_2)+N_2N_4(N_3-N_1)] \\ & \delta(\mathbf{k}_1+\mathbf{k}_2-\mathbf{k}_3-\mathbf{k}_4)\delta(\omega_1+\omega_2-\omega_3-\omega_4) \\ & \Theta(|\mathbf{k}_1-\mathbf{k}_4|-|\mathbf{k}_1-\mathbf{k}_3|)\ d\mathbf{k}_2d\mathbf{k}_4.\end{aligned} \tag{3.3}$$

N_i is the action density $N(\mathbf{k}_i)$ at wave number $\mathbf{k}_i$, and ω_i is the angular frequency at $\mathbf{k}_i$. The $\delta(\ldots)$ is the Dirac delta function and the term C is the coupling coefficient

(Webb, 1978; Tracy and Resio, 1982; or Appendix C) and

$$\begin{aligned} \Theta(x) &= 1 \qquad \text{if } x > 0 \\ \Theta(x) &= 0 \qquad \text{if } x \leq 0 \\ x &= |\mathbf{k}_1 - \mathbf{k}_4| - |\mathbf{k}_1 - \mathbf{k}_3| \,. \end{aligned} \tag{3.4}$$

The $\Theta(x)$ function determines a section of the integral which is not defined due to the assumption that $\mathbf{k}_1$ is closer to $\mathbf{k}_4$ than $\mathbf{k}_3$. It reduces the domain of computation by half, and we put factor 2 in Eq.(3.3), because it is assumed that the spectrum is symmetric around 0°.

We will use the following property of the Dirac delta function,

$$\delta(x - a) = 0 \qquad \text{if} \qquad x \neq a \tag{3.5}$$

where x and a are representative functions. The $\int \delta(x-a)dx = 1$ if the region includes $x = a$ and is zero otherwise. Consider

$$\int \delta(\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}_3 - \mathbf{k}_4) d\mathbf{k}_4$$

if $\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}_3 = \mathbf{k}_4$ for conservation of momentum, the $d\mathbf{k}_4$ integral equals one and the transfer integral becomes

$$T(\mathbf{k}_1, \mathbf{k}_3) = 2 \int \Xi(\mathbf{k}) \;\; \delta(\omega_1 + \omega_2 - \omega_3 - \omega_4) \;\; d\mathbf{k}_2 \tag{3.6}$$

where

$$\Xi(\mathbf{k}) = C(\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3, \mathbf{k}_4)[N_1 N_3 (N_4 - N_2) + N_2 N_4 (N_3 - N_1)]\Theta(x).$$

This delta function evaluation limits the wave number configuration to $\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}_3 = \mathbf{k}_4$ or $\mathbf{k}_1 + \mathbf{k}_2 = \mathbf{k}_3 + \mathbf{k}_4$. This means that the tips of the four wave-number vectors must form a parallelogram in wave number space.

In order to evaluate the integral numerically, we must fix values for $\mathbf{k}_1(x, y)$ and $\mathbf{k}_3(x, y)$ and consider the limiting properties of the angular velocity delta function. Then, for each of $(\mathrm{k}_1, \mathrm{k}_3)$, let $W(\mathbf{k}_2)$ equal the argument of the angular velocity delta function. To eliminate the angular velocity delta function, let $W(\mathrm{k}_2)$ be equal to zero; therefore

$$\begin{aligned} W(\mathbf{k}_2) &= \omega_1 + \omega_2 - \omega_3 - \omega_4 \\ &= \omega_1 + \omega(\mathbf{k}_2) - \omega_3 - \omega(\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}_3) \\ &= 0. \end{aligned} \tag{3.7}$$

Consider a $\mathrm{k}_1, \mathrm{k}_2, \mathrm{k}_3$ coordinate system. One point in the $\mathbf{k}_1 - \mathbf{k}_3$ plane would have a whole line of solutions parallel to the $\mathbf{k}_2$ axis. The set of solutions that will satisfy the conservation conditions can be represented as an egg-shaped two-dimensional locus in a Cartesian coordinate system in $\mathbf{k}_2$ space where k_{2x} is the x-axis and k_{2y} is the y-axis, as shown in Fig. 3.1. On this locus, $\mathbf{n}$ is the normal or radial direction and $\mathbf{s}$ is in the increasing θ or tangential direction.

Using a $(\mathbf{n}, \mathbf{s})$ coordinate system for $W(\mathbf{k}_2)$, the transfer integral, Eq.(3.6), can be written in terms of the new coordinate system:

$$T(\mathbf{k}_1, \mathbf{k}_3) = 2 \iint \Xi(\mathbf{k}) \ \delta[W(\mathbf{s}, \mathbf{n})] \ dsdn. \tag{3.8}$$

Another property of the Dirac delta function (Jackson, 1962) is

$$\delta[f(x)] = \frac{\delta(x - x_0)}{\left|\frac{df}{dx}\right|} \qquad \text{where } f(x_0) = 0$$

$$\delta(f)df = \delta(x)dx$$

where x and x_0 are representative functions. Using this property on $\delta[W(\mathbf{s}, \mathbf{n})]$, we find that

$$\delta[W(\mathbf{s}, \mathbf{n})] = \frac{\delta(n - 0)}{\left|\frac{\partial W(\mathbf{s}, \mathbf{n})}{\partial n}\right|}.$$

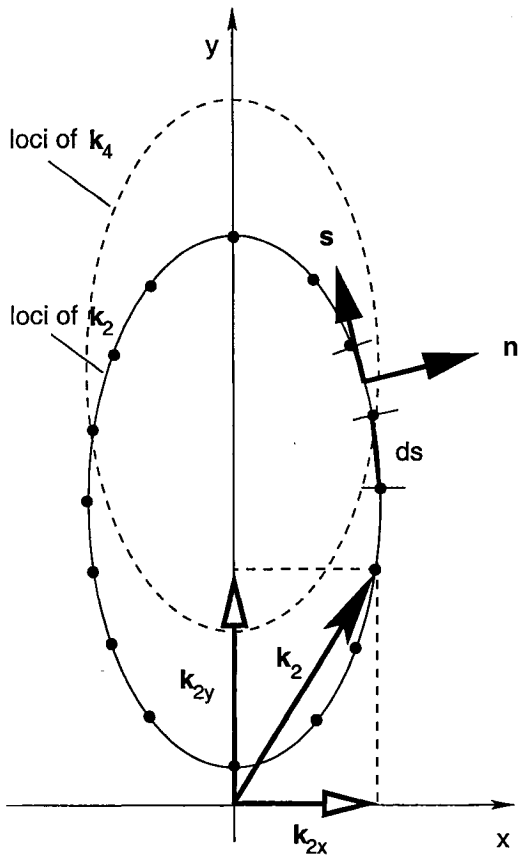


Figure 3.1: Loci of $\mathbf{k}_2$ and $\mathbf{k}_4$.

We must integrate around the s-curve for this property to hold, so that

$$\begin{aligned} T(\mathbf{k}_1, \mathbf{k}_3) &= 2 \iint \Xi(\mathbf{k}) \frac{\delta(n-0)}{\left|\frac{\partial W(\mathbf{s},\mathbf{n})}{\partial n}\right|} \, ds dn \\ &= 2 \oint \Xi(\mathbf{k}) \frac{1}{\left|\frac{\partial W(\mathbf{s},\mathbf{n})}{\partial n}\right|} \, ds. \end{aligned} \tag{3.9}$$

since $\int \delta(n-0) dn = 1$. The normal derivative in the denominator is the magnitude

of a gradient and can be written (Tracy and Resio,1982):

$$\left|\frac{\partial W(\mathbf{s},\mathbf{n})}{\partial n}\right|^{-1} = |\nabla W(k_{2x}, k_{2y})|^{-1}. \tag{3.10}$$

The phase space term $|\frac{\partial W}{\partial n}|$ can be evaluated at each point of the locus by evaluating the gradient of locus, Eq.(3.7). For deep water $\omega \propto k^{1/2}$. The locus equation can be simplified by using $Q = k_1^{1/2} - k_3^{1/2}$ and defining a new vector, $\mathbf{P} = \mathbf{k}_1 - \mathbf{k}_3$, as shown in Fig. 3.2. The expression for $W(\mathbf{k}_2)$ becomes:

$$Q + \left(k_2^{1/2} - |\mathbf{P} + \mathbf{k}_2|^{1/2}\right) = 0. \tag{3.11}$$

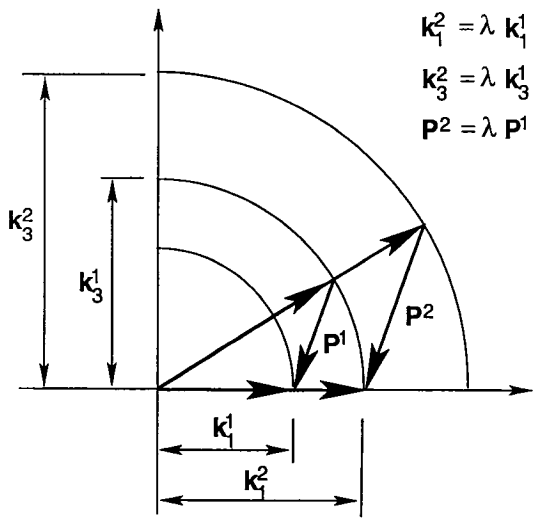


Figure 3.2: The integration grid and vector $\mathbf{P}$.

Since Q will remain constant during calculation of the $\mathbf{k}_2$-contour, the gradient of W will be:

$$\nabla W(k_{2x}, k_{2y}) = \frac{\partial W}{\partial x}\hat{\imath} + \frac{\partial W}{\partial y}\hat{\jmath} \tag{3.12}$$

where

$$\frac{\partial W}{\partial x} = \sqrt{g}\,\frac{\partial\left(k_2^{1/2} - |\mathbf{P}+\mathbf{k}_2|^{1/2}\right)}{\partial x} \tag{3.13}$$

and

$$\frac{\partial W}{\partial y} = \sqrt{g}\,\frac{\partial\left(k_2^{1/2} - |\mathbf{P}+\mathbf{k}_2|^{1/2}\right)}{\partial y}. \tag{3.14}$$

Writing the vectors in rectangular components yields:

$$|\mathbf{P}+\mathbf{k}_2|^{1/2} = \left[(P_x+x)^2 + (P_y+y)^2\right]^{1/4} \tag{3.15}$$

where P_x and P_y are the rectangular components of $\mathbf{P}$ and x and y are the rectangular components of $\mathbf{k}_2$. The magnitude of the gradient in the normal direction is:

$$|\nabla W(k_{2x}, k_{2y})| = \left[\left(\frac{\partial W}{\partial x}\right)^2 + \left(\frac{\partial W}{\partial y}\right)^2\right]^{1/2} \tag{3.16}$$

where

$$\frac{\partial W}{\partial x} = \sqrt{g}\left\{\frac{x}{2}(x^2+y^2)^{-3/4} - \frac{(P_x+x)}{2}[(P_x+2)^2+(P_y+2)^2]^{-3/4}\right\} \tag{3.17}$$

$$\frac{\partial W}{\partial y} = \sqrt{g}\left\{\frac{y}{2}(x^2+y^2)^{-3/4} - \frac{(P_y+y)}{2}[(P_x+2)^2+(P_y+2)^2]^{-3/4}\right\}. \tag{3.18}$$

The transfer integral, equation (3.9), can be written as:

$$\begin{aligned} T(\mathbf{k}_1,\mathbf{k}_3) &= 2\oint [N_1N_3(N_4-N_2) + N_2N_4(N_3-N_1)] \\ &\quad \times C(\mathbf{k})\Theta(x)\frac{1}{|\nabla W(k_{2x},k_{2y})|}\,ds \end{aligned} \tag{3.19}$$

or

$$T(\mathbf{k}_1,\mathbf{k}_3) = 2\oint D(N)\,G(\mathbf{k}) \tag{3.20}$$

where

$$\begin{aligned} D(N) &= [N_1 N_3 (N_4 - N_2) + N_2 N_4 (N_3 - N_1)] \\ G(\mathbf{k}) &= C(\mathbf{k})\Theta(x) \frac{1}{|\nabla W(k_{2x}, k_{2y})|} \, ds. \end{aligned} \tag{3.21}$$

The geometric function, $G(\mathbf{k})$, can be scaled by λ multipled by the basic loci, giving the product of scaling terms (see Tracy and Resio, 1982)

$$\sqrt{\lambda} \times \lambda^6 \times \lambda = \lambda^{15/2}. \tag{3.22}$$

Each successive transfer integral at the same orientation of $\mathbf{k}_1$ and $\mathbf{k}_3$ (parallel $\mathbf{P}$) can be written as

$$G(\mathbf{k})_{\mathbf{P}} = G(\mathbf{k})_{basic} \times \lambda^{15/2}. \tag{3.23}$$

Finally, in polar coordinates, the nonlinear energy transfer, Eq.(3.2), can be computed from the following equation,

$$\frac{dN_1}{dt} = \int_0^{\infty} \int_0^{2\pi} T(\mathbf{k}_1, \mathbf{k}_3) \; k_3 d\theta_3 dk_3. \tag{3.24}$$

Focusing on the integral only, we can simplify Eq.(3.24) as

$$\frac{dN_1}{dt} = \int_0^{\infty} \int_0^{2\pi} \oint_1^{nloci} \ldots \, ds d\theta_3 dk_3. \tag{3.25}$$

This implies that the nonlinear energy transfer must be computed over the entire 2-dimensional spectrum and number of loci of resonance. Figure 3.3 shows the scheme of WTR itegration.

If i is the number of frequencies and j is the number of spreading angles and l is the number of the four-wave interactions points on the resonance loci, the integral needs $i^2 \times j^2 \times l$ calculations to compute S_{nl4}, for the entire ij-grid. By comparison the DIA method (see next section) needs $i \times j$ calculations for the same grid. This

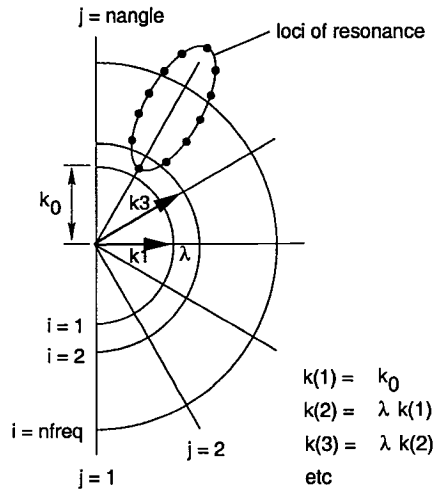


Figure 3.3: Configurations used in the WTR method.

number of computations makes the WTR method more time consuming.

3.2 Approximation Methodology

Exact solutions are time consuming and therefore not suitable for practical wave forecasts. Hasselmann et al. (1985) proposed parameterization methods to compute the nonlinear term efficiently. One of these methods is called the Discrete Interaction Approximation (DIA). This approximation will be discussed further in this section because this method is applied in all modern operational wave models.

DIA method simplifies the exact nonlinear transfer by one mirror-image pair of intermediate-range interaction configurations. It defines $\mathbf{k}_1 = \mathbf{k}_2 = \mathbf{k}$, where $\mathbf{k}_3$ and $\mathbf{k}_4$ have different magnitudes and lie at an angle to $\mathbf{k}$, as shown in Fig. 3.4,

$$\omega_1 = \omega_2 = \omega$$

$$\omega_3 = \omega(1+\lambda) = \omega^+ \tag{3.26}$$

$$\omega_4 = \omega(1-\lambda) = \omega^- \tag{3.27}$$

where $\lambda = 0.25$, and the resonance condition, $\mathbf{k}_1 + \mathbf{k}_2 = \mathbf{k}_3 + \mathbf{k}_4$, implies that the angles θ_3 and θ_4 of wavenumbers $\mathbf{k}_3(\mathbf{k}_+)$ and $\mathbf{k}_4(\mathbf{k}_-)$ relative to $\mathbf{k}$ are $\theta_3 = 11.5°$ and $\theta_4 = -33.6°$ respectively. DIA implies that we may write the change in action density as,

$$\begin{Bmatrix} \delta N \\ \delta N^+ \\ \delta N^- \end{Bmatrix} = \begin{Bmatrix} 2 \\ -1 \\ -1 \end{Bmatrix} C' g^{-8} f^{19} \left[N^2 (N^+ + N^-) - 2NN^+N^- \right] \Delta \mathbf{k} \Delta t \tag{3.28}$$

where C' is a numerical constant that represents the strength of the interaction.

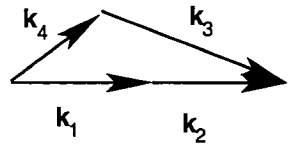


Figure 3.4: The two interaction configurations used in the DIA method.

In practice, the energy spectral densities F with respect to frequency f and propagation direction θ are used in modern wave models like WAM. Rewriting Eq.(3.28) in terms of $F(f, \theta)$ yields

$$\begin{aligned} \begin{Bmatrix} \delta S_{nl} \\ \delta S_{nl}^+ \\ \delta S_{nl}^- \end{Bmatrix} &= \begin{Bmatrix} 2 \\ -1 \\ -1 \end{Bmatrix} C g^{-4} f^{11} \\ &\times \left[F^2 \left(\frac{F^+}{(1+\lambda)^4} + \frac{F^-}{(1-\lambda)^4} \right) - 2 \frac{F F^+ F^-}{(1-\lambda^2)^4} \right] \end{aligned} \tag{3.29}$$

where C is a modified numerical constant proportional to C', and equal to 3×10^7 and F,F^+, and F^- are the energy densities, expressed in frequency-direction space

or $F(f,\theta)$. In finite depth, the nonlinear transfer is defined as

$$S_{nl}(\text{finite depth}) = R(\bar{k}d)\, S_{nl}(\text{infinite depth}) \tag{3.30}$$

where $\bar{k}$ is the mean wave number and R is a scaling factor, stated as

$$R(x) = 1 + \frac{5.5}{x}\left(1 - \frac{5x}{6}\right)\exp\left(-\frac{5x}{4}\right) \tag{3.31}$$

with $x = \frac{3}{4}\bar{k}d$.

Chapter 4

Developing New Methods

This chapter presents proposed new methods to improve the time for computing the nonlinear wave-wave interactions, based on the WTR theory. Each method also is based on a modified Boltzmann integral. The development of each method is shown here in sequence, from basic to advanced improvement.

4.1 DTA Method

4.1.1 Formulation of the model

As seen in Eq.(3.19) in Chapter 3, to find the transfer function, $T(\mathbf{k}_1, \mathbf{k}_3)$, we have to integrate along the loci, over a finite number of points, which we represent as *nloci*. The nonlinear energy transfer must then be computed over the entire spectrum, from the smallest k_3 to the highest k_3 and from $\theta = 0$ to $\theta = 2\pi$ as shown in Eq.(3.24). A typical wave model uses a spectral discretization of 30 wave number bins and 36 angular bins, where $wn(1) = k_0$, $wn(2) = \lambda\, wn(1)$, $wn(3) = \lambda\, wn(2)$, and so on, where λ is the incremental factor.

Suppose there are 36 points on the resonance locus. Therefore to make one evaluation of the nonlinear interactions, one needs $30 \times 36 \times 36$ or 38880 computations for one frequency and one angle bin. The entire spectrum would require $38880 \times 30 \times 36$

or more than $4\text{x}10^7$ computations.

In this section it is shown that certain sets of (k_1, k_3) give the dominant transfer, which means that on those sets the maximum nonlinear interaction occurs, much greater than the nonlinear transfer from other contributions. Figure 4.1 explicitly shows that there is a set of (k_1, k_3) which gives the maximum or dominant transfer in the nonlinear process in these cases.

Experimental computations are conducted for wave conditions represented by snapshot JONSWAP spectra, including the range of peakedness from $\gamma = 1$ to $\gamma = 7$. In terms of the polar coordinates, we use grid (k, θ) with wavenumbers spaced logarithmically in 'rings', $k_{i+1} = \lambda\, k_i$, where λ is usually in the range from about 1.05 to 1.21. We find through numerical experiments that given wavenumbers $k_{1(i)}$ and $k_{3(j)}$, with polar 'ring' indices i and j respectively, the 'ring' of dominant transfer can be stated as:

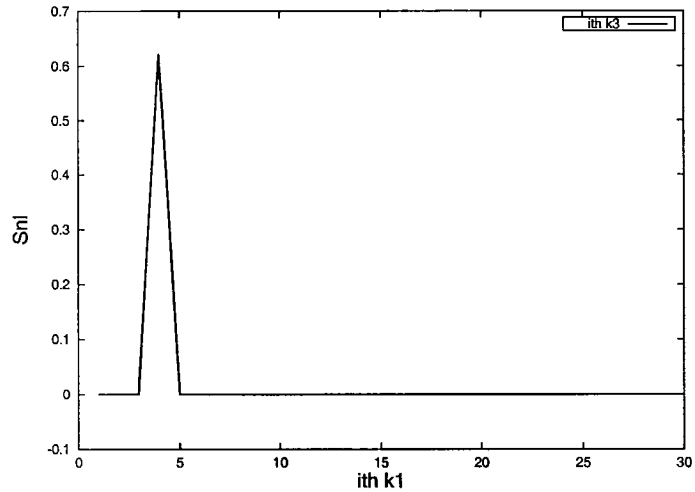
$$j = i + m_d \tag{4.1}$$

where m_d is defined as:

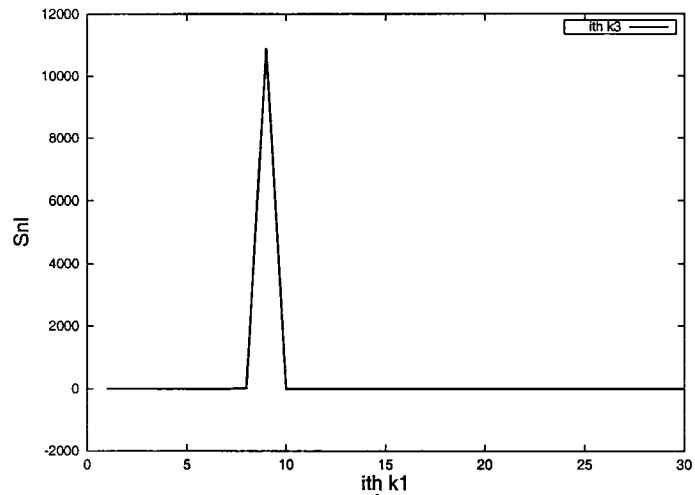
$$\begin{aligned} m_d &= \begin{cases} 1 \text{ if } m \leq 1 \\ \text{nearest integer of } m \text{ if } m > 1, \end{cases} \\ &\text{and} \\ m &= \frac{\ln \lambda_0}{\ln \lambda}\ . \end{aligned} \tag{4.2}$$

Here λ is an incremental factor of $\mathbf{k}$ and λ_0 is the value of λ that is used in most applications in operational third-generation wave models, which equals 1.21. See for example Tolman (1999). In this section $\lambda = 1.21$ is also being used.

With this result, a new method to approximate the Boltzmann integral, which is called as the *Dominant Transfer Approximation* (DTA) is proposed. Rather than integrating over the entire spectral domain from $k_3 = 0$ to $k_3 = \infty$, this method uses



(a) At $i_{k1}^{th} = 5$



(b) At $i_{k1}^{th} = 10$

Figure 4.1: Dominant transfer for certain set of k_1 and k_3, where i_{k1}^{th} is the i^{th} ring of wavenumber k_1.

only the value for k_3 which gives the maximum transfer,

$$\frac{dN_1}{dt} = F_d \int_0^{2\pi} T(\mathbf{k}_1, \mathbf{k}_{3_d})\ \Delta k_{3_d} k_{3_d} d\theta_3 \tag{4.3}$$

where

$$F_d = \frac{Max(Snl_{all-k})}{Max(Snl_{DTA})} \tag{4.4}$$

and where F_d, or the dominant factor, is a scaling factor. In this approximation, Snl_{all-k} is pre-computed from the entire spectrum (see Eqn.(3.24)) and Snl_{DTA} is computed from the dominant set,

$$Snl_{DTA} = \frac{dN_1}{dt} = \int_0^{2\pi} T(\mathbf{k}_1, \mathbf{k}_{3_d})\ \Delta k_{3_d} k_{3_d} d\theta_3\ . \tag{4.5}$$

The method for finding F_d is presented in the next section. Using Eq.(4.3), evaluation of one point of nonlinear transfer for one time step of the 30×36 spectral grid with 36 points on the locus of resonance only needs 1080 computations.

4.1.2 Finding the dominant factor, F_d

In this section, we are going to discuss F_d, the key to the DTA method. We want to see how F_d behaves if major JONSWAP parameters change. Before we implement this method to a real wave model, we must do some experiments to collect the data to find F_d. First the grid that is common for most wave models is set up. Then we compute S_{nl4} with the WTR method by employing Eq.(3.24). Next, we compute the same case as Eq.(4.5). Finally, we determine F_d based on the positive lobe of the nonlinear transfer, following Eq.(4.4), because this drives the spectral downshifting during wave development (Komen et.al., 1994). Once we find F_d, we can use it in a wave model to compute S_{nl4} with Eq.(4.3).

We must examine the behaviour of F_d as a function of the key JONSWAP parameters. We consider a variety of cases defined by prescribed JONSWAP input spectra, with Phillip's α coefficient varying from 0.001 to 0.04, f_p varying from 0.2 to 0.4 Hz, peakedness varying as $\gamma = 1, 3, 5$, and 7 and spectral spreading factor $= f(cos^{2n}\theta)$ varying with n=1, 2 and 6, representing windsea as well as swell cases. The experiments give results as shown in Figs. 4.2, 4.3a, and 4.3b. Results show that while F_d is a function of γ and spectral spreading $cos^{2n}\theta$, the Phillips' α coefficient and

peak frequency, f_p, have little influence on F_d. Therefore F_d is a function of γ and spreading factor only. Detailed discussion about this method can be found in Susilo and Perrie (2006a).

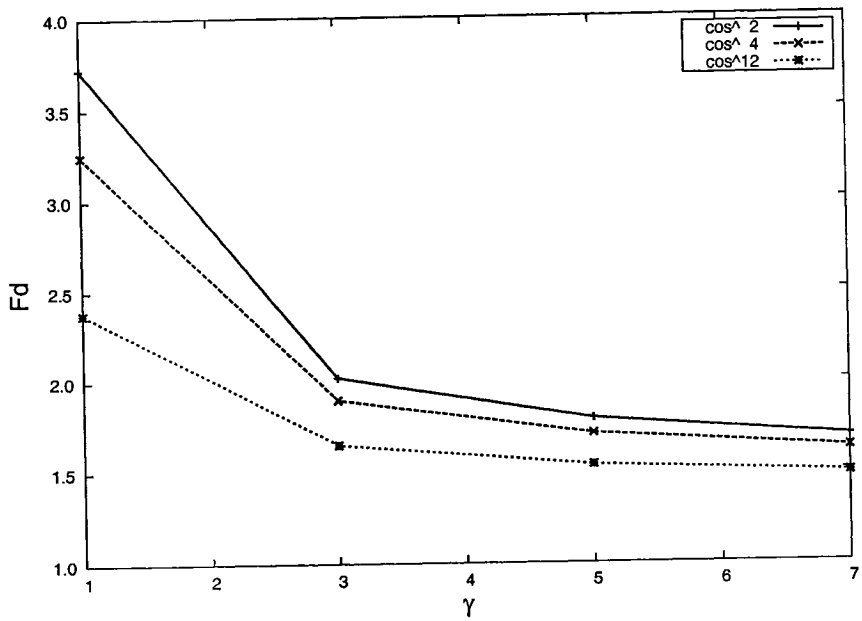
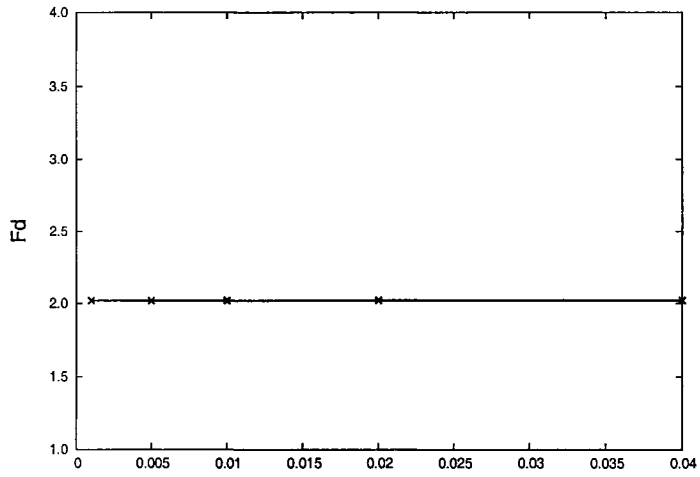


Figure 4.2: F_d as function of γ and *spreading factor*.

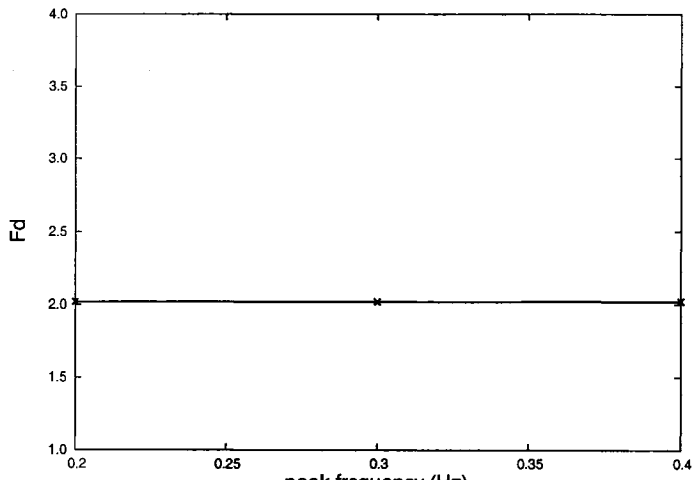
4.2 DTA and Fuzzy Logic

4.2.1 Background of fuzzy logic

Going back a couple of millennia, Aristotle formulated the idea of a bivalent value (two crisp values or a crisp set), for example A or not-A, alternately, an apple or not an apple. Aristotle's teaching is basic to the digital era, consisting of 0 or 1. However, not all cases can be analyzed by bivalent values. Suppose one eats the apple bit by bit, and finally there is no apple anymore. It goes from an apple to nothing. When it is down to half an apple, is it an apple or nothing (not-apple) ? In this case, we



(a) F_d vs α.



(b) F_d vs peak frequency.

Figure 4.3: F_d for different α and *peak frequency*.

can not say 'Apple *or* not-Apple' but we have to say 'Apple *and* not-Apple'. Now we are entering a new concept of multivalence, commonly called fuzzy logic. This is a system of logic in which a statement can be stated as a continuum of values in between 'false or true' or '0 or 1'.

In the 1960s, Prof. Lotfi A. Zadeh at the University of California Berkeley, introduced the fuzzy set. But this idea originally received a huge amount of criticism at that time. However, in the next decade, he continually broadened the fuzzy set theory. In 1974, Mamdani developed a controller for the steam engine based on a fuzzy algorithm from Zadeh's paper. Since then, applications of the fuzzy set or fuzzy logic in industrial settings have expanded, especially in Japan and Europe. More on the history and progress of fuzzy logic can be found in Yen and Langari (1999).

Next, we are going to review the crisp and fuzzy set in more detail. Table 4.1 shows some differences between bivalence and multivalence as listed by Kosko (1993).

Table 4.1: Bivalence vs multivalence (after Kosko, 1993).

BIVALENCE	MULTIVALENCE
A **or** not-A	A **and** not-A
all or none	some degree
0 or 1	continuum between 0 and 1

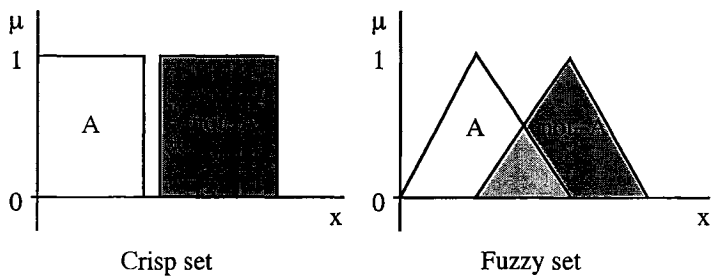


Figure 4.4: Bivalence and multivalence.

Figure 4.4 depicts a diagram of bivalence (or crisp set) and multivalence (fuzzy set) and its membership function, μ, or as it is called, grade of belonging. In a crisp

set, a membership of element x of set A or $not - A$ is defined by:

$$\mu_A(x) = \begin{cases} 0, & \text{if } x \notin A \\ 1, & \text{if } x \in A \end{cases}$$

$$\mu_{not-A}(x) = \begin{cases} 0, & \text{if } x \notin not - A \\ 1, & \text{if } x \in not - A \end{cases}$$

However, in fuzzy set theory, the membership μ is defined by $\mu_A(x) \in [0, 1]$ or $\mu_{not-A}(x) \in [0, 1]$, that expresses the degree to which x belongs to A, or $not - A$.

As a membership function differs between two sets, the logic of sets is also different (see Schmid (2005) or other references about basic fuzzy logic). Crisp logic defines:

- AND : $A \cap B$
- OR : $A \cup B$
- NOT : A'.

Fuzzy logic defines (the diagrams are shown in Fig. 4.5):

- AND : $\mu_{A \cap B} = \min[\mu_A(x), \mu_B(x)]$
- OR : $\mu_{A \cup B} = \max[\mu_A(x), \mu_B(x)]$
- NOT : $\mu_{A'} = 1 - \mu_A(x)$.

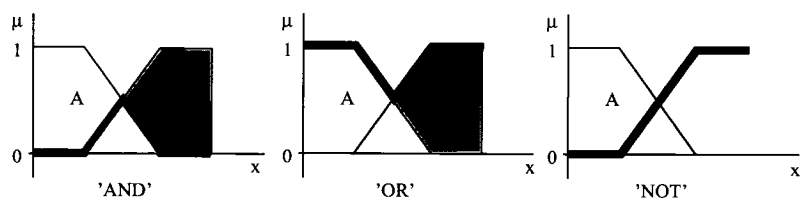


Figure 4.5: Logic of fuzzy set shown in bold line.

An example showing how the fuzzy logic formulation works can be depicted as shown in Figs. 4.6-4.7. The *Fuzzifier* will compute the membership of inputs, then the *Interencer* will denote the membership of outputs based on the logic of the fuzzy set. The *Interencer* also aggregates all outputs. Of course, in the end we need a single value or crisp value, not a multivalence value, and this is a job that is specified by the *Defuzzifier* when it determines the crisp value.

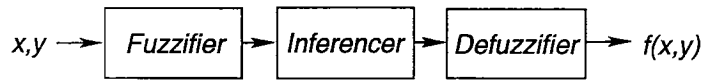


Figure 4.6: Fuzzy controller.

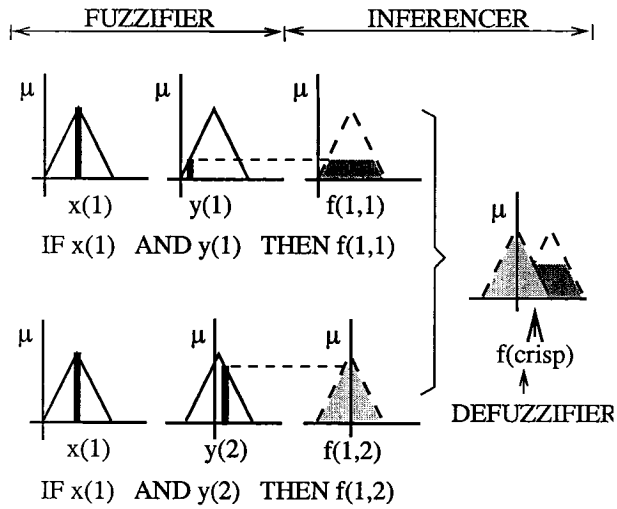


Figure 4.7: Example of fuzzy controller.

4.2.2 Use of fuzzy logic to determine F_d

As shown in previous section, F_d is a function of peakedness (γ) and spectral spreading factor, for a given spectrum. Thus, there are two inputs to get one output. Because

peakedness and spectral spreading are usually only defined once at the initialization of a simulation, we need to specify these variables so they can be determined at any time.

We define new variables, $slope_g$ to represent peakedness and $slope_s$ to represent the spectral spreading,

$$slope_g = \frac{1 - \frac{n(k_{max+1}, \theta_{max})}{n(k_{max}, \theta_{max})}}{(\lambda - 1)\, k_{max}} \tag{4.6}$$

$$slope_s = \frac{1 - \frac{n(k_{max}, \theta_{max+1})}{n(k_{max}, \theta_{max})}}{\Delta\theta\, k_{max}} \tag{4.7}$$

where n is the action density, λ is the incremental factor, (k_{max}, θ_{max}) is the location of (k, θ) where n is maximum. See Fig. 4.8 for details.

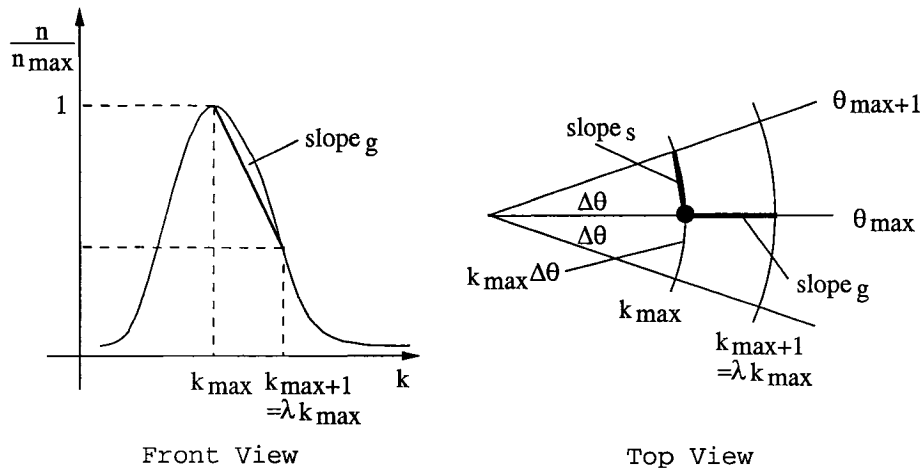


Figure 4.8: The $slope_g$ and $slope_s$.

The next step is to use Fuzzy Logic (FL) to compute the $F_d(slope_g, slope_s)$. Using 'if-then' rules, FL provides a simple way to determine F_d based on some numerical experiments that have to be done. On the other hand, a number, such as F_d, also can be defined by FL (Kosko, 1993). For instance, a number 0.9 can be defined as

90% of unity or 10% of zero (see schematic model suggested in Figure 4.9). Thus, we apply FL as a methodology to determine F_d.

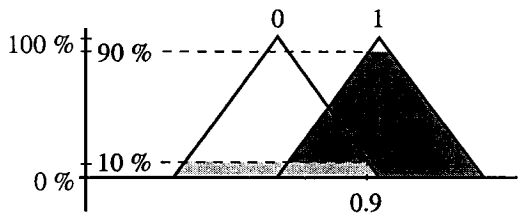


Figure 4.9: A number defined by FL.

In this case, a fuzzy controller gives one output from two inputs. The x and y represent $slope_g$ and $slope_s$ while $f(x, y)$ represents $F_d(slope_g, slope_s)$ respectively. Case examples are conducted to collect the FL required data, then sets of rules are determined. Samples are taken, representing windsea as well as swell cases, for peakedness varying as = 1, 3, 5, and 7 and spectral spreading varying as $cos^{2p}\theta$ where $p = 1$, 2, and 6.

Table 4.2 shows data from computed samples. Before establishing FL rules, it is convenient to relabel each case with a simple naming convention. We classify as '*Tiny*', '*Small*', '*Big*' and '*Large*' representing small $slope_g$ to large $slope_g$ and we also classify varying spreading factors as '*Low*', '*Mid*', and '*High*' for small $slope_s$ to large $slope_s$ conditions, respectively, in Table 4.3. From Table 4.3, we define the rules shown in Table 4.4. Membership functions of Table 4.3 can be depicted as shown in Fig. 4.10.

Table 4.2: F_d from computed samples.

F_d		$slope_g$			
		0.0069	0.5124	0.5782	0.6138
$slope_s$	0.1728	3.7209	2.0181	1.7889	1.6875
	0.3403	3.2459	1.8951	1.7037	1.6217
	0.9615	2.3765	1.6473	1.5317	1.4835

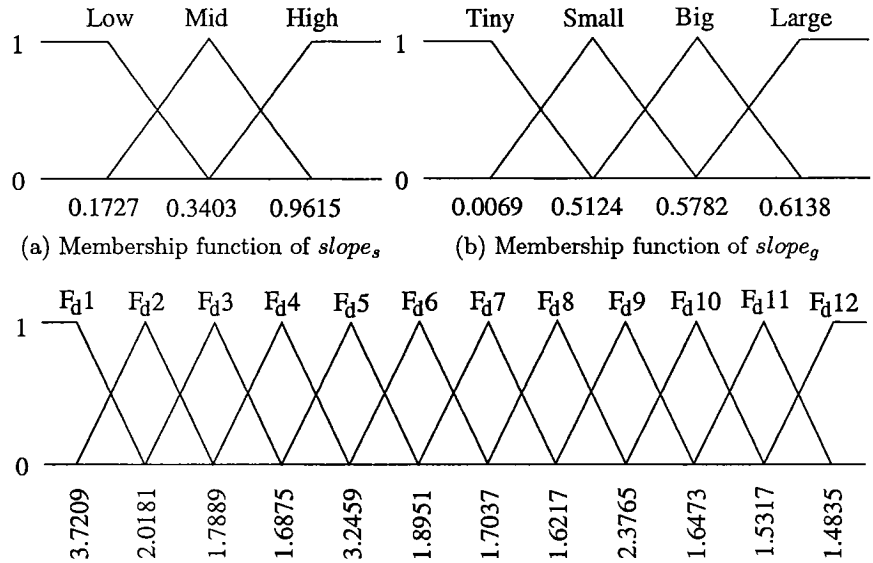
Table 4.3: Rewriting matrix of F_d in a simplified form.

	Tiny	Small	Big	Large
Low	F_d1	F_d2	F_d3	F_d4
Mid	F_d5	F_d6	F_d7	F_d8
High	F_d9	F_d10	F_d11	F_d12

Table 4.4: If-then FL rules from study cases.

1.	If *Low*	AND	*Tiny*	then F_d1
2.	If *Low*	AND	*Small*	then F_d2
3.	If *Low*	AND	*Big*	then F_d3
4.	If *Low*	AND	*Large*	then F_d4
5.	If *Mid*	AND	*Tiny*	then F_d5
6.	If *Mid*	AND	*Small*	then F_d6
7.	If *Mid*	AND	*Big*	then F_d7
8.	If *Mid*	AND	*Large*	then F_d8
9.	If *High*	AND	*Tiny*	then F_d9
10.	If *High*	AND	*Small*	then F_d10
11.	If *High*	AND	*Big*	then F_d11
12.	If *High*	AND	*Large*	then F_d12

The next example will show how the FL method works. Suppose we have inputs $slope_s$ and $slope_g$ as shown in Figure 4.11a and Figure 4.11b.



(a) Membership function of $slope_s$ (b) Membership function of $slope_g$

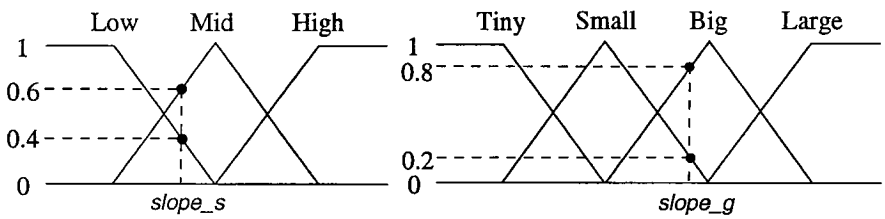
(c) Membership function of Fd

Figure 4.10: FL membership function of inputs and output.

The fuzzifier gives

$$\begin{aligned} \mu_{Low}(slope_s) &= 0.4 \\ \mu_{Mid}(slope_s) &= 0.6 \\ \mu_{Small}(slope_g) &= 0.2 \\ \mu_{Big}(slope_g) &= 0.8\,. \end{aligned}$$

With outputs from the fuzzifier, the inferencer defines 4 rules which are 'fired' or activated because of input combinations. Condition '*Low* **AND** *Small*' will turn on rule number 2. '*Low* **AND** *Big*' will turn on rule number 3. Other combination pairs, '$Mid - Small$' and '$Mid - Big$' will fire rules number 6 and 7 respectively. Following is a summary of 'fired' rules and the corresponding values of $Fd(\mu_A \cap \mu_B) = min(\mu_A, \mu_B)$.



(a) $slope_s$ (b) $slope_g$

Figure 4.11: Example of inputs, $slope_s$ and $slope_g$.

Rule 2. If *Low* **AND** *Small* then F_d2

$$\mu_{Low}(slope_s) \cap \mu_{Small}(slope_g) = \min[\mu_{Low}(slope_s), \mu_{Small}(slope_g)]$$
$$= \min[0.4, 0.2] \rightarrow 0.2\ F_d2$$

Rule 3. If *Low* **AND** *Big* then F_d3

$$\mu_{Low}(slope_s) \cap \mu_{Big}(slope_g) = \min[\mu_{Low}(slope_s), \mu_{Big}(slope_g)]$$
$$= \min[0.4, 0.8] \quad \rightarrow 0.4\ F_d3$$

Rule 6. If *Mid* **AND** *Small* then F_d6

$$\mu_{Mid}(slope_s) \cap \mu_{Small}(slope_g) = \min[\mu_{Mid}(slope_s), \mu_{Small}(slope_g)]$$
$$= \min[0.6, 0.2] \rightarrow 0.2\ F_d6$$

Rule 7. If *Mid* **AND** *Big* then F_d7

$$\mu_{Mid}(slope_s) \cap \mu_{Big}(slope_g) = \min[\mu_{Mid}(slope_s), \mu_{Big}(slope_g)]$$
$$= \min[0.6, 0.8] \quad \rightarrow 0.6\ F_d7\ .$$

The final step is the defuzzifier, which maps the fuzzy set from inputs to the output. There are many approaches to defuzzification, as mention by Patyra (1996),

i.e.

- Centre of area (COA)
- Centre of gravity (COG)
- Height defuzzification (HD)
- Centre of largest area (COLA)
- Mean of maxima (MOM).

Two methods which are common are the centre of gravity (COG) method and the centre of average (COA) method. The COG method calculates the centroid of the area of all membership outputs, and in most cases the centre of gravity is the same as the centre of area. Therefore these names often refer to the same method. More detailed definitions of the defuzzification method can be found in Patyra (1996). The methods are illustrated in Figure 4.12.

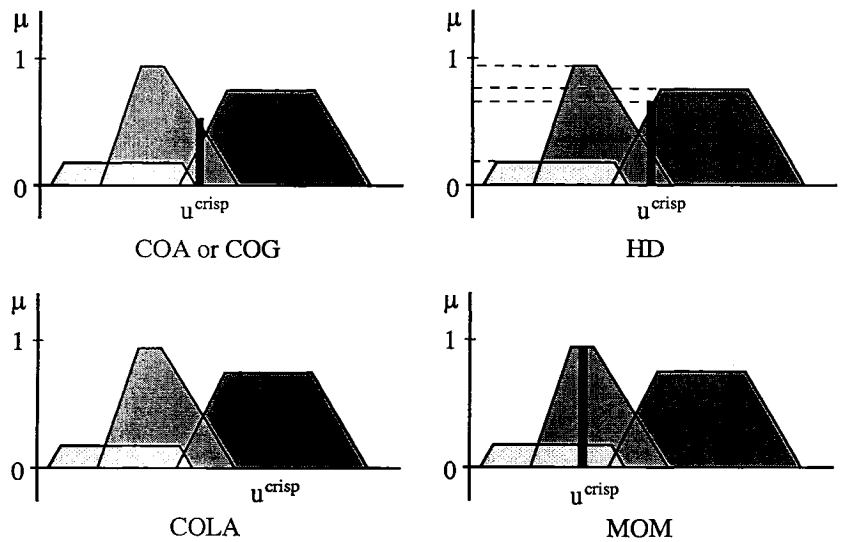


Figure 4.12: Methods of defuzzification (after Patyra, 1996).

We denote the final output as u^{crisp} and we compute the output with the COA method as defined by the following relation (see Passino, 1998),

$$u^{crisp} = \frac{\sum_i \mu_{premise_{(i)}} f_i}{\sum_i \mu_{premise_{(i)}}} . \tag{4.8}$$

Finally, we obtain defuzzification results from the inferencer, which yields

$$u^{crisp} = \frac{0.2F_d2 + 0.4F_d3 + 0.2F_d6 + 0.6F_d7}{0.2 + 0.4 + 0.2 + 0.6} .$$

In this case, $\mu_{premise_{(i)}}$ is $\mu_{premise_{(i)}}(slope_s)$ and $\mu_{premise_{(i)}}(slope_g)$. The final output, u^{crisp}, is the average of all F_d's from four relevant results that follow from the inputs. Susilo et al. (2006) have presented results and further discussion about this method.

4.3 AvDI Method

As can be seen on Eq.(3.20), the transfer integral T consists of density function D and geometry function G. Tracy and Resio (1982) used a special grid and scaling symmetries, and with this method they only compute the basic geometry function G_{basic}. Other Gs are denoted as $G_{next} = G_{basic} \times scale_{factor}$.

Based on Tracy and Resio's work, and the need to reduce integration loops, Susilo and Perrie (2006a) developed a method called the *Dominant Transfer Approximation* or DTA, as described in the previous sections. They found a set of (k_1, k_3) which gives a maximum transfer. See Fig. 4.1 for examples.

The DTA formulation selects the set of (k_1, k_3) and uses a multiplication factor or a scaling factor to approximate the integral over the 2-dimensional spectrum. Equation (3.25) can be rewritten as:

$$\frac{dN_1}{dt} \approx F_d \int_0^{2\pi} \oint_1^{nloci} \ldots \, ds d\theta_3 \qquad \text{or} \tag{4.9}$$

$$\frac{dN_1}{dt} \approx F_d \int_0^{2\pi} T(\mathbf{k}_1, \mathbf{k}_{3_d}) \, \Delta k_{3_d} k_{3_d} d\theta_3 \tag{4.10}$$

where F_d is a scaling factor, the so-called dominant factor and $(\mathbf{k}_1, \mathbf{k}_{3_d})$ is a set of k_1 and k_3 where maximum transfer occurs. Further improvement of the DTA method with Fuzzy Logic was developed by Susilo et al. (2006b), as dissused in Section 4.2.

A new method called *Advanced Dominant Interaction* or AvDI is proposed in this section, whereby a maximum transfer occurs at a certain spreading angle, as shown in Fig. 4.13. The use of AvDI enables a speed up in the computation by reducing the loops and uses the Newton-Cotes method to find the integral along the loci of resonance.

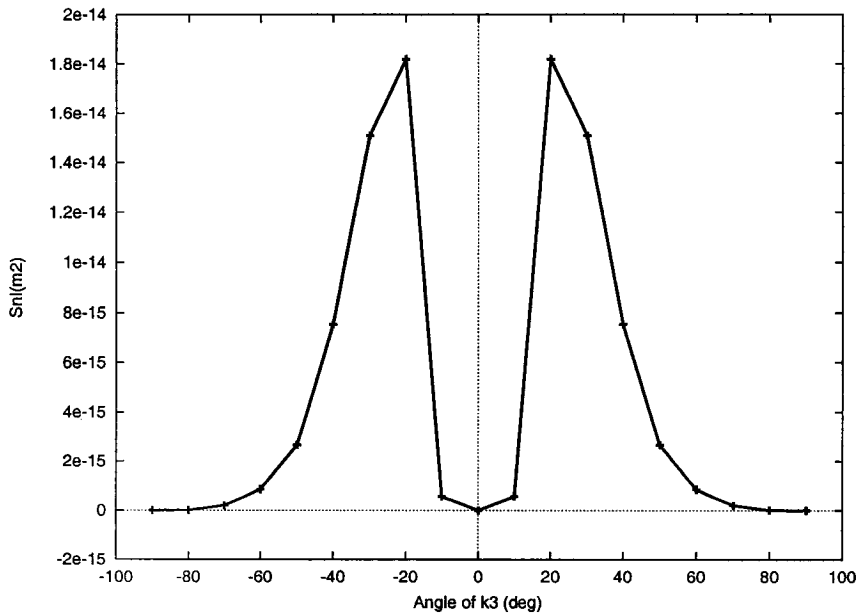


Figure 4.13: Example of dominant transfer along the angle k_3.

We apply the closed Newton-Cotes formula known as the *trapezoidal rule*. It approximates the area under a curve by a trapezoid. Following will explain the Newton-Cotes method. For simplicity, we take 2-point for the example, see Fig. 4.14 for detail. From Fig. 4.14, we can define

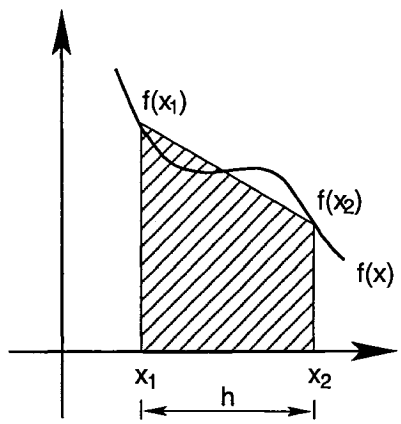


Figure 4.14: Finding the area of an integral by Newton-Cotes method (after Weisstein, 2004).

$$x_2 = x_1 + h \tag{4.11}$$

$$h = x_2 - x_1 \,. \tag{4.12}$$

The Lagrange interpolating polynomial through the points (x_1, f_1) and (x_2, f_2) states

$$\begin{aligned} P_2(x) &= \frac{x - x_2}{x_1 - x_2} f_1 + \frac{x - x_1}{x_2 - x_1} f_2 \\ &= \frac{x - (x_1 + h)}{-h} f_1 + \frac{x - x_1}{h} f_2 \\ &= \frac{x}{h}(f_2 - f_1) + \left(f_1 + \frac{x_1}{h} f_1 - \frac{x_1}{h} f_2\right) \end{aligned} \tag{4.13}$$

where index 2 refers to the points that are part of the formulation, integrating over

the interval gives

$$\begin{aligned}
\int_{x_1}^{x_2} f(x)\,dx &\approx \int_{x_1}^{x_2} P_2(x)\,dx \\
&\approx \int_{x_1}^{x_2} \frac{x}{h}(f_2 - f_1) + \left(f_1 + \frac{x_1}{h}f_1 - \frac{x_1}{h}f_2\right)\,dx \\
&\approx \frac{1}{2h}(f_2 - f_1)[x^2]_{x_1}^{x_2} + \left(f_1 + \frac{x_1}{h}f_1 - \frac{x_1}{h}f_2\right)[x]_{x_1}^{x_2} \\
&\approx \frac{1}{2h}(f_2 - f_1)[x_2^2 - x_1^2] + \left(f_1 + \frac{x_1}{h}f_1 - \frac{x_1}{h}f_2\right)[x_2 - x_1] \\
&\approx \frac{1}{2h}(f_2 - f_1)(x_2 + x_1)(x_2 - x_1) + \left(f_1 + \frac{x_1}{h}f_1 - \frac{x_1}{h}f_2\right)(h) \\
&\approx \frac{1}{2h}(f_2 - f_1)[(x_1 + h) + x_1](h) + (f_1 h + x_1 f_1 - x_1 f_2) \\
&\approx \frac{1}{2}(f_2 - f_1)(2x_1 + h) + f_1 h + x_1(f_1 - f_2) \\
&\approx x_1(f_2 - f_1) + \frac{1}{2}h(f_2 - f_1) + hf_1 - x_1(f_2 - f_1) \\
&\approx \frac{1}{2}h(f_1 + f_2)\,. \qquad (4.14)
\end{aligned}$$

This approximation can be written as

$$\int_{x_1}^{x_2} f(x)\,dx = \frac{1}{2}h(f_1 + f_2) - \frac{1}{12}h^3 f''(\xi) \qquad (4.15)$$

where the last term is the amount of error and $x_1 \leq \xi \leq x_2$, see Krommer and Ueberhuber (1994). Another approximation with a different number of points can be found in Ueberhuber (1997) or Weisstein (2004).

We use 2×10 points to represent the integral of all resonance points of the loci. With formulae from Ueberhuber (1997), a 10 points integration can be stated as

$$\begin{aligned}
\int_{x_1}^{x_{10}} f(x)\,dx \;=\; & \frac{9}{89600}h\,[2857(f_1 + f_{10}) + 15741(f_2 + f_9) + 1080(f_3 + f_8) \\
& +19344(f_4 + f_7) + 5778(f_5 + f_6)] - \frac{173}{14620}h^{11} f^{(10)}(\xi)\,. \qquad (4.16)
\end{aligned}$$

We can denote the integral along the loci of resonance as (this is an example of the

formulae for a half section only)

$$\oint_1^{nloci/2} f(s)\,ds = \frac{9}{89600}\Delta s\,[2857(f_{s1}+f_{s10}) + 15741(f_{s2}+f_{s9}) + 1080(f_{s3}+f_{s8}) + 19344(f_{s4}+f_{s7}) + 5778(f_{s5}+f_{s6})] - \frac{173}{14620}\Delta s^{11} f^{(10)}(\xi)\,. \tag{4.17}$$

With Eq.(4.17) and the dominant transfer along the related frequency and spreading angles, Eq.(4.9) can be simplified as

$$\frac{dN_1}{dt} \approx F_d \sum_{\theta_1}^{\theta_2} \oint_1^{18} \ldots\, ds \qquad \text{or} \tag{4.18}$$

$$\frac{dN_1}{dt} \approx F_d\,[T(\mathbf{k}_1, \mathbf{k}_{3_{d(-)}})\;\Delta k_{3_d} k_{3_d} \Delta\theta_3 + T(\mathbf{k}_1, \mathbf{k}_{3_{d(+)}})\;\Delta k_{3_d} k_{3_d} \Delta\theta_3] \tag{4.19}$$

where

$$\mathbf{k}_{3_{d(-)}} = (k_{3_d}, \theta_{k_1} - \frac{1}{9}\pi) \tag{4.20}$$

$$\mathbf{k}_{3_{d(+)}} = (k_{3_d}, \theta_{k_1} + \frac{1}{9}\pi)\,. \tag{4.21}$$

Now, computing ij-grid spectra with l locus points does not need $i^2 \times j^2 \times l$ calculations but only $i \times j \times 1 \times 2 \times 18$. Chapter 6 will show results of this method.

Chapter 5

Wave Modelling

Wave models have been developed over the last several decades. The first generation wave models did not have an explicit S_{nl} term. Non-linear energy transfers are either not included or expressed implicitly through S_{in} and S_{ds} terms. The second generation wave models handle the S_{nl} term by parametric methods by applying a reference spectrum (for example the JONSWAP spectrum) to reorganize the energy over the frequencies. The third generation wave models calculate the non-linear energy transfers explicitly without constraints on the shape of the wave spectrum. But it is necessary to make both analytic and numerical approximations to expedite the calculations.

We are going to discuss three popular wave models, *WAM, SWAN,* and *WAVEWATCH III*, and a new model developed here, called *WW3-AvDI.*

5.1 WAM

WAM (Wave Modelling) is the wave model developed by The Wave Model Development and Implementation (WAMDI) Group (1988). This model is the most widely used and best tested ocean wave model. The code is well documented and highly optimized to run on many different computational platforms. The WAM model has been extensively used for forecasting on global and regional scales at many weather

prediction centers around the world. More information about this model can be found in the WAMDI Group (1988) and Komen et al. (1994).

5.2 SWAN

The SWAN or Simulating WAves Nearshore model is a third-generation wave model for obtaining realistic estimates of wave parameters in coastal areas, lakes and estuaries from given wind, bottom and current conditions. The model is based on the wave action balance equation with sources and sinks, developed by Delft University of Technology, The Netherlands. SWAN can be used on any scale relevant for wind-generated surface gravity waves too. However, the WAM model and the WAVEWATCH III model, which have been designed specifically for ocean applications, are probably one order of magnitude more efficient than SWAN.

The basic scientific philosophy of SWAN is identical to that of WAM (Cycle 3 and 4). It uses the same formulations for the source terms (although SWAN uses the adapted code for the DIA technique). On the other hand, SWAN contains some additional formulations primarily for shallow water. Moreover, the numerical techniques employed are very different. WAVEWATCH III does not only use different numerical techniques but also different formulations for the wind input and the whitecapping.

The current version of SWAN is 40.51 and succeeds the previous version 40.41 as of August 2006. SWAN accounts for the following physics:

- wave propagation in time and space
- shoaling and refraction due to current and bathymetry
- frequency shifting due to currents and non-stationary depth
- wave generation by wind
- three- and four-wave interactions
- white-capping, bottom friction and depth-induced breaking

- wave-induced set-up
- transmission through and reflection (specular, diffuse and scattered) against obstacles.

SWAN computations can be made on a regular and a curvi-linear grid in a Cartesian or spherical co-ordinate system. SWAN runs can be done in serial mode, i.e. one SWAN program on one processor, as well as in parallel mode. SWAN provides the following output quantities (numerical files containing tables, maps and time series):

- one- and two-dimensional spectra
- significant wave height and wave periods
- average wave direction and directional spreading
- one- and two-dimensional spectral source terms
- root-mean-square of the orbital near-bottom motion
- wave-induced force (based on the radiation-stress gradients)
- dissipation, etc.

More detail and updated information about SWAN can be found on the SWAN website (2003) and a description of the model and of the physics was given by Booij et al. (1999) and Ris et al. (1999).

5.3 WAVEWATCH III

WAVEWATCH III is a third generation wave model developed at NOAA/NCEP by Tolman (1997, 1999a). It is a further development of the model WAVEWATCH I, built at Delft University of Technology, and WAVEWATCH II, built at NASA,

Goddard Space Flight Center. However, WAVEWATCH III differs from its previous programs in many important points such as the governing equations, the model structure, the numerical methods and the physical parameterizations.

WAVEWATCH III solves the spectral action density balance equation for wavenumber -direction spectra. The governing equations of WAVEWATCH III include refraction and straining of the wave field due to temporal and spatial variations of the mean water depth and of the mean current (tides, surges etc.), when applicable. Parameterizations of physical processes (source terms) include wave growth and decay due to the actions of wind, nonlinear resonant interactions, dissipation ('whitecapping') and bottom friction.

Wave propagation is considered to be linear. Relevant nonlinear effects such as resonant interactions are, therefore, included in the source terms (physics). The model includes options for choosing two source term packages: the first is based on cycles 1 through 3 of the WAM model (WAMDIG 1988); the second is based on Tolman and Chalikov (1996). The source term packages are selected at the compile level. For research purposes only, the model includes a full nonlinear interaction source term option (version 2.22).

The numerical features of this model are as follows:

- written in ANSI standard FORTRAN 77 (version 1.15 and 1.18) and FORTRAN 90 (version, 2.22)
- uses a regularly spaced longitude-latitude grid (longitude and latitude increment do not need to be equal) and, optionally, a Cartesian grid
- wave energy spectra are discretized using a constant directional increment (covering all directions), and a spatially varying wavenumber grid
- more options for propagation schemes with different orders of accuracy
- the source terms are integrated in time using a dynamically adjusted time stepping algorithm, which concentrates computational efforts during conditions with rapid spectral changes, for example, during explosively developing storms

- the model can optionally be compiled for a single processor, as well as parallel processors.

The model provides binary or ASCII output, as well as output for the GrADS graphics package by means of post processing. The output options are:

- gridded fields of 18 (16 for old versions) input and mean wave parameters such as the significant wave height, directions, frequencies etc.
- output of wave spectra at selected locations
- output of wave spectra along arbitrary tracks
- up to 9 restart files per model run
- files with boundary data for up to 9 separate nested runs

More details and updated information about WAVEWATCH III can be found on the website, Tolman (1997).

5.4 WW3-AvDI

In this thesis, the AvDI method was installed in WAWEWATCH III version 1.18. Because WAVEWATCH III (or WW3) provides architecture options for installing new source terms, a new subroutine for computing the nonlinear term using AvDI, was constructed. The new wave model is called WW3-AvDI.

Version 1.18 was chosen because it was the last version of WAVEWATCH III before the new FORTRAN 90 version was released. Moreover version 1.18 written in FORTRAN 77 has proven to produce good results. FORTRAN 77 also makes the model code easy to understand and to modify.

WAVEWATCH III version 1.18 puts files into several folders which can be classified as *compiler folder*, *raw material folder*, *input folder*, and *working folder*. Following is the name of each folder and its contents:

- **aux** raw auxiliary programs (source code etc.)
- **bin** executables and shell scripts for compiling and linking
- **exe** WWATCH executable files
- **ftn** source code and makefiles
- **inp** input files
- **obj** object modules of all subroutines of WWATCH
- **work** auxiliary work directory.

The model can be divided into four sections, as shown in Fig. 5.1. Because we use FORTRAN 77, first we have to set up the required dimension arrays. Then, the model compiles raw files and links to executable files. The next step is to set up inputs for boundary conditions and initial conditions, after which the main program can be run. The results from main program are managed by the postprocessor.

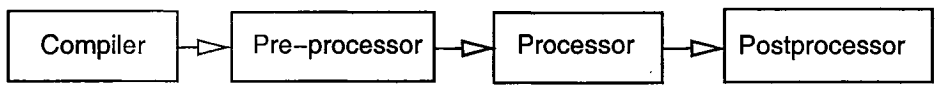


Figure 5.1: Diagram of the wave model.

Excluding the compiler, the model consists of six main programs; three pre-processor programs, one main processor program, and two postprocessor programs, i.e.

- Pre-processor: **ww3_grid**
  ww3_prep
  ww3_strt
- Processor: **ww3_shel**
- Postprocessor: **ww3_outf**
  ww3_outp.

The main program is built by many subprograms. Appendix D shows the main

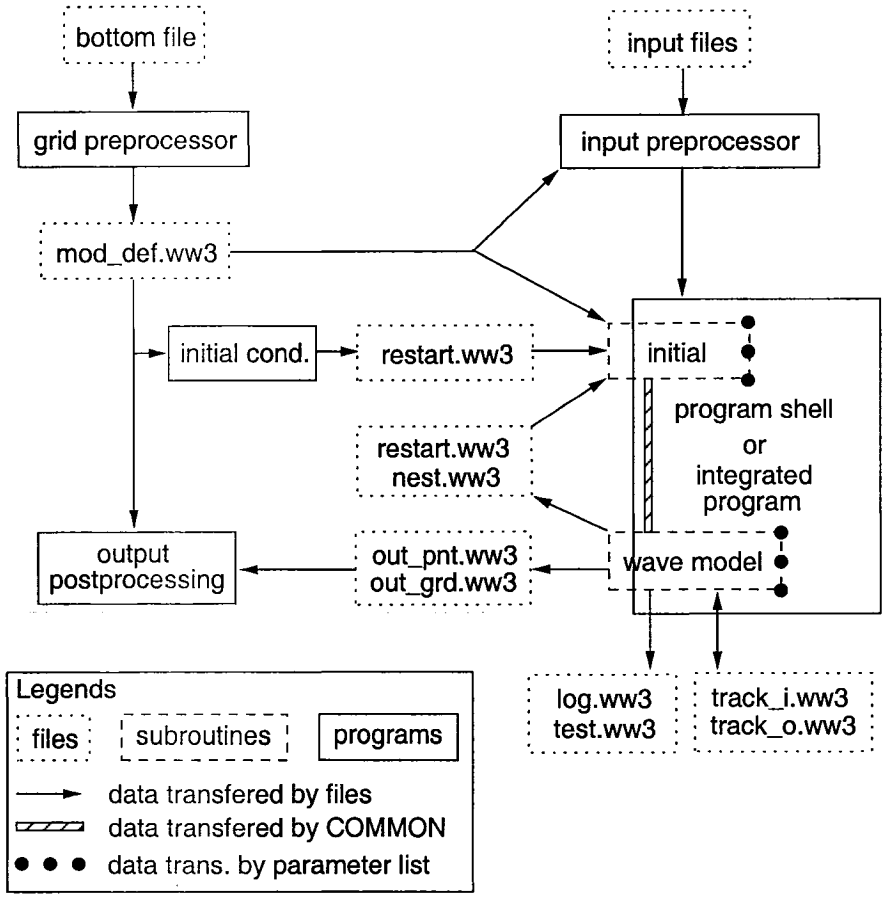


Figure 5.2: Flow chart of the wave model (after Tolman, 1999).

programs and their subprograms. Among programs, data are transferred by 3 methods, with parameter list, 'COMMON', and files respectively, as shown in Fig. 5.2.

Installing AvDI means switching the existing subprogram of nonlinear interaction with the new version. The WAVEWATCH III version 1.18 only provides **w3snl1.ftn**, a program to compute the nonlinear interaction term with DIA. However, the WW3-AvDI provides more options. It has **w3snl1.ftn**, **w3snlb.ftn**[1], and **w3snle.ftn**[2]

[1]'b' for Bedford Institute of Oceanography
[2]'e' for Exact

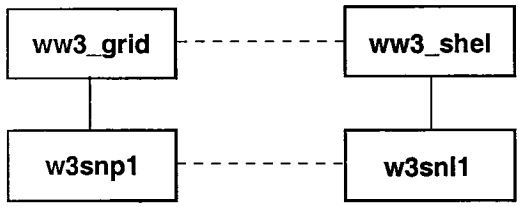


Figure 5.3: Program and subprograms for DIA.

which computes the nonlinear term with DIA, AvDI and WTR methods respectively. To choose those options, we turn on the code in the **switch** file in directory **bin**. Table 5.1 shows the code for each method.

Table 5.1: Switch of nonlinear interaction.

WAVEWATCH III v. 1.18		WW3-AvDI	
NL0	No nonlinear interactions used	NL0	No nonlinear interactions used
NL1	Discrete interaction approx. (DIA)	NL1	Discrete interaction approx. (DIA)
NLX	Experimental (user supplied)	NLB	Advanced Dominant Interaction
		NLE	Webb-Tracy-Resio method
		NLX	Experimental (user supplied)

However, the program can not run by just switching on the code. We need some modifications and new subprograms in the wave model. Modifications are needed in either the compiler or the main program, but only the main program has new subprograms. Table 5.2 and 5.3 show modified files and new files. Details for those files are displayed in Appendix E, F, and G.

The scheme for the new programs and subprograms for the new nonlinear term are presented in Figs. 5.4 and 5.5. For comparison, the scheme for DIA method is also displayed (see Fig. 5.3).

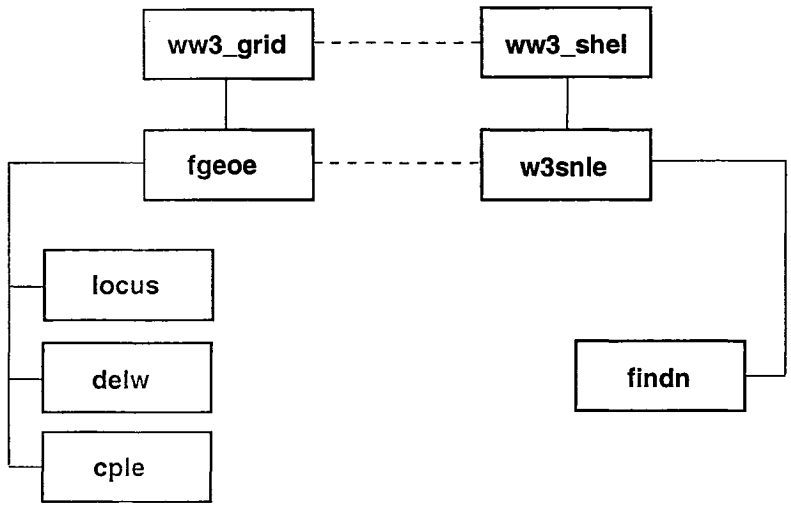


Figure 5.4: Program and subprograms for WTR.

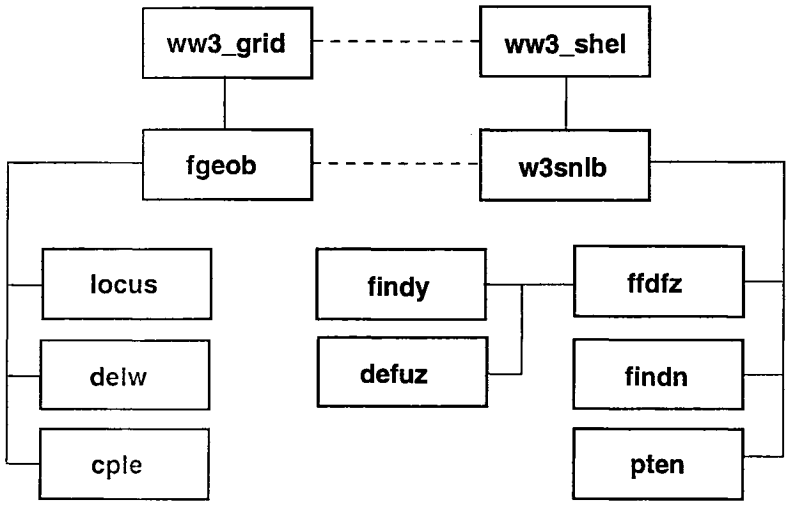


Figure 5.5: Program and subprograms for AvDI.

Table 5.2: Edited files of WW3-AvDI model.

Directory			
aux	**bin**	**ftn**	**inp**
w3adc.f	ad3	**dims.cmn**	**ww3_grid.inp**
	ad3_test	**dimx.cmn**	
	comp	**w3expo.ftn**	
	comp.gen	**w3init.ftn**	
	link	**w3iogr.ftn**	
	link.gen	**w3srce.ftn**	
	make_**makefile.prog**	**w3wave.ftn**	
	make_**makefile.subs**	**ww3_grid.ftn**	
	w3_new	**ww3_outp.ftn**	
		ww3_shel.ftn	

Table 5.3: New files of WW3-AvDI model.

Directory: **ftn**			
fzzl.cmn	**nc10.cmn**	**ffdfz.ftn**	**locus.ftn**
geob.cmn	**nl4p.cmn**	**fgeob.ftn**	**pten.ftn**
geoe.cmn	**cple.ftn**	**fgeoe.ftn**	**w3snlb.ftn**
grid.cmn	**defuz.ftn**	**findn.ftn**	**w3snle.ftn**
loci.cmn	**delw.ftn**	**findy.ftn**	

Chapter 6

Numerical Results

6.1 Prototype Model

AvDI was tested and the results are compared to those of the WTR method in this chapter. AvDI was then installed into a real wave model, in this case WAVEWATCH III (also denoted WW3). Then, the new wave model representing a combination of WAVEWATCH III and AvDI (WW3-AvDI) is tested with an ideal case and a real storm case, hurricane Juan. The testing of the WW3-AvDI model is presented in the second section of this chapter.

6.1.1 Setting the spectra grid

The spectra grid is set up on a half circle domain, with directional bins stretched from $-90°$ to $+90°$ with $10°$ resolution, for a total of 18 angular bins. There are 30 wavenumber bins. The wavenumber k is defined by $k(1) = k_0$, $k(2) = \lambda k(1)$, $k(3) = \lambda k(2)$, and so on, where λ is the incremental factor. In this thesis $\lambda = 1.21$ is used and the first wavenumber, k_0, is defined as

$$k_0 = \frac{k_{peak}}{\lambda^7}. \tag{6.1}$$

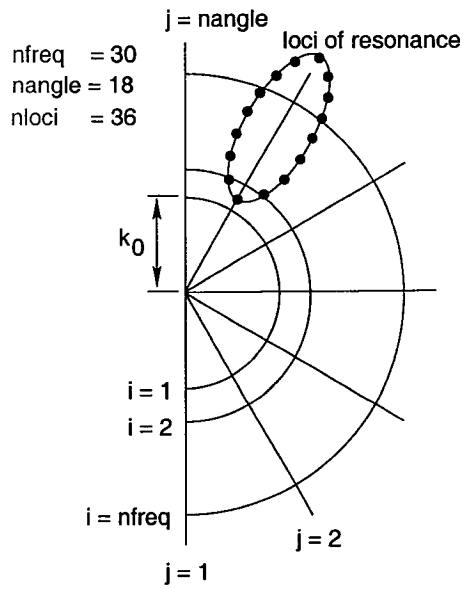


Figure 6.1: Spectra grid for experimental case.

Points of the resonance loci are also set up at 10° resolution, so that there are 36 points. Therefore, the grid has dimension: $nfreq \times nangle \times nloci = 30 \times 18 \times 36$, as shown in Fig. 6.1.

With these dimensions, the WTR method needs $30 \times 18 \times 30 \times 18 \times 36 = 10,497,600$ calculations. But AvDI will need only $30 \times 18 \times 1 \times 2 \times 18 = 19,440$ calculations. The experiment was run using 3 different methods; WTR, DTA and AvDI. Table 6.1 depicts 'do-loops' for each method and DIA's 'do-loop' for comparison. This shows that AvDI can significantly reduce the number of calculations, if compared to DTA or WTR.

6.1.2 Results and discussions

Some experiments were conducted to get results for comparison between AvDI, DTA and WTR. We also ran the DIA method. These cases assume a JONSWAP spectrum with parameters; $\alpha = 0.01$, peak spectral width parameters $\sigma_a = 0.07$, $\sigma_b = 0.09$,

Table 6.1: Do-loop for each method.

WTR	DTA	AvDI	DIA
do ik1 = 1,nfreq do ia1 = 1,nangle do ik3 = 1,nfreq do ia3 = 1,nangle do ilc = 1,nlocus ... ... end do end do end do end do end do	do ik1 = 1,nfreq do ia1 = 1,nangle do ia3 = 1,nangle do ilc = 1,nlocus ... ... end do end do end do end do	do ik1 = 1,nfreq do ia1 = 1,nangle do ia3 = 1,2 do ilc = 1,18 ... ... end do end do end do end do	do ik1 = 1,nfreq do ia1 = 1,nangle ... ... end do end do

$f_p = 0.3$ Hz, $\gamma = 1, 3, 5, 7$ and spreading factor $= f(\cos^2\theta)$, and wind speed, $U_{10} = 20$ m/s.

The computation times corresponding to each method at $t = 0$ and $t > 0$ are shown in Table 6.2 and Fig. 6.2. The program records the time that is needed to compute the basic geometry function as presented in the table. At $t > 0$, the computation time of the basic geometry function is not included. It can be seen that saving the basic geometry data can speed up the WTR method by 50%, but the same approach does not produce the same result for DTA or AvDI, because neither DTA nor AvDI is integrated along k_3. However, compared to the others, AvDI is the fastest method involving the loci of resonance. As shown in either Table 6.2 or Fig. 6.2, AvDI is only two orders of magnitude slower than DIA. This result occurs because AvDI employs two more integrations, i.e. along the directional angle of k_3 which is represented by two angles, $-20°$ and $+20°$ respectively and 18 points, representative of the resonance loci. Therefore, AvDI is 36 times slower than DIA.

One- and two- dimensional results for S_{nl4} are shown in Figs. 6.3 and 6.5-6.8 respectively. We show 1-dimensional results for all four methods. However, for 2-dimensional spectra, we show results for WTR and AvDI, as well as for DIA.

The one-dimensional comparisons in Figs. 6.3a-6.3d suggests that results from AvDI compare well with WTR results, particularly for the low frequency positive lobe. Some discrepancies are apparent for the negative lobe at higher frequency,

particularly for higher γ, but the overall level of agreement is good. On the other hand, DIA results are quite different.

Figure 6.3a suggests that although DIA has the same basic pattern as the other two formulations, DIA tends to overestimate the positive and negative transfer rates, and the transfer at higher frequencies. At $\gamma = 3.0$, the negative DIA lobe has much greater magnitude than positive lobe, or the corresponding results from WTR or AvDI. While positive transfer also occurs at higher frequency, it is much larger than the transfer suggested by WTR or AvDI results (Fig. 6.3b).

At $\gamma = 5.0$ (Fig. 6.3c), two positive DIA peaks are suggested at low frequency relative to the spectral peak. Additional positive transfer at high frequency is quite large, compared to WTR and AvDI results, the negative transfer is displaced compared to peak negative values for WTR and AvDI and is much larger in magnitude, comparatively. DIA's plot in Fig. 6.3d shows similar trends, although clearly more complicated with two positive peaks at low frequencies and positive transfer at high frequencies, and the maximum positive transfer is shifted compared to peak positive values for WTR and AvDI. Two negative peaks are also evident on the rear face of the spectral peak. Magnitudes of these peaks are greater those of WTR or AvDI.

Two-dimensional results, as shown in Figs. 6.5-6.8, demonstrate that computational results from S_{nl4} by AvDI are similar to results from WTR. Contours depicting the negative peak lobe of AvDI's results tend to be slightly wider than those of WTR's results. The positive high frequency lobe also tends to be too high in AvDI's results, especially at higher γ, compared to WTR results.

Results from DIA are presented in panels (b) in Figs. 6.5-6.8, and are quite different from WTR or AvDI, in both magnitude and contour shape. Magnitudes for DIA are always 1-2 orders different from WTR and AvDI, which are comparable. In terms of shape, at $\gamma = 1.0$, DIA has wider directional contours than either WTR or AvDI (Fig. 6.5b). With increasing γ, the differences to results from WTR and AVDI become more accentuated. At $\gamma = 3.0$ (Fig. 6.6b), DIA results suggest a negative lobe which is much more dominant than the positive lobe. Two positive peaks also

Table 6.2: Relative computational time compared to the DIA method.

Method	Time compared to DIA	
	$t = 0$	$t > 0$
Exact	12163	12163
WTR	4864	2074
DTA	633	527
AvDI	36	29

appear in the higher frequency region as shown in Fig. 6.6b.

At $\gamma = 5.0$, DIA results in four minor positive peaks and two major negative peaks, displaying a pattern that is markedly different from WTR and AvDI patterns. as shown in Fig. 6.7b. This behaviour is accentuated in Fig. 6.8b showing results when $\gamma = 7.0$. Figure 6.8b suggests that the maximum positive lobe shifts to lower frequency compared to WTR and AvDI and negative peaks are more obvious than at $\gamma = 5.0$.

From Figs. 6.3-6.8 there is overall systematic agreement between WTR and AvDI. By comparison, DIA exhibits maxima in positive and negative lobes that are larger than those of WTR or AvDI. Qualitatively, all three formulations in rough agreement for $\gamma = 1.0$ with a broad positive peak located in the region from 0.2 Hz to 0.3 Hz, and a broad negative centered on frequencies from about 0.4 to 0.5 Hz. This captures the tendency for nonlinear energy transfers to broaden the spectrum for frequencies less than the spectral peak and to tend towards a bimodal form at high frequencies as seen in many observations (Long and Resio, 2007). In the case when $\gamma = 3.0$, 5.0 and 7.0, the degree of agreement between WTR and AvDI continues to be very good throughout the range of the comparison. However, DIA exhibits broader larger positive lobes and dominant negative lobes displaced to higher frequencies (about 0.4 to 0.5 Hz). DIA results feature detailed secondary peaks not evident in WTR or AvDI. These differences in the directional distributions of AvDI and DIA will have notable differences on the modelled distributions of wave angles in a spectrum.

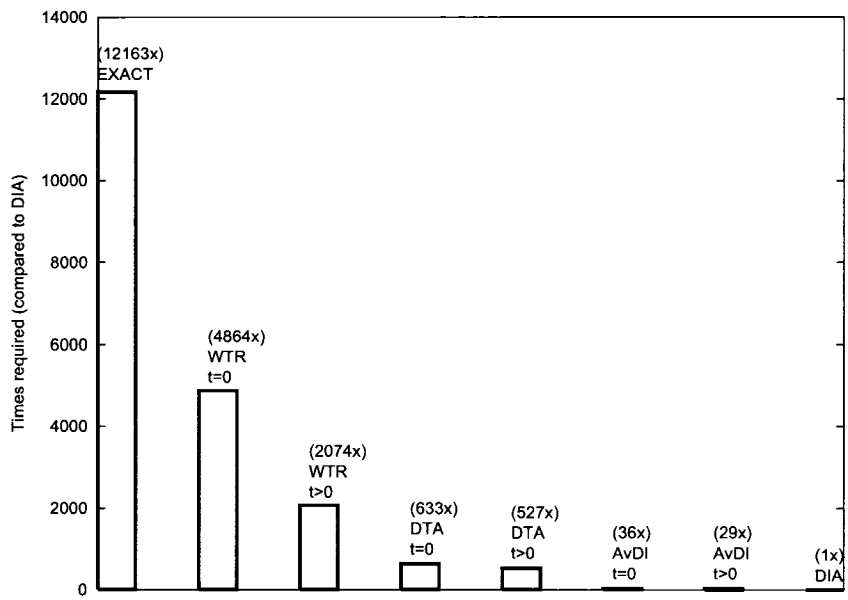
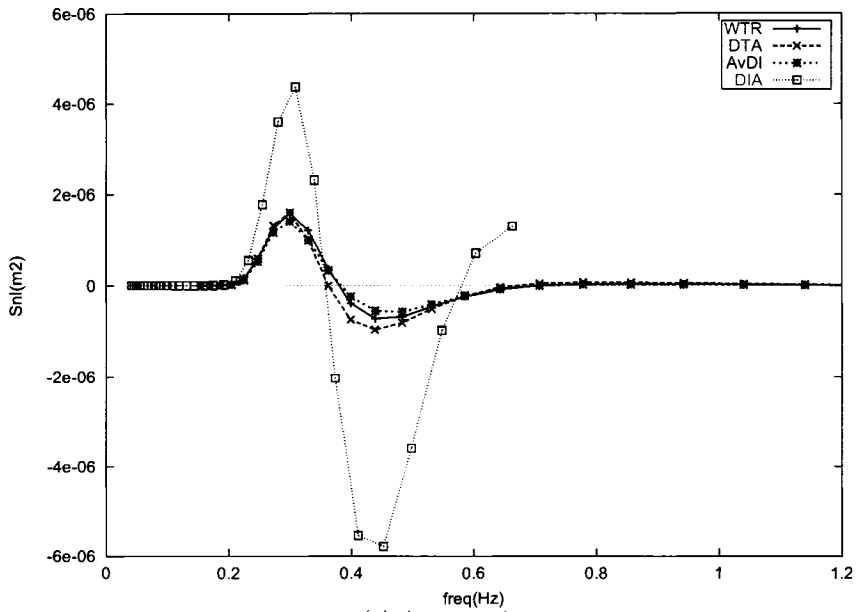
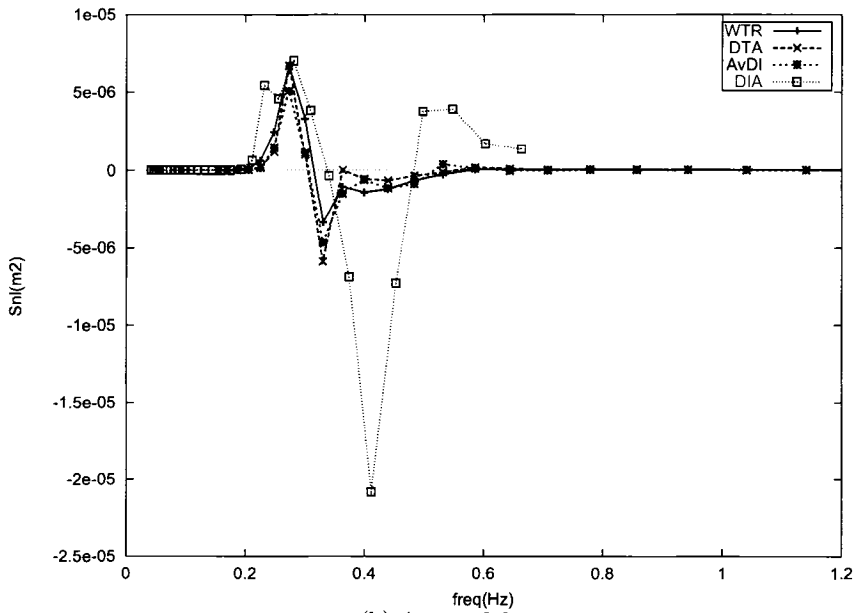


Figure 6.2: Relative computational time of for different S_{nl4} methods compared to the DIA method for a $30 \times 18 \times 36$ grid.

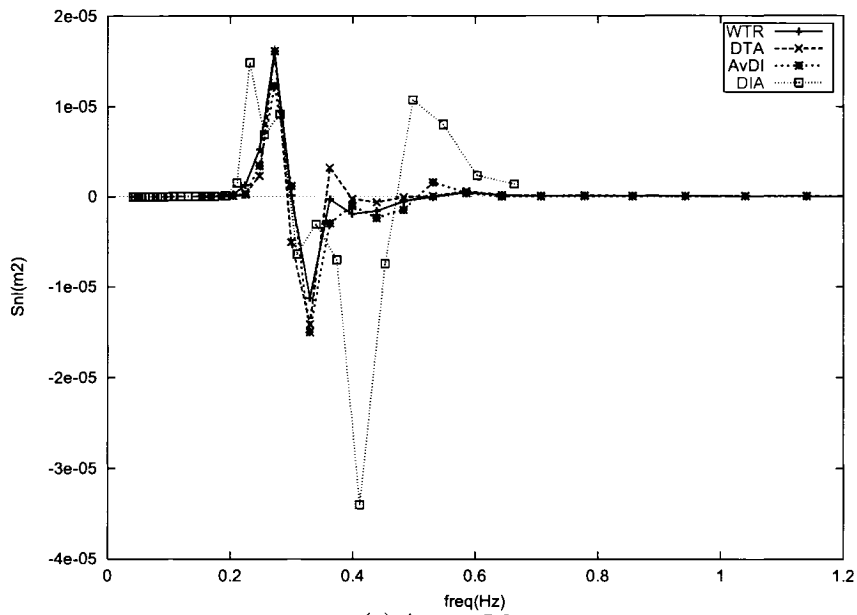


(a) At $\gamma = 1.0$.

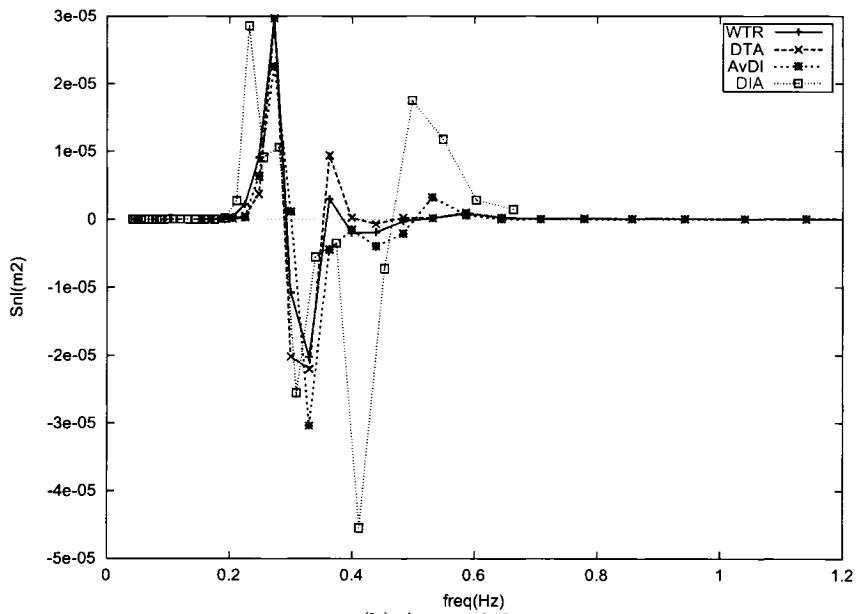


(b) At $\gamma = 3.0$.

Figure 6.3: S_{nl} comparisons in 1 dimension for JONSWAP spectrum with peakedness $\gamma =$ (a) 1.0 and (b) 3.0, respectively.

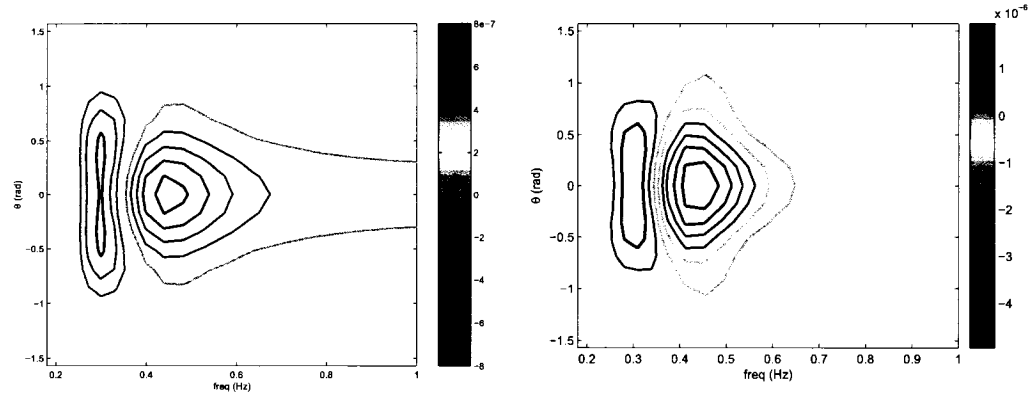


(a) At $\gamma = 5.0$.



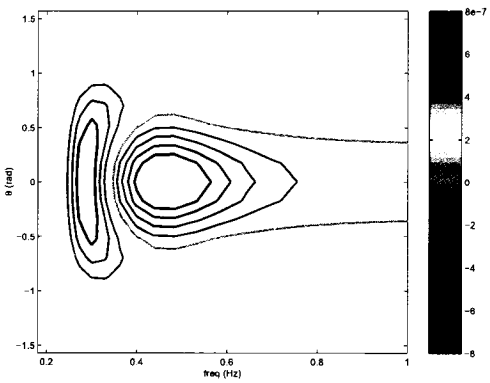
(b) At $\gamma = 7.0$.

Figure 6.4: S_{nl} comparisons in 1 dimension for JONSWAP spectrum with peakedness $\gamma =$ (a) 5.0 and (b) 7.0, respectively.



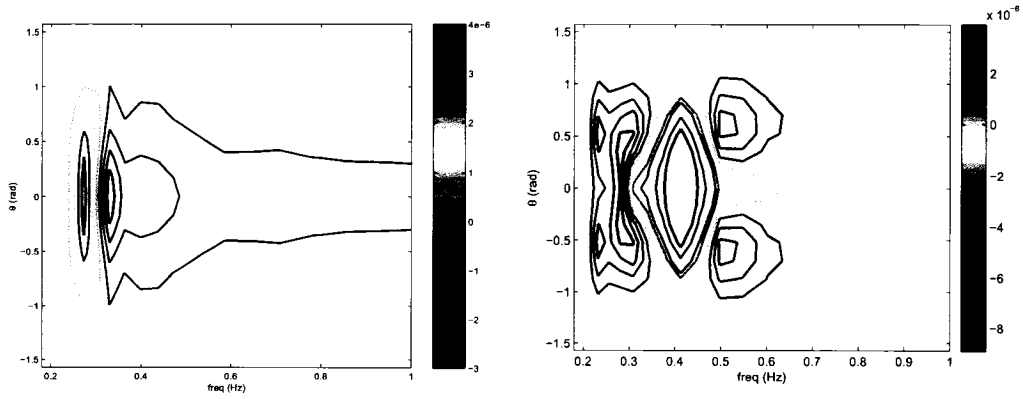
(a) WTR

(b) DIA



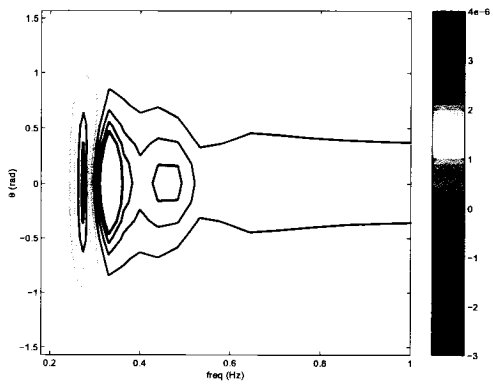
(c) AvDI

Figure 6.5: S_{nl} in 2 dimensions using a JONSWAP input spectrum with peakedness $\gamma = 1.0$, comparing (a) WTR, (b) DIA, (c) and AvDI.



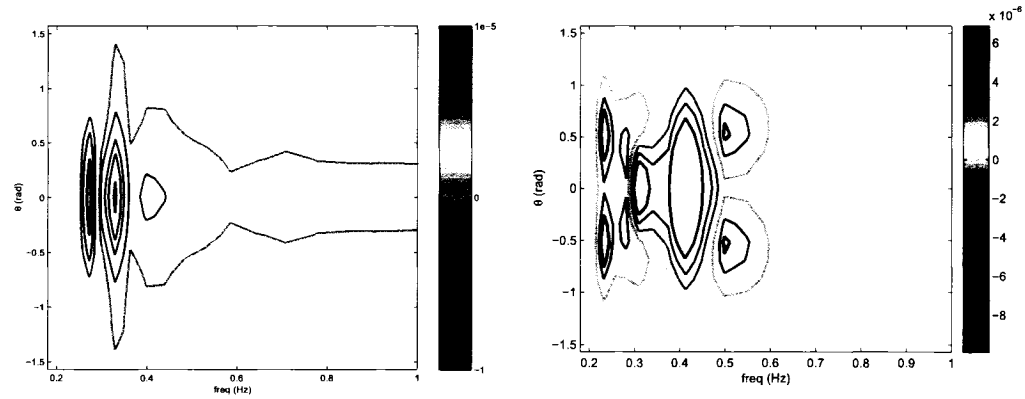
(a) WTR

(b) DIA



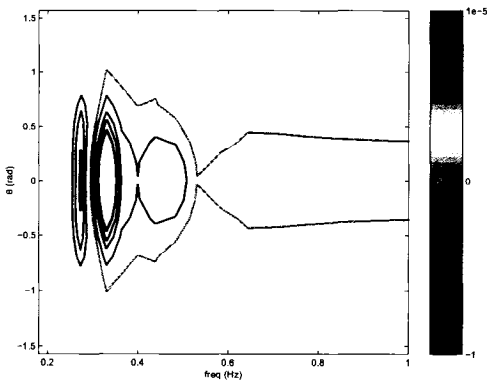
(c) AvDI

Figure 6.6: As in Fig. 6.5, S_{nl} results for $\gamma = 3.0$ comparing (a) WTR, (b) DIA, and (c) AvDI.



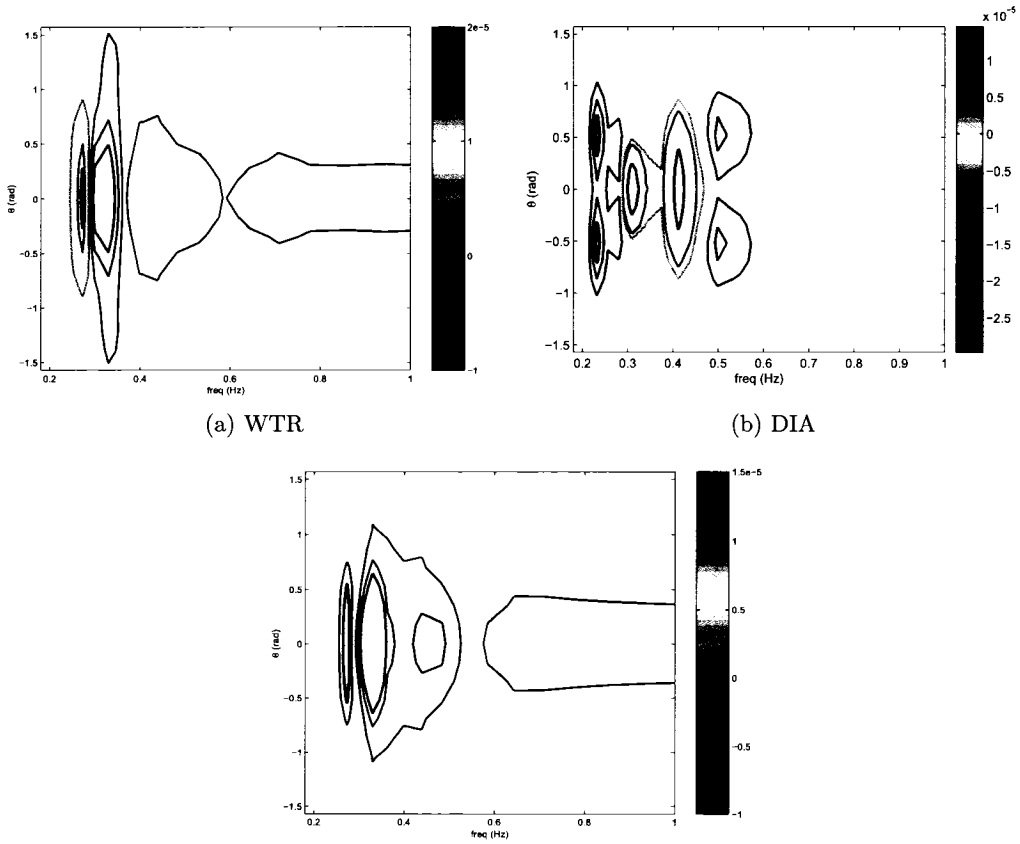
(a) WTR

(b) DIA



(c) AvDI

Figure 6.7: As in Fig. 6.5, S_{nl} results for $\gamma = 5.0$ comparing (a) WTR, (b) DIA, and (c) AvDI.



(a) WTR

(b) DIA

(c) AvDI

Figure 6.8: As in Fig. 6.5, S_{nl} results for $\gamma = 7.0$ comparing (a) WTR, (b) DIA, and (c) AvDI.

6.2 WW3-AvDI Model

WAVEWATCH III uses a semi-implicit integration scheme. This scheme requires a diagonal term to estimate the source function at each new time step, defined as

$$DT(f,\theta) = \frac{\partial S(f,\theta)}{\partial E(f,\theta)} \tag{6.2}$$

According to Van Vledder (2005) and Eq.(3.24), the diagonal term in WW3-AvDI can be computed as

$$DT(\mathbf{k}) = \frac{\partial}{\partial N}\left[2\int\oint D(N)\frac{G(\mathbf{k})}{ds}\,ds\,d\mathbf{k}_3\right] \tag{6.3}$$

$$= 2\int\oint \frac{\partial}{\partial N}\left[D(N)\right]\frac{G(\mathbf{k})}{ds}\,ds\,d\mathbf{k}_3 \tag{6.4}$$

or

$$DT(\mathbf{k}_1) = 2\int\oint [N_3(N_4 - N_2) - N_2N_4]\,\frac{G(\mathbf{k})}{ds}\,ds\,d\mathbf{k}_3 \tag{6.5}$$

$$DT(\mathbf{k}_3) = 2\int\oint [N_1(N_4 - N_2) + N_2N_4]\,\frac{G(\mathbf{k})}{ds}\,ds\,d\mathbf{k}_3\,. \tag{6.6}$$

The spectra grid is set up in a circle domain, with directional bins stretched from $-180°$ to $+180°$ with $10°$ resolution, implying 36 angular bins. There are 30 frequency bins with the first frequency, $k_0 = 0.0418$ Hz, and the frequency increment factor is 1.1. The wavenumber increment factor is set as $\lambda = 1.1^2 = 1.21$. The resonance loci are also set at $10°$ resolution, so that there are 36 points. Therefore, the grid has dimension: $nfreq \times nangle \times nloci = 30 \times 36 \times 36$.

Table 6.3 shows the value of F_d used in WW3-AvDI. Because a different computational domain is used for JONSWAP tests, compared to the full wave model simulations of hurricane Juan, an adjustment must be made before installing AvDI into WW3, for the latter tests. In general, WW3 simulates waves on a $360°$ directional distribution, whereas the prototype tests used of subsection 6.1.1 only consider integration over a half-circle, from $-90°$ to $+90°$. A correction to F_d is needed to

Table 6.3: F_d used in WW3-AvDI.

F_d		$slope_g$			
		0.0069	0.5124	0.5782	0.6139
$slope_s$	0.1728	12.5000	9.0761	9.5266	10.1712
	0.3403	10.3665	8.4495	9.1633	10.0000
	0.9615	7.0642	7.7174	9.2124	10.5817

accommodate the WW3 360° domain. From numerical experiments, it is found that $F_{d(360°)} = 0.375 F_{d(180°)}$.

As shown in Fig. 6.3, AvDI also has a small amount of numerical disturbance at higher frequencies. Therefore, a stabilizer to remove this disturbances is built into WW3-AvDI. After some adjustments, two tests were conducted, one representing an idealized case and the other representing a real storm. SWAMP Case II was chosen for the ideal case, where the wind blows uniformly over time and distance. Hurricane Juan was chosen as the real storm case.

6.2.1 SWAMP case

The Case II of the SWAMP Group (1985), *Fetch- and Duration-Limited Growth* test, was conducted to test WW3-AvDI performance for an ideal situation. Figure 6.9 features the geometry of the wind field and boundary conditions. A homogeneous wind field blows perpendicular to offshore constantly. The wind speed, U_{10}, is 20 m/s and it is infinite in the lateral and downstream directions. The initial wave energy at time $t = 0$ is zero and remains zero for $t > 0$ at the coastline $y = 0$.

Different formulations of source terms were employed to run the tests. The model is classified according to the source terms. A model using the WAM3 source terms for wind input and dissipation and DIA for the nonlinear interaction term is called WAM3-DIA. WAM3-AvDI uses WAM3 for wind input and dissipation and AvDI for the nonlinear term. TC-AvDI uses Tolman and Chalikov's wind input and dissipation and AvDI for the nonlinear term.

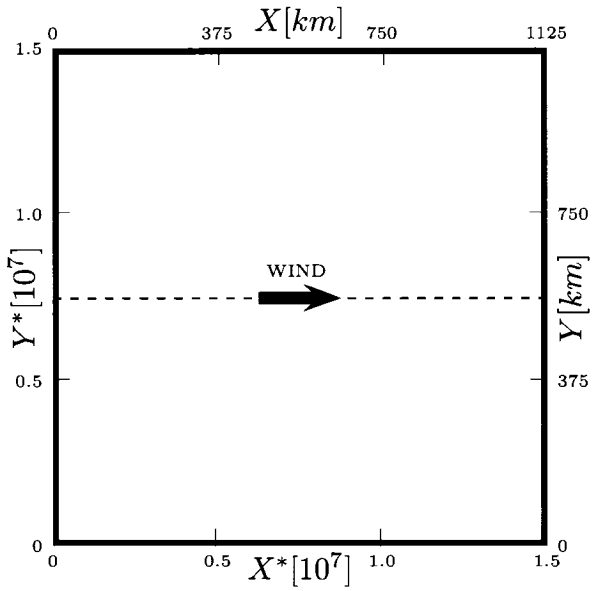


Figure 6.9: Wind field geometry (after the SWAMP Group, 1985).

Following are the **switch** selections that must be turned on before running the model

- **WAM3-DIA**

  DUM LRB4 SHRD GRTST SPTST PR2 **ST1 NL1** BT1 WND1 CUR1 SEED BPI0 BPO1 PNT5 O1 O2 O2a O3 O4 O5 O6 O7

- **WAM3-AvDI**

  DUM LRB4 SHRD GRTST SPTST PR2 **ST1 NLB** BT1 WND1 CUR1 SEED BPI0 BPO1 PNT5 O1 O2 O2a O3 O4 O5 O6 O7

- **TC-AvDI**

  DUM LRB4 SHRD GRTST SPTST PR2 **ST2 NLB** BT1 WND1 CUR1 SEED BPI0 BPO1 PNT5 **STAB2 XW0 O0** O1 O2 O2a O3 O4 O5 O6 O7.

The test results, showing the total wave energy and peak frequency as a function of fetch- and duration-limited growth, are shown in Figs. 6.10 through 6.13.

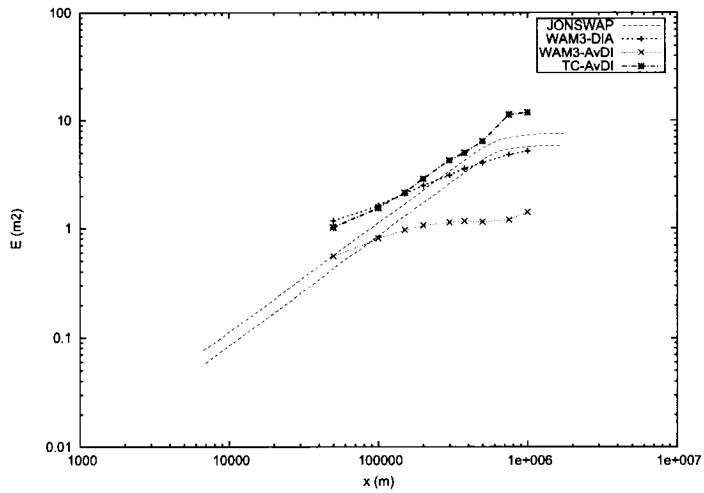


Figure 6.10: The total energy curves as function of fetch-limited growth.

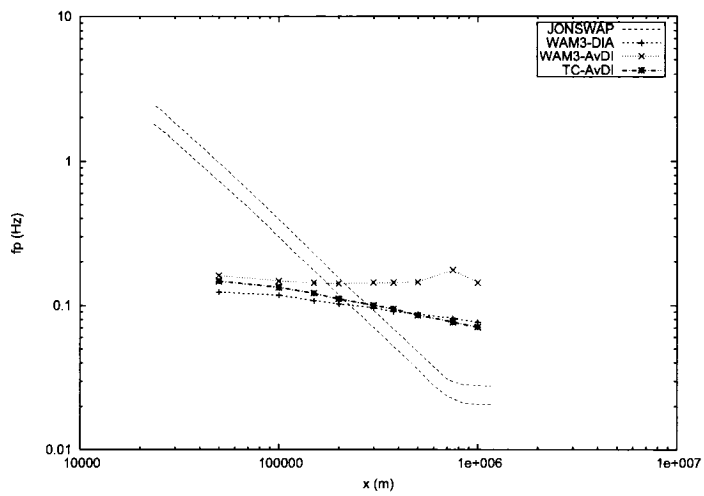


Figure 6.11: The peak frequency curves as function of fetch-limited growth.

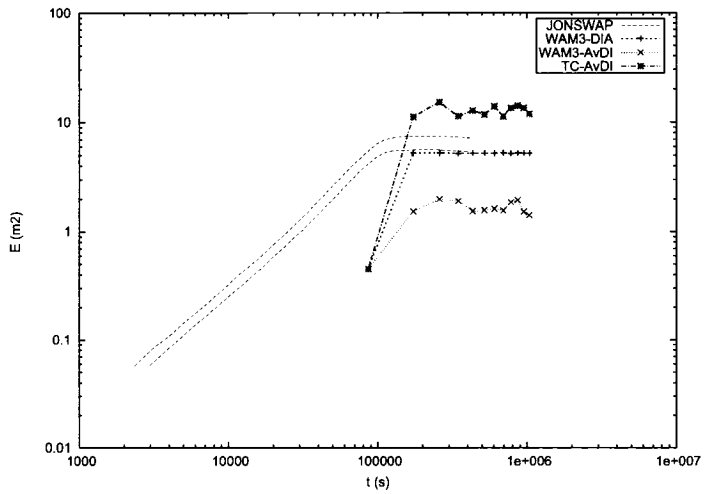


Figure 6.12: The total energy curves as function of duration-limited growth.

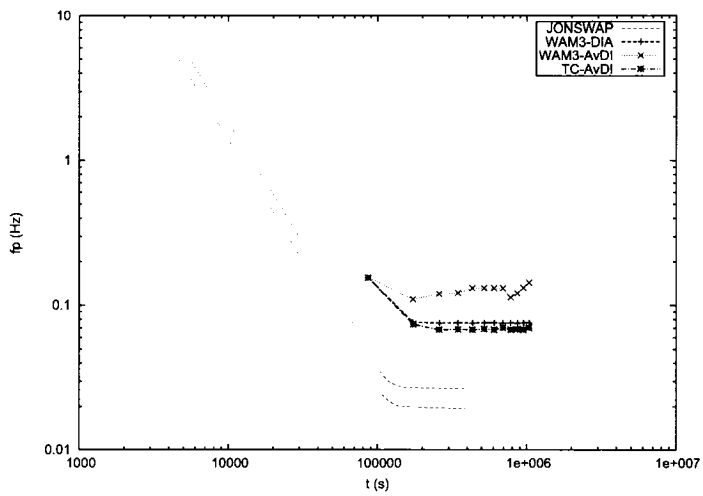


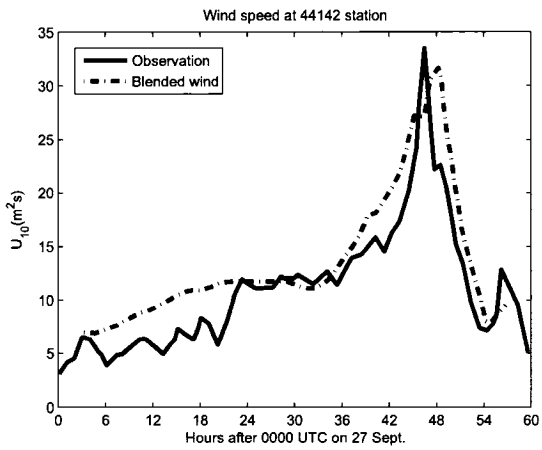
Figure 6.13: The peak frequency curves as function of duration-limited growth.

6.2.2 Hurricane Juan

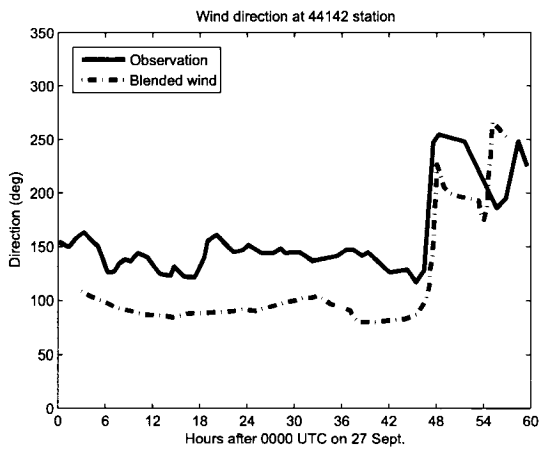
A detailed description of hurricane Juan is given by Fogarty et al. (2006) and Fogarty (2003) in the Canadian Hurricane Center website. Juan reached hurricane strength on 26 Sept. near Bermuda, and moved northwestward, as a subtropical ridge to the northeast of its location extended to the west. It attained maximum winds of 90 knots at 1800 UTC on 27 Sept., and turned northward towards Nova Scotia, with increasing propagation speed. By 1800 UTC on 28 Sept., Juan was north of the Gulf Stream, and its intensity began to weaken due to the cooler continental shelf waters. Because of its accelerating translational speed, Juan quickly passed over these cooler waters and made landfall near Halifax (0300 UTC on 29), with sustained winds of 85 knots. A feature of Juan's development is the phenomenal acceleration of its translation speed, increasing dramatically from 2.28 ms^{-1} at 1200 UTC on 27 Sept. to 20 ms^{-1} at 1200 UTC on 29 Sept.

Xu et al. (2006) used interpolation methods to blend observed hurricane winds with numerical weather prediction (NWP) model winds to accurately represent the wind fields, as shown on Fig. 6.14. These data were used in this test as a wind input field. The results from WW3-AvDI were compared to the default model in which the DIA method was employed to compute the S_{nl4} term (called WW3-DIA). Results are also compared to observed *in situ* data, obtained from the Canadian Meteorological Service of Canada (MSC) operational buoys (buoys 44142 and 44137) off Nova Scotia and a directional waverider (DWR) that is located at the mouth of Lunenburg Bay. The buoys 44142, 44137, and DWR are located in water depths of 1300 m, 4500 m, and 29 m respectively. Figure 6.16 shows the hurricane track and buoys locations. It also shows the resolution grid, at 15′ devision. Its domain is from 40°W to 75°W and 20°N to 65°N.

The starting frequency f_0 was $f_0 = 0.04118$ following Xu et al. (2006). This number was also used in this test. The frequency increment factor is 1.1, the number of frequencies is 30 and the number of directions is 36 (10° resolution). The initial spectrum is $ITYPE = 3$ on **ww3_strt.inp**, in which the local spectrum is calculated

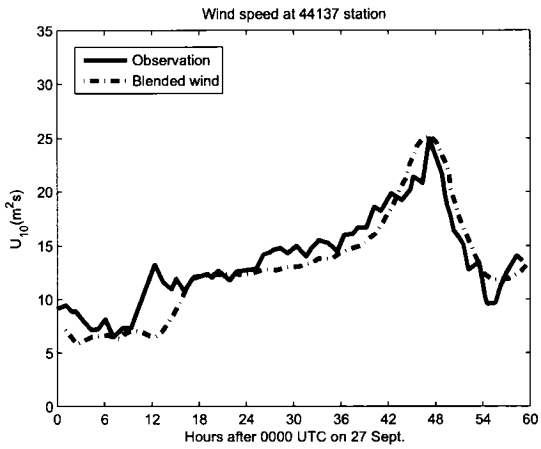


(a)

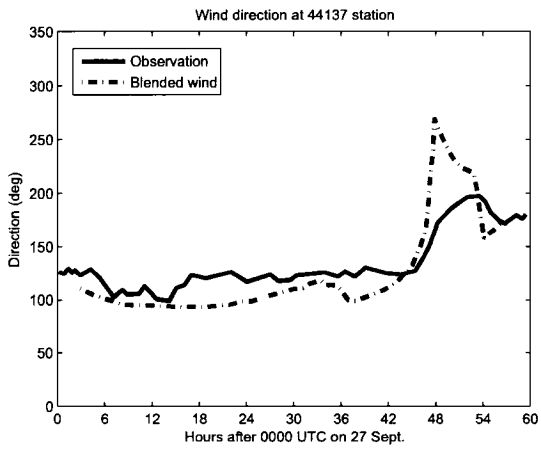


(b)

Figure 6.14: Comparison between interpolated blended winds and observations (from Xu, F. et al., 2006), (a) wind speed at 44142 station and (b) wind direction at 44142 station.



(c)



(d)

Figure 6.15: Comparison between interpolated blended winds and observations (from Xu, F. et al., 2006), (a) wind speed at 44137 station and (b) wind direction at 44137 station.

using local wind velocity. Before compiling, the **switch** was set up as

DUM LRB4 SHRD GRTST SPTST PR2 ST2 **NLB** BT1 WND1 RWND CUR1 SEED
BPI0 BPO1 PNT5 XW0 O0 O1 O2 O2a O3 O4 O5 O6 O7

After setting up the switch and inputs, the model was compiled and run. Mean wave parameters (significant wave height and peak periode) on three buoys are presented on Figs. 6.17 through 6.18, and the one- and two-dimensional spectra are plotted on Figs. 6.19 through 6.22.

6.2.3 Discussions for both cases

a. Fetch- and duration-limited case

Figures 6.10 through 6.13 present plots of fetch- and duration-limited growth. The wave energy and peak frequency are plotted against distance and time respectively and compared to result from JONSWAP observations. The white band on those figures represents the JONSWAP growth curve ±5% (the SWAMP Group, 1985). All models show the same trend, energy increases as distance or time increases, then asymptotes to a constant after some long fetch or time. However, the peak frequency decreases as distance or time increases, then reaches a constant after a limiting fetch or time is achieved. However, the slopes are different from our model results, compared to those of JONSWAP results.

AvDI results are better with Tolman-Chalikov's source terms than with WAM3 source terms. As shown in Figures 6.10 through 6.13, TC-AvDI graphs present a reasonably similar trend compared to either JONSWAP or WAM-DIA results. It is also shown that WAM3 source terms and AvDI are not a good combination. This is a result of AvDI's formulation, which needs to have specifically tuned source terms for S_{in} and S_{ds}, as is done in all modern wave models.

b. Hurricane Juan case

The Hs and peak period (Tp) comparisons (Figs. 6.17-6.18) suggest that WW3-AvDI and WW3-DIA are capable of simulating Juan-generated waves reasonably well, and that results from both models are somewhat comparable, using the interpolated winds from Xu et al. (2006). They suggest that these interpolated winds lead to better Hs and Tp estimates than the COAMPS winds, noting that the latter tend to produce underestimates in Hs and Tp values.

The comparisons in Figs. 6.17-6.18 show biases. At buoy 44137, before about 0400 UTC on 28 Sept., simulated Hs estimates agree well with observations, while simulated Tp values are biased low. After 2200 UTC on 28 Sept., simulated Tp values agree well with observations, especially during the storm's peak, but simulated values from both models over-estimate observed Hs values. At buoy 44142, simulated Hs values from WW3-DIA under-estimate observed data during and after the highest winds. Xu et al. (2006) suggest that this DIA result may be due to over-estimates in the sea surface roughness for high winds, as well as contributions from strong swell and current components, resulting in reduced effective winds in the region of these buoys. However, as shown by the WW3-AvDI result in Fig. 6.17b, part of the bias is due to the formulation for nonlinear wave-wave interactions: under-estimate from DIA is partially reduced by AvDI.

Observed 1D wave spectra data were collected at buoys 44137 and 44142, and observed 2D wave spectra data, at the DWR. Figures 6.19a-6.19f compare the simulated 1D spectra with observations at buoys 44137 and 44142, at the storm's peak intensity, three hours before the peak wave, and three hours after the peak wave. Overall, the simulated 1D spectra from WW3-DIA achieve a better match to observations than those resulting from WW3-AvDI. This is suggested over the entire frequency range, before, during, and after the highest winds occurrences, under wind wave-dominated as well as swell- dominated conditions. This is illustrated at 44142 as well as 44137; the simulated 1D spectral peak is improved by 200% by WW3-DIA during peak (swell) waves (0020 UTC on 29 Sept.), compared to WW3-AvDI.

Figure 6.22 compares the 1D spectrum from the models to observed data from the DWR. At this location the results from WW3-AvDI appear similar to the observed data, and WW3-DIA gives serious underestimates in peak values. Lags are evident in the timing of the peak, reflecting a lag in the simulated wind fields. A low bias in the WW3-AvDI results is present at about 0.1 Hz, compared to WW3-DIA results and the observed data.

Figure 6.23 gives simulated 2D wave spectra comparisons with the DWR observations at the time of maximum waves (0411 UTC on 29 September). At this time, the spectra are relatively narrow in directional and frequency range (Fig. 6.23a), whereas after this point, swell is dominant. To some extent the simulated 2D spectrum generated by WW3-DIA suggests wider directional and frequency range distributions, and does not capture the height of the peak, or the extent of the narrowing of the observed spectrum. By comparison, WW3-AvDI obtains a narrower higher spectrum that is qualitatively closer to characteristics of the observed data.

It is not surprising that results show some good agreements, and also some disagreements, with observation data because the wave model is not been tuned for the new method. Some tuning or calibration of variables would improve results. However, our achievement is that the AvDI method makes the Boltzmann integral applicable for a practical wave model.

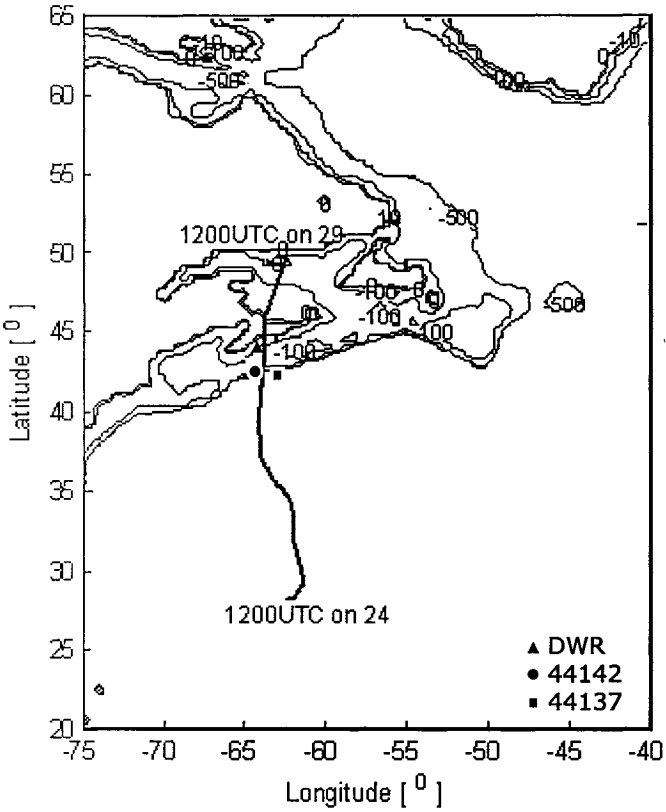
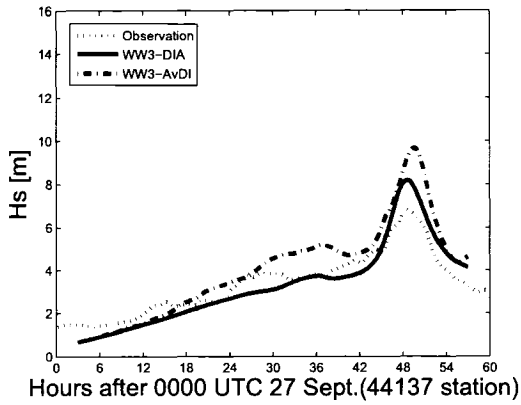
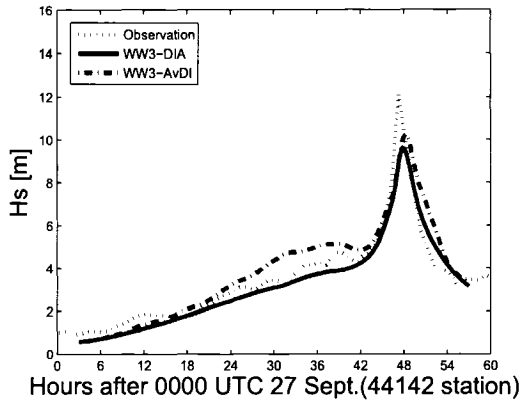


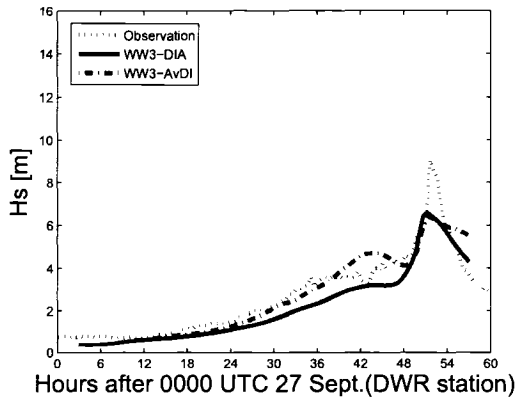
Figure 6.16: Grid domain and storm track of hurricane Juan: WW3 grid (15′). Observations are at open ocean buoys 44142 (64.02°W, 42.5°N) and 44137 (62.0°W, 42.26°N) located in 1300 m, and 4500 m depth water, respectively, and at nearshore the DWR (64.18°W, 44.24°N) in 29 m depth water.



(a) Buoy 44137.

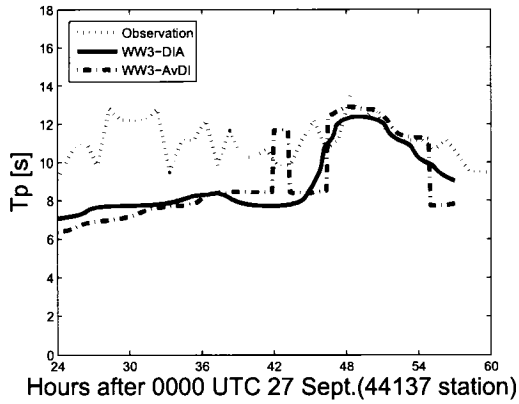


(b) Buoy 44142.

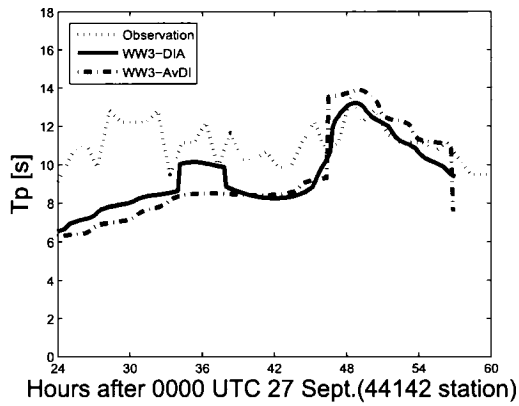


(c) Buoy DWR.

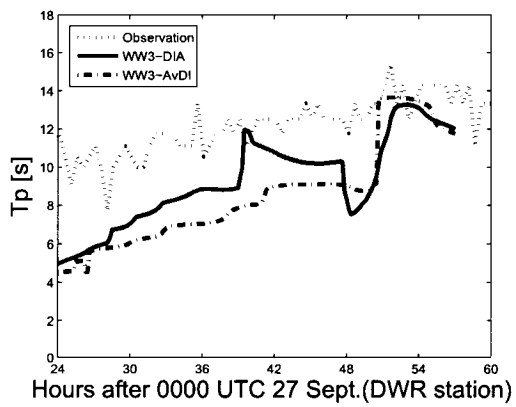
Figure 6.17: Comparison of observed and simulated estimates for Hs at different locations.



(a) Buoy 44137.

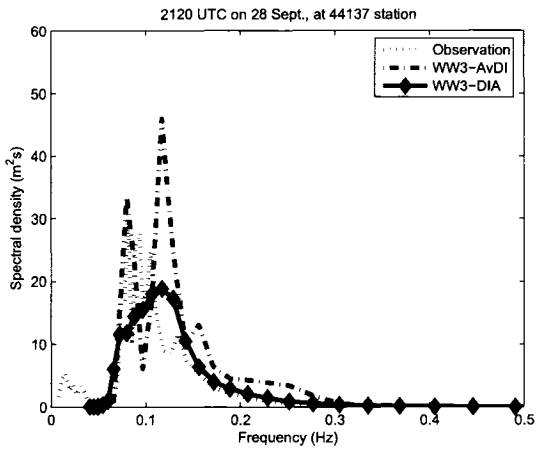


(b) Buoy 44142.

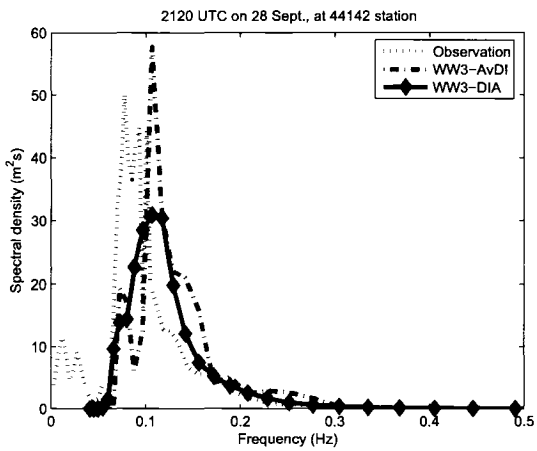


(c) Buoy DWR.

Figure 6.18: Comparison of observed and simulated estimates for Tp at different locations.

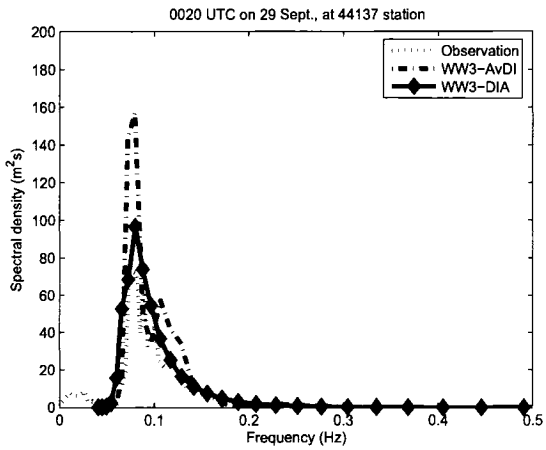


(a)

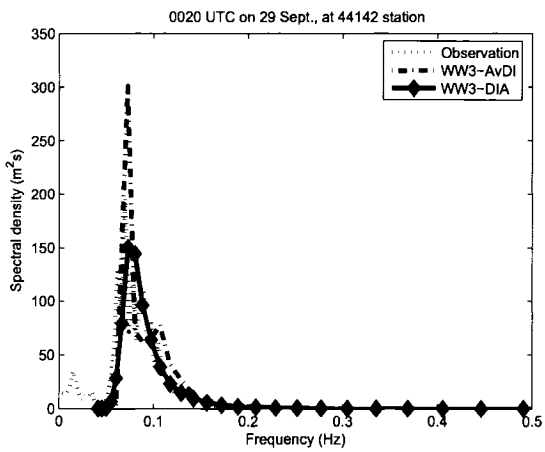


(b)

Figure 6.19: Comparison of one-dimensional spectra between observations and WW3 simulations using blended winds, at 2120 UTC on 28 Sept., at buoys (a) 44137 and (b) 44142.

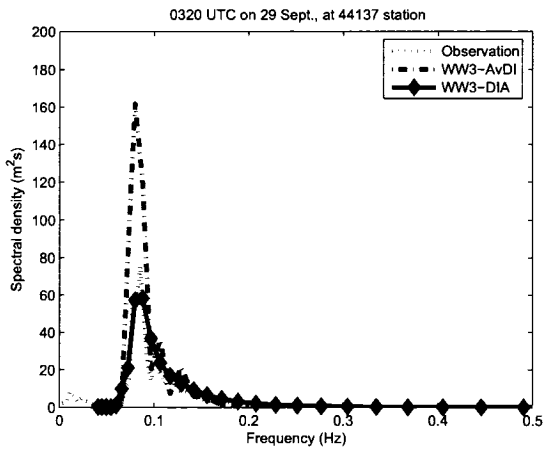


(a)

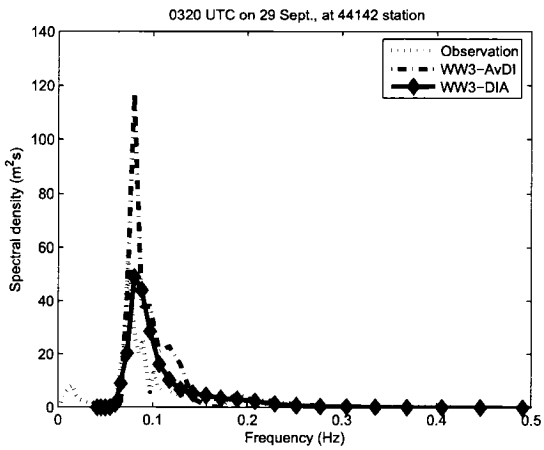


(b)

Figure 6.20: Comparison of one-dimensional spectra between observations and WW3 simulations using blended winds, at 0020 UTC on 29 Sept., at buoys (a) 44137 and (b) 44142.



(a)



(b)

Figure 6.21: Comparison of one-dimensional spectra between observations and WW3 simulations using blended winds, at 0320 UTC on 29 Sept., at buoys (a) 44137 and (b) 44142.

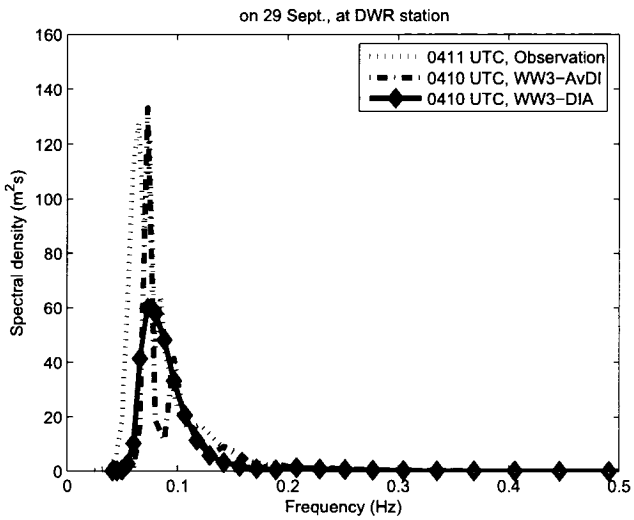
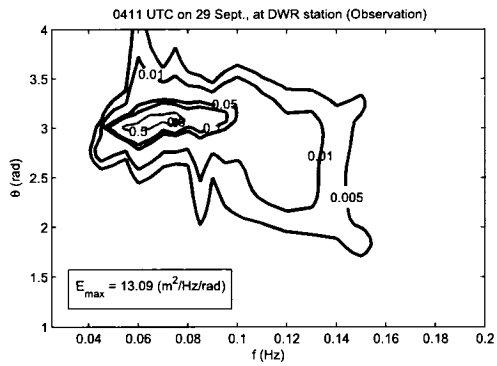
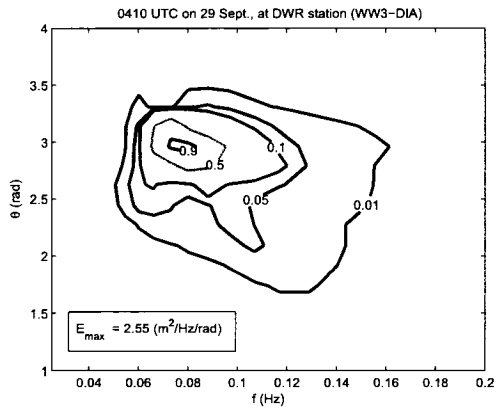


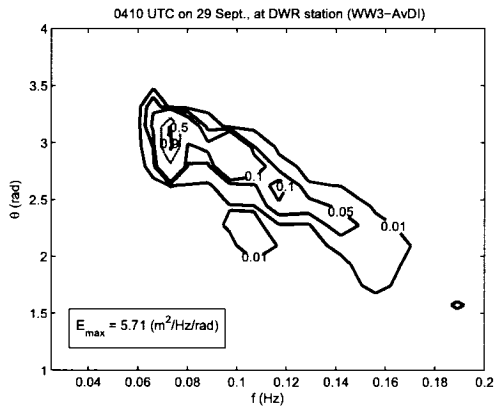
Figure 6.22: Comparison of one-dimensional spectra between observations and WW3 simulations using blended winds at DWR location, at the time of maximal wave energy.



(a)



(b)



(c)

Figure 6.23: Comparison of two-dimensional spectra at the DWR location, at the time of maximal wave energy, showing (a) DWR observations, (b) WW3-DIA simulation, and (c) WW3-AvDI simulation. Contours indicate fraction of E_{max}. Contours values are: 0.005, 0.01, 0.05, 0.10, 0.50, and 0.9. Direction is in the nautical convention.

Chapter 7

Conclusions and Recommendations

From this study it can be shown that the nonlinear transfer due to 4-wave interactions (quadruplet) can be approximated by the dominant transfer related to selected wavenumbers and a scaling factor F_d. The latter is shown to follow a well-defined variation, depending on wave maturity, as specified in terms of wave peakedness γ and the spectral spreading function.

For a variety of JONSWAP input spectra, it can be seen that this advanced dominant interaction (AvDI) formulation compares well with the full integration of the nonlinear transfer due to wave-wave interactions. It is faster than DTA and is competitive with DIA. In actual computations, the WTR method needs five loops to compute the entire spectral grid, whereas the DTA program requires only four loops. AvDI also requires four loops but at the same time reduces domain integrations; therefore AvDI is more efficient than DTA.

Table 6.2 and Figure 6.2 show that the computation time can be improved by setting up the spectra grid and using the geometrical scaling similarity property. The geometry function is not a function of time, and so we can retain its value for all the successive computations. As shown in Table 6.2, we do not need to compute the $G(\mathbf{k})$ again at $t > 0$. Reducing the loop and domain integration results in a relative speed up the AvDI method, and reduces the requirement on computational memory because fewer $G(\mathbf{k})$ basic points are needed.

For the experimental case, AvDI results show a good agreement with WTR, both in 2-dimensional and 1-dimensional S_{nl4} cases. Although two orders of magnitude slower than DIA, this prototype AvDI can be readily applied for practical operational wave simulations and forecasts.

In fetch- and duration-limited cases, it is shown that AvDI works well with Tolman and Chalikov (1996)'s wind input and wave-dissipation source terms. The combination between AvDI and wind input and wave-dissipation source terms from WAM plots gives different patterns, compared to the JONSWAP observational swath and the results from the Tolman and Chalikov (1996) source terms. However, it is not surprising that the results are different because the wave model source terms have not been tuned for the new method (AvDI).

Comparisons with the standard formulation for WW3, using the DIA formulation for nonlinear wave-wave interactions, show several results. With respect to JONSWAP input spectra, and computations from an accurate method as represented by the WTR formulation, AvDI is much more accurate than DIA, in terms of distributions of the nonlinear transfer, as well as height of the positive and negative lobes of the transfer peaks.

In terms of real data comparisons from hurricane Juan, we suggest that WW3-AvDI is competitive with WW3-DIA in estimating peak periods Tp and significant wave height Hs values at the peak of the storm. At buoy 44137, both models tend to over-estimate peak Hs values, while under-estimating peak Hs values at 44142 and the DWR. Both models tend to under-estimate Tp before the arrival of the storm at the three buoys, and then to simulate the observed peak values relatively well.

It is more difficult to simulate 1D and 2D spectra, from open ocean conditions. Our results suggest that although our new formulation, WW3-AvDI, can qualitatively simulate observed 1D and 2D wave spectra at the peak of hurricane Juan, it appears to have a tendency to over-predict the 1D spectral peak, in some instances, compared to WW3-DIA. In other cases, WW3-DIA has a tendency to under-predict the peaks.

We show that the new formulation has a sufficiently fast execution time and can

be implemented in an operational wave forecast model. For a particular grid that has been shown, the estimated run time is about 135x faster than WTR, and 36x slower than DIA. Moreover, compared to WTR, much less computational memory is required, because fewer basic wave-wave interaction geometry points are needed. However, as mentioned earlier, AvDI results compare well with WTR results, in both 1D and 2D tests. Therefore, as a prototype, AvDI can be readily applied for practical wave simulations and forecasts. Additional research needs to consider appropriate tuning of source terms for wind input and wave dissipation. In fact, we recommend that the wave model should be properly tuned to accommodate the new methodology for nonlinear term computation (AvDI method) before using this model.

Another approach is to recognize that DIA probably gives a good approximation for developed waves in the open sea but is relative poor in spectral wave approximations for developing wave situations. Therefore, a possible recommendation for further development is to combine two methods, i.e. AvDI and DIA. In this scenario, AvDI would be used to compute the nonlinear wave-wave interactions during young sea conditions, and DIA would be used for developed seas. A problem with this approach is that the real ocean is never so homogeneous as to have all fully developed or all growing wind-wave spectra. For rapidly developing storms, the waves are always growing and developing, throughout the domain of the storm.

AvDI method is fast but still not practical compared to DIA. Further research needs to look to additional methods to optimize the Boltzmann integral and yet to retain the accuracy achieved by the AvDI formulation.

Bibliography

[1] Bascom, W., 1964, Waves and Beaches, The Dynamics of the Ocean Surface, Anchor Books, Doubleday & Co. Inc., New York.

[2] Booij, N., Ris, R.C., and Holthuijsen, L.H., 1999, *A third-generation wave model for coastal regions, part I, model description and validation*, J. Geoph. Research, C4, 104, 7649-7666.

[3] Bretschneider, C.I., 1969, Topics in Ocean Engineering, Volume 1, Gulf Publishing Company, Houston Texas.

[4] Brooke, John., 2003, Wave Energy Conversion, Elsevier Ltd., Oxford.

[5] Brown, J., Colling, A., Park, D., Phillips, J., Rothery, D., and Wright, J., 1989, Waves, Tides and Shallow-Water Processes, Pergamon Press Plc., Oxford.

[6] Collins, J.I., Chiang, W., and Wu, F., 1981, *Refraction of directional spectra*, Proceedings, Directional Wave Spectra Applications, ASCE, New York, N.Y., pp 251-266.

[7] Earle, M.D., and Bishop, J.M., 1984, A Practical Guide to Ocean Wave Measurement and Analysis, Endeco, Inc., Marion, M.A., USA.

[8] Fogarty, T. C., Greatbatch, J. R. and Ritchie H., 2006, *The role of anomalously warm sea surface temperatures on the intensity of Hurricane Juan (2003) during its approach to Nova Scotia.* In press in Monthly Weather Rev.

[9] Fogarty, T. C., 2003. Available at http://www.novaweather.net/Hurricane_Juan.html [2006, 6 November].

[10] Hashimoto, N., Haagsma, I.J.G., and Holthuijsen, L.H., 2002, *Four-wave interactions in SWAN*, Proc. 28th Int. Conf. on Coastal Eng., Cardiff, pp. 392-404.

[11] Hasselmann, K., 1962, *On the non-linear energy transfer in gravity-wave spectrum. Part I: General theory*, J. Fluid Mech., 12, 481-500.

[12] Hasselmann, S., and Hasselmann, K., 1981, A Symmetrical Method of Computing the Nonlinear Transfer in a Gravity Wave Spectrum, Max Planck Inst. Tech. Report, Hamburg.

[13] Hasselmann, S., and Hasselmann, K., 1985, *Computations and parameterizations of the nonlinear energy transfer in a gravity-wave spectrum. Part I: A new method for efficient computations of the exact nonlinear transfer integral*, J. Physical Oceanography, 15, 1369-1377.

[14] Hasselmann, S., Hasselmann, K., Allender, J.H., and Barnett, T.P., 1985, *Computations and parameterizations of the nonlinear energy transfer in a gravity-wave spectrum. Part II: Parameterizations of the nonlinear energy transfer for application in wave models*, J. Physical Oceanography, 13, 1378-1391.

[15] Jackson, J. D., 1962, Classical Electrodynamic, Wiley, New York.

[16] Kinsman, B., 1965, Wind Waves: Their Generation and Propagation on the Ocean Surface, Prentice-Hall, Inc., Englewood Cliffs, N.J.

[17] Komar, P.D., 1976, Beach Processes and Sedimentation, Prentice-Hall, Inc., Englewood Cliffs, N.J.

[18] Komen, G.J., Cavaleri, L., Donelan, M., Hasselmann, K., Hasselmann, S., and Janssen, P.A.E.M., 1994, Dynamics and Modelling of Ocean Waves, Cambridge University Press, England, UK.

[19] Kosko, B., 1993, Fuzzy Thinking: The New Science of Fuzzy Logic, Hyperion, New York.

[20] Krasnopolsky, V.M., Chalikov, D.V., and Tolman, H.L., 2002, *A neural network techique to improve computational efficiency of numerical oceanic models*, Ocean Modelling 4, 363-383.

[21] Krommer, A.R., and Ueberhuber, C.W., 1994, Numerical Integration on Advanced Computer Systems, Springer-Verlag, Berlin.

[22] Long, C.E., and Resio, D.T., 2007, *Wind wave spectral observations in Currituck Sound, North Carolina*, J. Geophys. Res, **112**, doi: 1029/2006JC003835.

[23] Longuet-Higgins, M.S., 1952, *On the statistical distribution of the heights of sea-waves*, Journal of Marine Research, 11, 245-266.

[24] Longuet-Higgins, M.S., Cartwright, D.E., and Smith, N.D., 1963, *Observation of the directional spectrum of sea waves using the motions of a floating buoy*, Ocean Wave Spectra, Prentice Hall, 111-136.

[25] Lin, R.Q., and Perrie, W., 1999, *Wave-wave interactions in finite depth water*, J. Geophys. Res. Vol. 104, no. c5, pages 11, 193-11, 213.

[26] Mitsuyasu, H., 1981, *Directional spectra of ocean waves in generation area*, Proc. of the Conf. on Directional Wave Spectra Applications, Barkeley, California, 87-101.

[27] Mitsuyasu, H., 2002, *A historical note on the study of ocean surface waves*, Journal of Oceanography, Vol. 58, pp. 109 to 120.

[28] Passino, K.M., and Yurkovich, S., 1998, Fuzzy Control, Addison-Wesley Longman, Inc., USA.

[29] Patyra, M.J., and Mlynek, D.M., 1996, Fuzzy Logic, Implementation and Applications, John Wiley & Sons, Ltd., England.

[30] Phillips, O.M., 1960, *On the dynamics of unsteady gravity waves of finite amplitude, Part 1*, J. Fluid Mech., 9, 193-217.

[31] Pierson, W.J. and Moskowitz, L., 1964, *A proposed spectral form for fully developed wind seas based on the similarity theory of S.A. Kitaigorodskii*, J. Geophys. Res., December 1964, 69(24), 5181-5203.

[32] Polnikov, V.G. and Farina, L., 2002, *On the problem of optimal approximation for the four-wave kinetic integral*, Nonlin. Proc. Geophys., 9, 497-512.

[33] Rahman, M., 1991, Applied Differential Equations for Scientists and Engineers, **2**, Computational Mechanics Publications, Southampton, UK.

[34] Rahman, M., 1995, Water Waves: Relating Modern Theory to Advanced Engineering Applications, Clarendon Press, Oxford, 356 pages.

[35] Resio, D.T., and Perrie, W., 1991, *A numerical study of nonlinear energy fluxes due to wave-wave interactions, I, Methodology and basics results*, J. Fluid Mech., 223, 603-629.

[36] Resio, D.T., Pihl, J.H., Tracy, B.A., and Vincent, C.L., 2001, *Nonlinear energy fluxes and the finite depth equilibrium range in wave spectra*, J. Geophysical Research, **106**, no. C4, 6985-7000.

[37] Ris, R.C., Booij, N., Holthuijsen, L.H., 1999, *A third-generation wave model for coastal regions, Part II: Verification*, J. Geoph. Research, C4, 104, 7997-7681.

[38] Schmid, C., 2005, *Introduction to fuzzy techniques.* Available at http://www.atp.ruhr-uni-bochum.de/rt1/syscontrol/node111.html [2006, 5 August].

[39] Susilo, A., and Perrie, W., 2006, *A dominant transfer approximation for the nonlinear wave-wave interactions in wind wave spectra*, in press, Int. J. of Applied Math. and Eng. Sciences (IJAMES).

[40] Susilo, A., Perrie, W. and Rahman, M., 2006, *Applied dominant transfer and fuzzy logic for nonlinear wave-wave interactions*, in press, Int. J. of Applied Math. and Eng. Sciences (IJAMES).

[41] Susilo, A., and Perrie, W., 2007, *A new approximate computation method for nonlinear wave-wave interactions*, submitted to Ocean Modelling.

[42] Sverdrup, H.U., and Munk, W.H., 1947, Wind, Sea, and Swell: Theory of Relations for Forecasting, U.S. Navy Hydrographic Office, Pub. No. 601.

[43] The SWAMP Group, Ocean Wave Modeling, Plenum Press, New York, 1985.

[44] SWAN, 2003. Available at http://vlm089.citg.tudelft.nl/swan/index.htm [2006, 10 December].

[45] Tolman , H. L, and Chalikov, D.V., 1996, *Source terms in a third-generation wind wave model*, J. Phys. Oceanography, 26, 2497-2518.

[46] Tolman, H.L., 1997, User Manual and System Documentation of WAVEWATCH-III version 1.15, NOAA / NWS / NCEP / OMB Technical Note 151, 97 pp.

[47] Tolman, H.L., 1999, User Manual and System Documentation of WAVEWATCH-III version 1.18, NOAA / NWS / NCEP / OMB Technical Note 166, 110 pp. Available at http://polar.ncep.noaa.gov/waves [2006, 16 December].

[48] Tolman, H.L., 2004, *Inverse modeling of discrete interaction approximations for nonlinear interactions in wind waves*, Ocean Modelling 6, 405-422.

[49] Tracy, B.A., and Resio, D.T., 1982, Theory and Calculation of the Nonlinear Energy Transfer between Sea Waves in Deep Water, WIS Report 11.

[50] Ueberhuber, C. W., 1997, Numerical Computation 2: Methods, Software, and Analysis, Springer-Verlag, Berlin.

[51] Van Vledder, G. Ph., 2000, *Improved method for obtaining the integration space for the computation of nonlinear quadruplet wave-wave interactions*, Alkyon Hydraulic Consultancy & Research, Emmeloord, The Netherland.

[52] Van Vledder, G. Ph., 2006, *The WRT method for the computation of non-linear four-wave interactions in discrete spectral wave models*, J. Coastal Engineering, 53, 223-242.

[53] The WAMDI GROUP, 1988, *The WAM model - A third generation ocean wave prediction model*, J. Physical Oceanography, 18, 1775-1810.

[54] Webb, D.J., 1978, *Non-linear transfers between sea waves*, Deep Sea Research, **25**, pp 279-298.

[55] Weisstein, Eric W., 2004, *Newton-Cotes formulas*, from MathWorld–A Wolfram Web Resource. Available at http://mathworld.wolfram.com/Newton-CotesFormulas.html [2006, 10 August].

[56] Wikipedia contributors, 2001, *Beaufort scale*. Available at http://en.wikipedia.org/wiki/Beaufort scale [2006, 20 July].

[57] Xu, F., Perrie, W., Toulany, B., and Smith, P.C., 2006, *Wind-Generated Waves in Hurricane Juan*, in press, Ocean Modelling.

[58] Yen, J., and Langari, R., 1999, Fuzzy Logic: Intelligence, Control, and Information, Prentice-Hall, Inc., Upper Saddle River, N.J., USA.

[59] Young, I.R., 1999, Wind Generated Ocean Waves, Elsevier, Oxford.

Index

Appendix A

Beaufort Wind Force Scale

Table A.1 is taken from Met Office website, United Kingdom. [1]

Note for Table A.1

- † Beaufort Number
- * These columns are a guide to show roughly what may be expected in the open sea, remote from land.
- Numbers in brackets indicate the probable maximum height of waves. In enclosed waters, or when near land with an offshore wind, wave heights will be smaller and the waves steeper.

[1] http://www.met-office.gov.uk/weather/marine/guide/beaufortscale.html [2006, 1 September].

Table A.1: Beaufort's scale (available at Met Office website, UK).

BN†	Description	Wind speed at 10 m (knots)	Wind speed at 10 m (m/s)	Description in forecasts	Sea of state	Specification for use at sea*	Probable height of wave (m)
0	Calm	< 1	0.0-0.2	Calm	Calm	Sea like a mirror	0.0
1	Light air	1-3	0.3-1.5	Light	Calm	Ripples with the appearance of scales are formed, but without foam crests.	0.1 (0.1)
2	Light breeze	4-6	1.6-3.3	Light	Smooth	Small wavelets, still short but more pronounced. Crests have a glassy appearance and do not break.	0.2 (0.3)
3	Gentle breeze	7-10	3.4-5.4	Light	Smooth	Large wavelets, crests begin to break. Foam of glassy appearance, perhaps scattered white horses.	0.6 (1.0)
4	Moderate breeze	11-16	5.5-7.9	Moderate	Slight	Small waves becoming longer, fairly frequent white horses.	1.0 (1.5)
5	Fresh breeze	17-21	8.0-10.7	Fresh	Moderate	Moderate waves, taking a more pronounced long form; many white horses are formed. Chance of some spray.	2.0 (2.5)
6	Strong breeze	22-27	10.8-13.8	Strong	Rough	Large waves begin to form; the white foam crests are more extensive everywhere. Probably some spray.	3.0 (4.0)
7	Near gale	28-33	13.9-17.1	Strong	Very rough	Sea heaps up and white foam from breaking waves begins to be blown in streaks along the direction of the wind.	4.0 (5.5)
8	Gale	34-40	17.2-20.7	Gale	High	Moderately high waves of greater length; edges of crests begin to break into spindrift. The foam is blown in well-marked streaks along the direction of the wind.	5.5 (7.5)
9	Strong gale	41-47	20.8-24.4	Severe gale	Very high	High waves. Dense streaks of foam along the direction of the wind. Crests of waves begin to topple, tumble and roll over. Spray may affect visibility.	7.0 (10.0)
10	Storm	48-55	24.5-28.4	Storm	Very high	Very high waves with long overhanging crests. The resulting foam, in great patches, is blown in dense white streaks along the direction of the wind. On the whole, the surface of the sea takes a white appearance. The 'tumbling' of the sea becomes heavy and shock-like. Visibility affected.	9.0 (12.5)
11	Violent storm	56-63	28.5-32.6	Violent storm	Phenomenal	Exceptionally high waves (small and medium-sized ships might be lost to view for a time behind the waves). The sea is completely covered with long white patches of foam lying along the direction of the wind. Everywhere the edges of the wave crests are blown into froth. Visibility affected.	11.5 (16.0)
12	Hurricane	≥ 64	≥ 32.7	Hurricane force	Phenomenal	The air is filled with foam and spray. Sea completely white with driving spray; visibility seriously affected.	14.0 (-)

Appendix B

The Study of Ocean Surface Waves

List of acronyms in Table B.1.

OSJ	: The Oceanographic Society of Japan (founded in 1941).
SWOP	: Stereo Wave Observation Project.
SMB	: Sverdrup, Munk and Bretschneider.
PNJ	: Pierson, Neumann and James.
ICCE	: International Conference on Coastal Engineering (started from 1950).
WAM	: Wave Model.
JONSWAP	: Joint North Sea Wave Project.
JWA3G	: Japan Weather Associations Third Generation Wave Model.
ARSLOE	: Atlantic Remote Sensing Land Ocean Experiment.
RIAM	: Wave Observation Project.
Project	
SWADE	: The Surface Waves Dynamics Experiment.
HEXOS	: Humidity EXchange Over the Sea.
RASEX	: Ris Air-Sea Exchange.
SOWEX	: Southern Ocean Waves Experiment.

Table B.1: Advances in the study of ocean surface waves in the latter half of the twentieth century (after Mitsuyasu, 2001).

	1940s	1950s	1960s	1970s	1980s	1990s
Generation mechanism	basic studies Wuest, Roll	new theory Phillips and Miles	extention turbulence	Townsend Gent and Taylor	Al'Zanaidi and Hui numerical study	Belcher Miles (revisit)
Statistical theory	theory of random noise (S.O. Rice)	wave spectrum, wave statistics	spectral form, nonlinear effect	directional spectra, similarity form	high frequency wave spectrum	wave number-frequency spectra
Nonlinear theory	nonlinear theory of regular wave	nonlinear theory of random waves	wave interaction, wave instability	computation dispersion relation	computation wave breaking	wave breaking and energy dissipation
Experiments (Lab. & Ocean)	basic studies visual observation	wave observation (instrumental)	advanced experiment	local equilibrum ocean experiment	wave dynamic exp., satellite observation	microwave remote sensing
Air-sea and wave projects		SWOP, Sun Glitter Project		JONSWAP, RIAM Project	ARSLOE, HEXOS	SWADE, SOWEX, RASEX
Wave forecasting	Sverdrup and Munk	SMB method, PNJ method	numerical method (1st generation)	numerical method (2nd generation)	WAM (3rd generation)	JWA3G data assimilation
International symposium	Ocean Surface Waves	ICCE	Ocean Wave Spectra	Marseille Sympo. (NATO)	Miami Sympo., Sendai Sympo.	Miami Sympo., Sydney Sympo.
Progress in science and technology	atomic bomb, computer(ENIAC), transistor, OSJ	Challenger VIII satellite (Sputnik), DNA	big bang (3k radi.) satellite (Appollo), quark model	Club of Rome rep., pollution problem, SEASAT, super LSI	Global warming, space stuttle, GEOSAT, ozone hole	top quark, complex system ERSI 2 ADEOS

Appendix C

Coupling Coefficient

The coupling coefficient C is defined as

$$C(\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3, \mathbf{k}_4) = \frac{\pi g^2 D^2(\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3, \mathbf{k}_4)}{4 w_1 w_2 w_3 w_4} \qquad (m^{-4} s^{-4}).$$

where

$$\begin{aligned} D(\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3, \mathbf{k}_4) = & \frac{2(w_1+w_2)^2 (k_1 k_2 - \mathbf{k}_1 \cdot \mathbf{k}_2)(k_3 k_4 - \mathbf{k}_3 \cdot \mathbf{k}_4)}{w_{1+2}^2 - (w_1+w_2)^2} \\ & + \frac{2(w_1-w_3)^2 (k_1 k_3 - \mathbf{k}_1 \cdot \mathbf{k}_3)(k_2 k_4 - \mathbf{k}_2 \cdot \mathbf{k}_4)}{w_{1-3}^2 - (w_1-w_3)^2} \\ & + \frac{2(w_1-w_4)^2 (k_1 k_4 - \mathbf{k}_1 \cdot \mathbf{k}_4)(k_2 k_3 - \mathbf{k}_2 \cdot \mathbf{k}_3)}{w_{1-4}^2 - (w_1-w_4)^2} \\ & + \frac{1}{2}[(\mathbf{k}_1 \cdot \mathbf{k}_2)(\mathbf{k}_3 \cdot \mathbf{k}_4) + (\mathbf{k}_1 \cdot \mathbf{k}_3)(\mathbf{k}_2 \cdot \mathbf{k}_4) + (\mathbf{k}_1 \cdot \mathbf{k}_4)(\mathbf{k}_2 \cdot \mathbf{k}_3)] \\ & - \frac{1}{4}(\mathbf{k}_1 \cdot \mathbf{k}_2 + \mathbf{k}_3 \cdot \mathbf{k}_4)(w_1+w_2)^4 + \frac{1}{4}(\mathbf{k}_1 \cdot \mathbf{k}_3 + \mathbf{k}_2 \cdot \mathbf{k}_4)(w_1-w_3)^4 \\ & + \frac{1}{4}(\mathbf{k}_1 \cdot \mathbf{k}_4 + \mathbf{k}_2 \cdot \mathbf{k}_3)(w_1-w_4)^4 + \frac{5}{2} k_1 k_2 k_3 k_4 \\ & + (w_1+w_2)^2 (w_1-w_3)^2 (w_1-w_4)^2 (k_1+k_2+k_3+k_4). \end{aligned}$$

Here,

$$k_i = |\mathbf{k}_i|, \qquad w_i = \sqrt{k_i}$$

and

$$w_{1+2} = \sqrt{|\mathbf{k}_1 + \mathbf{k}_2|}, \qquad w_{1-3} = \sqrt{|\mathbf{k}_1 - \mathbf{k}_3|}, \qquad w_{1-4} = \sqrt{|\mathbf{k}_1 - \mathbf{k}_4|}.$$

Appendix D

Main and Subprograms

The following are diagrams of pre-processor, processor, and postprocessor subprograms used by WAVEWATCH III.

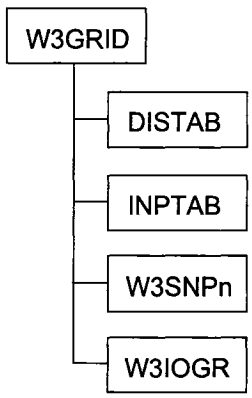


Figure D.1: **ww3_grid**'s subprograms (after Tolman, 1999).

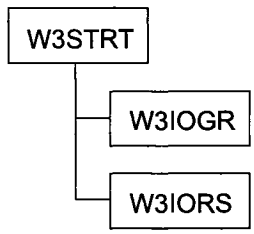


Figure D.2: **ww3_strt**'s subprograms (after Tolman, 1999).

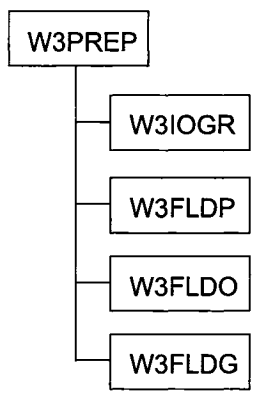


Figure D.3: **ww3_prep**'s subprograms (after Tolman, 1999).

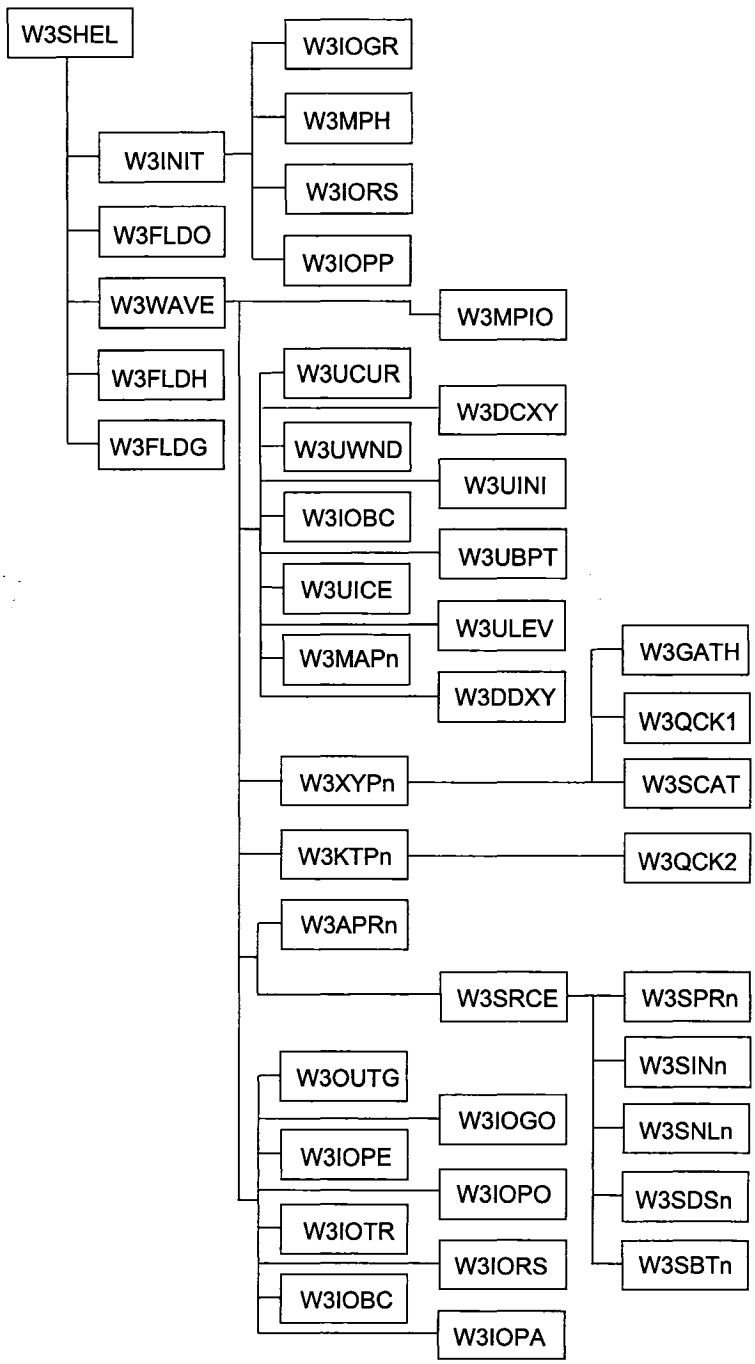


Figure D.4: **ww3_shel**'s subprograms (after Tolman, 1999).

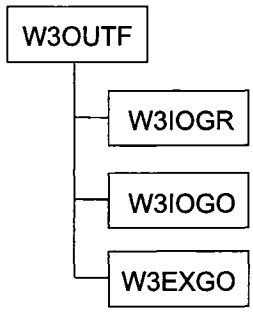


Figure D.5: **ww3_outf**'s subprograms (after Tolman, 1999).

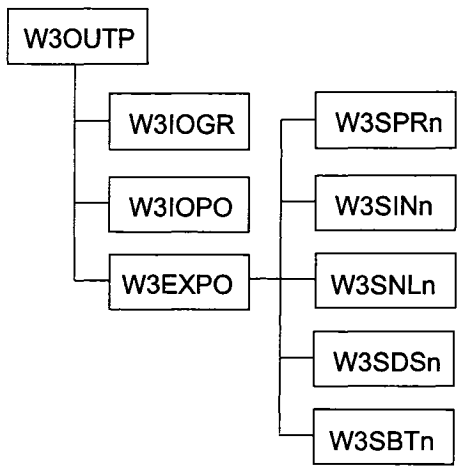


Figure D.6: **ww3_outp**'s subprograms (after Tolman, 1999).

Appendix E

Edited Files

The following are edited files and their contents as the results of installing the new nonlinear term in WAVEWATCH III.

- Directory: **aux**
  1. **w3adc.f**

- Directory: **bin**
  1. **ad3**
  2. **ad3_test**
  3. **comp**
  4. **comp.gen**
  5. **link**
  6. **link.gen**
  7. **make_makefile.prog**
  8. **make_makefile.subs**
  9. **w3_new**

- Directory: **ftn**
  1. **dims.cmn**
  2. **dimx.cmn**
  3. **w3expo.ftn**

4. **w3init.ftn**
5. **w3iogr.ftn**
6. **w3srce.ftn**
7. **w3wave.ftn**
8. **ww3_grid.ftn**
9. **ww3_outp.ftn**
10. **ww3_shel.ftn**

- Directory: **inp**
  1. **ww3_grid.inp**

Directory: aux

1. w3adc.f

before:

```
      PARAMETER ( MMFILE =  20 )
```

after:

```
      PARAMETER ( MMFILE =  30 )
```

Directory: bin

1. ad3

before:

```
. . .
echo "'##outp.dat'  '$path_i/outp.dat' "      >> w3adc.inp
```

after:

```
. . .
echo "'##outp.dat'  '$path_i/outp.dat' "      >> w3adc.inp
echo "'##nl4p.cmn'  '$path_i/nl4p.cmn' "      >> w3adc.inp
echo "'##loci.cmn'  '$path_i/loci.cmn' "      >> w3adc.inp
echo "'##grid.cmn'  '$path_i/grid.cmn' "      >> w3adc.inp
echo "'##geob.cmn'  '$path_i/geob.cmn' "      >> w3adc.inp
echo "'##geoe.cmn'  '$path_i/geoe.cmn' "      >> w3adc.inp
echo "'##fzzl.cmn'  '$path_i/fzzl.cmn' "      >> w3adc.inp
echo "'##nc10.cmn'  '$path_i/nc10.cmn' "      >> w3adc.inp
```

2. ad3_test

The changes are same as **ad3**.

3. comp

At section 'Compile, Generic' use this line

```
g77 -c $name.f                > $name.out 2> $name.err
```

4. comp.gen

The change is same as **comp**.

5. link

At section 'Link all things, Generic' use this line

```
g77 -o $prog $objects         > link.out 2> link.err
```

6. link.gen

The change is same as **link**.

7. make_makefile.prog

before:

```
. . .
OK='NLO NL1 NLX' ;;
```

after:

```
. . .
OK='NLO NL1 NLB NLE NLX' ;;
```

before:

```
. . .
NL1) nl1='w3snp1'
     nl2='w3snl1' ;;
```

after:

```
. . .
NL1) nl1='w3snp1'
     nl2='w3snl1' ;;
NLB) nl1='delw cple locus fgeob'
     nl2='findy defuz ffdfz findn pten w3snlb' ;;
NLE) nl1='delw cple locus fgeoe'
     nl2='findn w3snle' ;;
```

8. make_makefile.subs

before:

```
. . .
files='w3sin1 w3sin2 w3snl1 w3snp1 w3sds1 w3sds2 w3sbt1' ;;
```

after:

```
. . .
files='w3sin1 w3sin2 w3snl1 w3snp1 w3sds1 w3sds2 w3sbt1' ;;
files='w3sin1 w3sin2 w3snl1 w3snp1 w3sds1 w3sds2 w3sbt1'
files="$files findn findy defuz ffdfz fgeob fgeoe"
files="$files pten w3snlb w3snle" ;;
```

before:

```
. . .
files='wavnu1 wavnu2 distab inptab fej5p' ;;
```

after:

```
. . .
files='wavnu1 wavnu2 distab inptab fej5p' ;;
files='wavnu1 wavnu2 distab inptab fej5p'
files="$files delw cple locus" ;;
```

before:

```
. . .
tab1.cmn tab2.cmn aux1.cmn auxt.cmn outp.dat
```

after:

```
. . .
tab1.cmn tab2.cmn aux1.cmn auxt.cmn outp.dat \
nl4p.cmn loci.cmn grid.cmn geob.cmn geoe.cmn fzzl.cmn \
nc10.cmn
```

9. w3_new

before:

```
'spec'   ) cd $main_dir/ftn ; touch dims.cmn ;;
```

after:

```
'spec'   ) cd $main_dir/ftn ; touch dims.cmn
                              touch nl4p.cmn ;;
```

Directory: ftn

1. dims.cmn

before:

```
      PARAMETER     ( NTH    =     24 )
      PARAMETER     ( NK     =     25 )
```

after:

```
      PARAMETER     ( NTH    =     36 )
      PARAMETER     ( NK     =     30 )
```

2. dimx.cmn

before:

```
      PARAMETER     ( MX     =     12 )
      PARAMETER     ( MY     =     12 )
      PARAMETER     ( MSEA   =    100 )
```

after:

```
      PARAMETER     ( MX     =     nX )
      PARAMETER     ( MY     =     nY )
      PARAMETER     ( MSEA   =    nX x nY )
```

where nX and nY are number of points in xy-plane.

3. w3expo.ftn

before:

```
. . .
##auxt.cmn
```

after:

```
. . .
##auxt.cmn
##nl4p.cmn
##geob.cmn
##geoe.cmn
##grid.cmn
##fzzl.cmn
##nc10.cmn
```

before:

```
. . .
C/NL1     CALL W3SNL1 ( A, CG, WNMEAN*DEPTH,  XNL, DNL )
```

after:

```
. . .
C/NL1     CALL W3SNL1 ( A, CG, WNMEAN*DEPTH,  XNL, DNL )
C/NLB     CALL W3SNLB ( A, CG, UABS,          XNL, DNL )
C/NLE     CALL W3SNLE ( A, CG,                XNL, DNL )
```

4. w3init.ftn

before:

```
. . .
##auxt.cmn
```

after:

```
. . .
##auxt.cmn
##nl4p.cmn
##geob.cmn
##geoe.cmn
##grid.cmn
##fzzl.cmn
##nc10.cmn
```

5. w3iogr.ftn

before:

```
. . .
##auxt.cmn
```

after:

```
. . .
##auxt.cmn
##nl4p.cmn
##geob.cmn
##geoe.cmn
##grid.cmn
##fzzl.cmn
##nc10.cmn
```

before:

```
. . .
      INTEGER         ISEA, IX, IY
```

after:

```
. . .
      INTEGER         ISEA, IX, IY, IZ, IR
```

before:

```
. . .
C/NL1      TNAME2 = 'Discrete Interaction Approx.  '
```

after:

```
. . .
C/NL1      TNAME2 = 'Discrete Interaction Approx.  '
C/NLB      TNAME2 = 'Advance Dominant Int., AvDI   '
C/NLE      TNAME2 = 'WTR  Method                   '
```

In 'Source term parameters' add following lines

```
*
C/NLB      IF ( WRITE ) THEN
C/NLB          WRITE (NDSM)
C/NLB     &          (WKA(IR),IR=1,NNR),(PHA(IR),IR=1,NNR)
C/NLB        ELSE
C/NLB          READ  (NDSM,END=801,ERR=802,IOSTAT=IERR)
C/NLB     &          (WKA(IR),IR=1,NNR),(PHA(IR),IR=1,NNR)
C/NLB        ENDIF
*
C/NLB      IF ( WRITE ) THEN
C/NLB          WRITE (NDSM)
C/NLB     &      (((X2BL(IX,IY,IZ),IX=1,NNA),IY=1,2),IZ=1,NLOCI),
C/NLB     &      (((Y2BL(IX,IY,IZ),IX=1,NNA),IY=1,2),IZ=1,NLOCI),
C/NLB     &      (((X4BL(IX,IY,IZ),IX=1,NNA),IY=1,2),IZ=1,NLOCI),
C/NLB     &      (((Y4BL(IX,IY,IZ),IX=1,NNA),IY=1,2),IZ=1,NLOCI),
C/NLB     &      (((GEOM(IX,IY,IZ),IX=1,NNA),IY=1,2),IZ=1,NLOCI),
C/NLB     &       H10
C/NLB        ELSE
C/NLB          READ  (NDSM,END=801,ERR=802,IOSTAT=IERR)
C/NLB     &      (((X2BL(IX,IY,IZ),IX=1,NNA),IY=1,2),IZ=1,NLOCI),
C/NLB     &      (((Y2BL(IX,IY,IZ),IX=1,NNA),IY=1,2),IZ=1,NLOCI),
C/NLB     &      (((X4BL(IX,IY,IZ),IX=1,NNA),IY=1,2),IZ=1,NLOCI),
C/NLB     &      (((Y4BL(IX,IY,IZ),IX=1,NNA),IY=1,2),IZ=1,NLOCI),
C/NLB     &      (((GEOM(IX,IY,IZ),IX=1,NNA),IY=1,2),IZ=1,NLOCI),
C/NLB     &       H10
C/NLB        ENDIF
*
C/NLE      IF ( WRITE ) THEN
```

```
C/NLE          WRITE (NDSM)
C/NLE     &          (WKA(IR),IR=1,NNR),(PHA(IR),IR=1,NNR)
C/NLE        ELSE
C/NLE          READ  (NDSM,END=801,ERR=802,IOSTAT=IERR)
C/NLE     &          (WKA(IR),IR=1,NNR),(PHA(IR),IR=1,NNR)
C/NLE        ENDIF
*
C/NLE      IF ( WRITE ) THEN
C/NLE          WRITE (NDSM)
C/NLE     & ((((X2BLE(IX,IY,IR,IZ),IX=1,NNA),IY=1,NNA),IR=1,NNR),IZ=1,NPA),
C/NLE     & ((((Y2BLE(IX,IY,IR,IZ),IX=1,NNA),IY=1,NNA),IR=1,NNR),IZ=1,NPA),
C/NLE     & ((((X4BLE(IX,IY,IR,IZ),IX=1,NNA),IY=1,NNA),IR=1,NNR),IZ=1,NPA),
C/NLE     & ((((Y4BLE(IX,IY,IR,IZ),IX=1,NNA),IY=1,NNA),IR=1,NNR),IZ=1,NPA),
C/NLE     & ((((GEOME(IX,IY,IR,IZ),IX=1,NNA),IY=1,NNA),IR=1,NNR),IZ=1,NPA)
C/NLE        ELSE
C/NLE          READ  (NDSM,END=801,ERR=802,IOSTAT=IERR)
C/NLE     & ((((X2BLE(IX,IY,IR,IZ),IX=1,NNA),IY=1,NNA),IR=1,NNR),IZ=1,NPA),
C/NLE     & ((((Y2BLE(IX,IY,IR,IZ),IX=1,NNA),IY=1,NNA),IR=1,NNR),IZ=1,NPA),
C/NLE     & ((((X4BLE(IX,IY,IR,IZ),IX=1,NNA),IY=1,NNA),IR=1,NNR),IZ=1,NPA),
C/NLE     & ((((Y4BLE(IX,IY,IR,IZ),IX=1,NNA),IY=1,NNA),IR=1,NNR),IZ=1,NPA),
C/NLE     & ((((GEOME(IX,IY,IR,IZ),IX=1,NNA),IY=1,NNA),IR=1,NNR),IZ=1,NPA)
C/NLE        ENDIF
*
```

6. w3srce.ftn

before:

```
. . .
C     C/NL1   Discrete interaction approximation.
```

after:

```
. . .
C     C/NL1   Discrete interaction approximation.
C     C/NLB   Exact interaction approximation (AvDI).
C     C/NLE   Exact interaction approximation (BIO-EXA).
```

before:

```
. . .
##auxt.cmn
```

after:

```
. . .
##auxt.cmn
##nl4p.cmn
##geob.cmn
##geoe.cmn
##grid.cmn
##fzzl.cmn
##nc10.cmn
```

before:

```
. . .
C/NL1      CALL W3SNL1 ( SPEC, CG1, WNMEAN*DEPTH, VSNL, VDNL )
```

after:

```
. . .
C/NL1      CALL W3SNL1 ( SPEC, CG1, WNMEAN*DEPTH, VSNL, VDNL )
C/NLB      CALL W3SNLB ( SPEC, CG1, U10ABS,       VSNL, VDNL )
C/NLE      CALL W3SNLE ( SPEC, CG1,               VSNL, VDNL )
```

before:

```
. . .
C/NL1     &                                  + VSNL(IS)
```

after:

```
. . .
C/NL1     &                                  + VSNL(IS)
C/NLB     &                                  + VSNL(IS)
C/NLE     &                                  + VSNL(IS)
```

before:

```
. . .
C/NL1     &                    + VDNL(IS)
```

after:

```
. . .
C/NL1     &                    + VDNL(IS)
C/NLB     &                    + VDNL(IS)
C/NLE     &                    + VDNL(IS)
```

7. w3wave.ftn

before:

```
. . .
##auxt.cmn
```

after:

```
. . .
##auxt.cmn
##nl4p.cmn
##geob.cmn
##geoe.cmn
##grid.cmn
##fzzl.cmn
##nc10.cmn
```

8. ww3_grid.ftn

before:

```
. . .
##aux1.cmn
```

after:

```
. . .
##aux1.cmn
##nl4p.cmn
##geob.cmn
##geoe.cmn
##grid.cmn
##nc10.cmn
```

before:

```
. . .
      WRITE (NDSO,903) NTH, DTH*RADE, NK, FR1, XFR
```

after:

```
. . .
      WRITE (NDSO,903) NTH, DTH*RADE, NK, FR1, FR1*XFR**(NK-1), XFR
```

before:

```
. . .
##aux1.cmn
```

after:

```
. . .
##aux1.cmn
##nl4p.cmn
##geob.cmn
##geoe.cmn
##grid.cmn
##nc10.cmn
```

before:

```
. . .
C/NL1       NRNL    = NRNL + 1
```

after:

```
. . .
C/NL1       NRNL    = NRNL + 1
C/NLB       NRNL    = NRNL + 1
C/NLE       NRNL    = NRNL + 1
```

before:

```
. . .
C/NL1       WRITE (NDSO,1922)
```

after:

```
. . .
C/NL1       WRITE (NDSO,1922)
C/NLB       WRITE (NDSO,7922)
C/NLE       WRITE (NDSO,8922)
```

before:

```
. . .
C/NL1       CALL W3SNP1 ( LAMBDA )
*
```

after:

```
. . .
C/NL1       CALL W3SNP1 ( LAMBDA )
*
C/NLB       WRITE (NDSO,8923) NNR,NNA,NPA,FR1,XFR
C/NLB       WRITE (NDSO,888)
C/NLB       CALL FGEOB
*
C/NLE       WRITE (NDSO,8923) NNR,NNA,NPA,FR1,XFR
C/NLE       WRITE (NDSO,888)
C/NLE       CALL FGEOE
*
```

before:

```
. . .
C/NL1       WRITE (NDSO,1922)
```

after:

```
. . .
C/NL1       WRITE (NDSO,1922)
C/NLB       WRITE (NDSO,7922)
C/NLE       WRITE (NDSO,8922)
```

before:

```
. . .
           &    '       Lowest frequency       (Hz) :',F9.4/
```

after:

```
. . .
           &    '       Frequency range        (Hz) :',F9.4,'-',F6.4/
```

before:

```
. . .
C/NL1      &    '       shallow water constants     :',F8.2,2F6.2/)
*
```

after:

```
. . .
C/NL1     &    '       shallow water constants     :',F8.2,2F6.2/)
*
C/NLB 7922 FORMAT (/'  Nonlinear interactions (AvDI)    :'/
C/NLB     &    '  (this method suits for deep water only!)'/
C/NLB     &    '-----------------------------------------------')
C/NLB 8923 FORMAT ( '       Number of frequency (=30)   :',I4/
C/NLB     &         '       Number of angle     (=36)   :',I4/
C/NLB     &         '       -input must be equal as in bracket, ' /
C/NLB     &         '     if not, go to dims.cmn to edit those- ' /
C/NLB     &         '       Number points in loci       :',I4/
C/NLB     &         '       -go to nl4p.cmn to edit it- '   /
C/NLB     &         '       First frequency (Hz)        :',F8.3/
C/NLB     &         '       Freq. multiplication factor :',F8.3/)
*
C/NLE 8922 FORMAT (/'  Nonlinear interactions (BIO-EXA) :'/
C/NLE     &    '  (this method suits for deep water only!)'/
C/NLE     &    '-----------------------------------------------')
C/NLE 8923 FORMAT ( '       Number of frequency         :',I4/
C/NLE     &         '       Number of angle             :',I4/
C/NLE     &         '       -go to dims.cmn to edit those- ' /
C/NLE     &         '       Number points in loci       :',I4/
C/NLE     &         '       -go to nl4p.cmn to edit it- '   /
C/NLE     &         '       First frequency (Hz)        :',F8.3/
C/NLE     &         '       Freq. multiplication factor :',F8.3/)
*
```

before:

```
. . .
C/O2a  998 FORMAT (80I1)
*
```

after:

```
. . .
C/O2a  998 FORMAT (80I1)
*
 888 FORMAT (/'  Computing GEOMETRY term..... ...'/)
```

9. ww3_outp.ftn

before:

```
. . .
##auxt.cmn
```

after:

```
. . .
##auxt.cmn
##nl4p.cmn
##geob.cmn
##geoe.cmn
##grid.cmn
##fzzl.cmn
##nc10.cmn
```

10. ww3_shel.ftn

before:

```
. . .
##spar.cmn
```

after:

```
. . .
##spar.cmn
##nl4p.cmn
##geob.cmn
##geoe.cmn
##grid.cmn
##fzzl.cmn
##nc10.cmn
```

Directory: inp

1. ww3_grid.inp

Do not activate nonlinear constants.

```
$
$ Nonlinear interactions - - - - - - - - - - - - - - - - - - - - - - -
$   Not defined          : -
$   Discrete I.A.        : lambda and C source term,conversion factor
$                          kd in Eq. (2.24),minimum kd, and
$                          constants c1-3 in depth scaling function.
$                            line 1: Cf. WAM model.
$                            line 2: Cf. Tolman and Chalikov (1996).
$
$   0.25     2.78E7     0.75    0.50    5.5    0.833   -1.25
$   0.25     1.00E7     0.75    0.50    5.5    0.833   -1.25
$
$   Experimental         : -
$
$ Dissipation  - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Appendix F

New COMMONs

List of new 'COMMON's in alphabetical order and their contents (see next page).

1. **fzzl.cmn**
2. **loci.cmn**
3. **geob.cmn**
4. **geoe.cmn**
5. **grid.cmn**
6. **nc10.cmn**
7. **nl4p.cmn**

1. fzzl.cmn

```
C/
C/ ---- WW-III SOURCE FUNCTION COMMON : SNL4(BIO)------------------/
C/                                        Written by  : Adhi Susilo
C/                                        Last update : 21-Feb-2006
C/
*
      REAL            XSP, XGAM, FACTD
```

2. loci.cmn

```
C/
C/ ---- WW-III SOURCE FUNCTION COMMON : SNL4(BIO)------------------/
C/                                        Written by  : Adhi Susilo
C/                                        Last update : 25-Dec-2005
C/
      REAL            XLOC2,YLOC2,XLOC4,YLOC4,DS
*
      COMMON /LOCI/   XLOC2(NPA),YLOC2(NPA),
     &                XLOC4(NPA),YLOC4(NPA),DS(NPA)
```

3. geob.cmn

```
C/
C/ ---- WW-III SOURCE FUNCTION COMMON : SNL4(BIO)----------------- /
C/                                        Written by  : Adhi Susilo
C/                                        Last update : 11-Apr-2006
C/                                                      16-Mar-2006
C/                                                      25-Dec-2005
C/
      REAL            H10,GEOM,X2BL,X4BL,Y2BL,Y4BL
*
      COMMON /GEOB/   H10,GEOM(NNA,2,NLOCI),
     &                X2BL(NNA,2,NLOCI),X4BL(NNA,2,NLOCI),
     &                Y2BL(NNA,2,NLOCI),Y4BL(NNA,2,NLOCI)
```

4. geoe.cmn

```
C/
C/ ---- WW-III SOURCE FUNCTION COMMON : SNL4(BIO)----------------- /
C/                                        Written by  : Adhi Susilo
C/                                        Last update : 25-Dec-2005
C/
      REAL            GEOME,X2BLE,X4BLE,Y2BLE,Y4BLE
*
      COMMON /GEOE/   GEOME(NNA,NNA,NNR,NPA),
     &                X2BLE(NNA,NNA,NNR,NPA),X4BLE(NNA,NNA,NNR,NPA),
     &                Y2BLE(NNA,NNA,NNR,NPA),Y4BLE(NNA,NNA,NNR,NPA)
```

5. grid.cmn

```
C/
C/ ---- WW-III SOURCE FUNCTION COMMON : SNL4(BIO)----------------- /
C/                                        Written by  : Adhi Susilo
C/                                        Last update : 25-Dec-2005
C/
*
      REAL            DENS13,WKA,PHA
*
      COMMON /GRID/   DENS13(NNR,NNA),WKA(NNR),PHA(NNR)
```

6. nc10.cmn

```
C/
C/ ---- WW-III SOURCE FUNCTION COMMON : SNL4(BIO)----------------- /
C/                                        Written by  : Adhi Susilo
C/                                        Last update : 11-Apr-2006
C/
      REAL            TR10,DG10
*
      COMMON /NC10/   TR10(NLOCI),DG10(NLOCI)
```

7. nl4p.cmn

```
C/
C/ ---- WW-III SOURCE FUNCTION COMMON : SNL4(BIO)----------------- /
C/                                        Written by  : Adhi Susilo
C/                                        Last update : 16-Mar-2006
C/                                                      25-Dec-2005
C/
      REAL           XLM
      INTEGER        NNR,NNA,NPA,I3S,IPAP,IPAN,ILOCI,NLOCI
*
      PARAMETER ( NNR = NK  )
      PARAMETER ( NNA = NTH )
*
C/ NPA is number of points of a loci
      PARAMETER ( NPA = 36  )
*
C/ I3S is k3 chosen
      PARAMETER ( I3S =  1  )
*
C/ IPAp/n is k3 chosen angle
      PARAMETER ( IPAP =  2 )
      PARAMETER ( IPAN = -2 )
*
C/ For I10, the selected points along loci
      PARAMETER ( NLOCI = NPA/2 )
```

Appendix G

New Programs

List of new programs in alphabetical order and their contents (see next page).

1. **cple.ftn***
2. **defuz.ftn**
3. **delw.ftn***
4. **ffdfz.ftn**
5. **fgeob.ftn**
6. **fgeoe.ftn**
7. **findn.ftn**
8. **findy.ftn**
9. **locus.ftn***
10. **pten.ftn**
11. **w3snlb.ftn**
12. **w3snle.ftn**

Note:
* these programs are taken from Resio's work and are modified by author so they can be suitable for the wave model, WAVEWATCH III version 1.18.

1. cple.ftn

```
C/ ------------------------------------------------------------ /
      SUBROUTINE CPLE(X1, Y1, X2, Y2, X3, Y3, X4, Y4, CSQ)
##docb
C/                  +-----------------------------------+
C/                  |     FISHERIES & OCEANS CANADA     |
C/                  | BEDFORD INSTITUTE OF OCEANOGRAPHY |
C/                  |   Dartmouth, Nova Scotia, Canada  |
C/                  |                &                  |
C/                  |       DALHOUSIE UNIVERSITY        |
C/                  |    Halifax, Nova Scotia, Canada   |
C/                  |                                   |
C/                  |           Adhi Susilo             |
C/                  |                          FORTRAN 77 |
C/                  | Last update :          25-Dec-2005 |
C/                  +-----------------------------------+
C/
C  1. Purpose :
C
C     Calculate the coupling coefficient.
C
C  2. Method :
C
C     Compute C(k1,k2,k3,k4) with Webb's theorem.
C     This subroutine is from subroutine of a program called
C     windwave3.f, written by Resio.
C
C  3. Parameters :
C
C     Parameter list
C     ----------------------------------------------------------------
C       Xn,Yn    Real   Components of wave number in x-y axis.
C       CSQ      Real   The coupling coefficient.
C
C     Local parameters
C     ----------------------------------------------------------------
C     See source code.
C
C  4. Subroutines used :
```

```
C
C       None.
C
C  5. Called by :
C
C       FGEOB,FGEOE
C           Finding the geometry term.
C
C  6. Error messages :
C
C       None.
C
C  7. Remarks :
C
C       None.
C
C  8. Structure :
C
C     See source code.
C
C  9. Switches :
C
C       None.
C
C 10. Source code :
C
C/------------------------------------------------------------------ /
      IMPLICIT NONE
C
C/ Local parameters
C/
      REAL X1,Y1,X2,Y2,X3,Y3,X4,Y4,Z,Z1,Z2,Z3,
     &     PI4,EPS,WK1,WK2,WK3,WK4,W1,W2,W3,W4,
     &     DOT12,DOT13,DOT14,DOT23,DOT24,DOT34,
     &     WSQP12,WSQ12,WSQM13,WSQ13,WSQM14,WSQ14,
     &     P1,P2,P3,P4,P5,P6,P7,DS,D,CSQ
*
      DATA PI4/.78539/
*
      EPS = 1.0E-30
```

```
      WK1 = SQRT(X1*X1+Y1*Y1)
      WK2 = SQRT(X2*X2+Y2*Y2)
      WK3 = SQRT(X3*X3+Y3*Y3)
      WK4 = SQRT(X4*X4+Y4*Y4)
      W1 = SQRT(WK1)
      W2 = SQRT(WK2)
      W3 = SQRT(WK3)
      W4 = SQRT(WK4)
      DOT12 = X1*X2+Y1*Y2
      DOT13 = X1*X3+Y1*Y3
      DOT14 = X1*X4+Y1*Y4
      DOT23 = X2*X3+Y2*Y3
      DOT24 = X2*X4+Y2*Y4
      DOT34 = X3*X4+Y3*Y4
      WSQP12= SQRT( (X1+X2)*(X1+X2) + (Y1+Y2)*(Y1+Y2) )
      WSQ12 = (W1+W2)*(W1+W2)
      WSQM13= SQRT((X1-X3)*(X1-X3)+(Y1-Y3)*(Y1-Y3))
      WSQ13 = (W1-W3)*(W1-W3)
      WSQM14= SQRT( (X1-X4)*(X1-X4) + (Y1-Y4)*(Y1-Y4) )
      WSQ14 = (W1-W4)*(W1-W4)
      Z1 = WSQP12-WSQ12
      Z2 = WSQM13-WSQ13
      Z3 = WSQM14-WSQ14
      Z = 2.*WSQ12*(WK1*WK2-DOT12)*(WK3*WK4-DOT34)
      P1= Z/(Z1+EPS)
      Z = 2.*WSQ13*(WK1*WK3+DOT13)*(WK2*WK4+DOT24)
      P2= Z/(Z2+EPS)
      Z = 2.*WSQ14*(WK1*WK4+DOT14)*(WK2*WK3+DOT23)
      P3= Z/(Z3+EPS)
      P4= 0.5 *(DOT12*DOT34+DOT13*DOT24+DOT14*DOT23)
      P5= 0.25*((DOT13+DOT24)*WSQ13*WSQ13-(DOT12+DOT34)*WSQ12*WSQ12)
      P6= 0.25*(DOT14+DOT23)*WSQ14*WSQ14+2.5*WK1*WK2*WK3*WK4
      P7= WSQ12*WSQ13*WSQ14*(WK1+WK2+WK3+WK4)
      DS= P1+P2+P3+P4+P5+P6+P7
      D = DS*DS
      CSQ = PI4*D/(W1*W2*W3*W4+EPS)
*
C/ End of CPLE ------------------------------------------------- /
C/
      END
```

2. defuz.ftn

```
C/ ------------------------------------------------------------ /
      REAL FUNCTION DEFUZ(WA, WB, WC, WD, FA, FB, FC, FD)
##docb
C/                  +----------------------------------+
C/                  |     FISHERIES & OCEANS CANADA    |
C/                  | BEDFORD INSTITUTE OF OCEANOGRAPHY |
C/                  |   Darmouth, Nova Scotia, Canada  |
C/                  |                &                 |
C/                  |       DALHOUSIE UNIVERSITY       |
C/                  |    Halifax, Nova Scotia, Canada  |
C/                  |                                  |
C/                  |           Adhi Susilo            |
C/                  |                         FORTRAN 77 |
C/                  | Last update :         20-Feb-2006 |
C/                  +----------------------------------+
C/
C  1. Purpose :
C
C     To find crips value from fuzzy logic.
C
C  2. Method :
C
C     (Sum weight*crips)/(Sum weight)
C
C  3. Parameters :
C
C     Parameter list
C     ------------------------------------------------------------
C       DEFUZ      Real   The value of defuzzying.
C
C     Local parameters
C     ------------------------------------------------------------
C     See source code.
C
C  4. Subroutines used :
C
C       None.
C
```

```
C  5. Called by :
C
C       FFDFZ
C
C  6. Error messages :
C
C       None.
C
C  7. Remarks :
C
C       None.
C
C  8. Structure :
C
C     See source code.
C
C  9. Switches :
C
C       None.
C
C 10. Source code :
C
C/ ------------------------------------------------------------------ /
*
      REAL  WA,WB,WC,WD,FA,FB,FC,FD,FNOM,FDOM
*
      FNOM  = FA+FB+FC+FD
      FDOM  = WA+WB+WC+WD
      DEFUZ = FNOM/FDOM
C
C/ End of DEFUZ ---------------------------------------------------- /
C
      END
```

3. delw.ftn

```
C/ ------------------------------------------------------------ /
      REAL FUNCTION DELW(X2,Y2,X4,Y4)
##docb
C/                  +-----------------------------------+
C/                  |     FISHERIES & OCEANS CANADA     |
C/                  | BEDFORD INSTITUTE OF OCEANOGRAPHY |
C/                  |   Dartmouth, Nova Scotia, Canada  |
C/                  |                &                  |
C/                  |       DALHOUSIE UNIVERSITY        |
C/                  |    Halifax, Nova Scotia, Canada   |
C/                  |                                   |
C/                  |           Adhi Susilo             |
C/                  |                          FORTRAN 77 |
C/                  | Last update :         25-Dec-2005 |
C/                  +-----------------------------------+
C/
C  1. Purpose :
C
C     Calculate the magnitute of grad W.
C
C  2. Method :
C
C     See Tracy-Resio 1982. This subroutine is from subroutine of
C     a program called windwave3.f, written by Resio.
C
C  3. Parameters :
C
C     Parameter list
C     ----------------------------------------------------------
C       NONE
C     ----------------------------------------------------------
C
C     Local parameters
C     ----------------------------------------------------------
C       Xn,Yn   Real   Components of wave number in x-y axis.
C     ----------------------------------------------------------
C
C  4. Subroutines used :
```

```
C
C       None.
C
C  5. Called by :
C
C       FGEOB,FGEOE
C           Finding geometry term.
C
C  6. Error messages :
C
C       None.
C
C  7. Remarks :
C
C       None.
C
C  8. Structure :
C
C     See source code.
C
C  9. Switches :
C
C       None.
C
C 10. Source code :
C
C/ ------------------------------------------------------------ /
      IMPLICIT NONE
C
C/ ------------------------------------------------------------ /
C/ Local parameters
C/
      REAL X2,Y2,X4,Y4,ZZX,ZZY,
     &     XK2SQ,XK4SQ,Z2,Z4,ZZSUM
*
      XK2SQ = X2*X2+Y2*Y2
      XK4SQ = X4*X4+Y4*Y4
      Z2 = XK2SQ**(-0.75)
      Z4 = XK4SQ**(-0.75)
      ZZX= X2*Z2-X4*Z4
```

```
      ZZY= Y2*Z2-Y4*Z4
      ZZSUM = ZZX*ZZX+ZZY*ZZY
      IF (ZZSUM.LT.1.0E-25) ZZSUM=1.0E-25
      DELW = 0.5*SQRT(ZZSUM)
*
      RETURN
*
C/ End of DELW ------------------------------------------------- /
C/
      END
```

4. ffdfz.ftn

```
C/ ------------------------------------------------------------ /
      SUBROUTINE FFDFZ (XSP, XGAM, FACTD)
##docb
C/                  +-----------------------------------+
C/                  |     FISHERIES & OCEANS CANADA     |
C/                  | BEDFORD INSTITUTE OF OCEANOGRAPHY |
C/                  |   Darmouth, Nova Scotia, Canada   |
C/                  |                &                  |
C/                  |       DALHOUSIE UNIVERSITY        |
C/                  |    Halifax, Nova Scotia, Canada   |
C/                  |                                   |
C/                  |           Adhi Susilo             |
C/                  |                         FORTRAN 77 |
C/                  | Last update :         16-Mar-2006 |
C/                  |                       20-Feb-2006 |
C/                  +-----------------------------------+
C/
C  1. Purpose :
C
C     Calculate the dominant factor, Fd.
C
C  2. Method :
C
C     Compute Fd(spreading factor,gamma) with Fuzzy Logic.
C
C  3. Parameters :
C
C     Parameter list
C     ----------------------------------------------------------------
C       XSP,XGAM   Real   Slope-T(theta), Slope-K(wavenumber)
C       FACTD      Real   The dominant factor, Fd.
C
C     Local parameters
C     ----------------------------------------------------------------
C     See source code.
C
C  4. Subroutines used :
C
```

```
C       FINDY, DEFUZ.
C
C  5. Called by :
C
C       W3SNLB
C
C  6. Error messages :
C
C       None.
C
C  7. Remarks :
C
C       None.
C
C  8. Structure :
C
C     See source code.
C
C  9. Switches :
C
C       None.
C
C 10. Source code :
C
C/ ------------------------------------------------------------ /
      IMPLICIT NONE
C
##fzzl.cmn
C/ Local parameters
C
      INTEGER ISP,IGM
*
      REAL SP1,SP2,SP3,GM1,GM3,GM5,GM7,
     &     FD01,FD02,FD03,FD04,FD05,FD06,
     &     FD07,FD08,FD09,FD10,FD11,FD12,
     &     CSP1,CSP2,CSP3,CGM1,CGM3,CGM5,CGM7,
     &     C1,C2,C3,C4,F1,F2,F3,F4,
     &     FINDY,DEFUZ
*
      EXTERNAL FINDY,DEFUZ
```

```
*
C/ ------------------------------------------------------------ /
C FL DATA
*
C Slope of spreading factor:
      DATA SP1,SP2,SP3     /0.17276792,0.34032625,0.96153523/
*
C Slope of gamma factor:
      DATA GM1,GM3         /0.00693812,0.51239269/
      DATA GM5,GM7         /0.57823480,0.61386923/
*
C Fd(gamma,spreading factor), IPA=2
      DATA FD01,FD02,FD03,FD04  /12.5000, 9.0761, 9.5266,10.1712/
      DATA FD05,FD06,FD07,FD08  /10.3665, 8.4495, 9.1633,10.0000/
      DATA FD09,FD10,FD11,FD12  / 7.0642, 7.7174, 9.2124,10.5817/

C/ ------------------------------------------------------------ /
*
*
C=====================================================
C    Findind the magnifying factor with Fuzzy logic
C=====================================================
*
C Two inputs, one output

C Determine the domain class of speading
      IF(XSP.LE.SP1)                    ISP = 1
      IF(XSP.GT.SP1.AND.XSP.LE.SP2)     ISP = 2
      IF(XSP.GT.SP2.AND.XSP.LT.SP3)     ISP = 3
      IF(XSP.GE.SP3)                    ISP = 4
*
C Determine the domain class of gamma
      IF(XGAM.LE.GM1)                   IGM = 1
      IF(XGAM.GT.GM1.AND.XGAM.LE.GM3)   IGM = 2
      IF(XGAM.GT.GM3.AND.XGAM.LE.GM5)   IGM = 3
      IF(XGAM.GT.GM5.AND.XGAM.LT.GM7)   IGM = 4
      IF(XGAM.GE.GM7)                   IGM = 5

C Determine crip values for spreading factor
```

```
      IF(ISP.EQ.1) THEN
       CSP1 = 1.
       GO TO 444
      END IF

      IF(ISP.EQ.2) THEN
       CSP1 = FINDY(SP1,SP2,XSP)
       CSP2 = 1-CSP1
       GO TO 444
      END IF

      IF(ISP.EQ.3) THEN
       CSP2 = FINDY(SP2,SP3,XSP)
       CSP3 = 1-CSP2
       GO TO 444
      END IF

      IF(ISP.EQ.4) THEN
       CSP3 = 1.
       GO TO 444
      END IF

 444  CONTINUE

C Determine crip values for gamma factor

      IF(IGM.EQ.1) THEN
       CGM1 = 1.
       GO TO 555
      END IF

      IF(IGM.EQ.2) THEN
       CGM1 = FINDY(GM1,GM3,XGAM)
       CGM3 = 1-CGM1
       GO TO 555
      END IF

      IF(IGM.EQ.3) THEN
       CGM3 = FINDY(GM3,GM5,XGAM)
       CGM5 = 1-CGM3
```

```
      GO TO 555
     END IF

     IF(IGM.EQ.4) THEN
      CGM5 = FINDY(GM5,GM7,XGAM)
      CGM7 = 1-CGM5
      GO TO 555
     END IF

     IF(IGM.EQ.5) THEN
      CGM7 = 1.
      GO TO 555
     END IF

555  CONTINUE

C Determine rules to find the mag. factor

C First combination

     IF(ISP.EQ.1.AND.IGM.EQ.1) THEN
      FACTD = FD01
      GO TO 777
     END IF

      C3 = 0.
      C4 = 0.
      F3 = 0.
      F4 = 0.

     IF(ISP.EQ.1.AND.IGM.EQ.2) THEN
      C1 = CGM1
      C2 = CGM3
      F1 = C1*FD01
      F2 = C2*FD02
      FACTD = DEFUZ(C1,C2,C3,C4,F1,F2,F3,F4)
      GO TO 777
     END IF

     IF(ISP.EQ.1.AND.IGM.EQ.3) THEN
```

```
      C1 = CGM3
      C2 = CGM5
      F1 = C1*FD02
      F2 = C2*FD03
      FACTD = DEFUZ(C1,C2,C3,C4,F1,F2,F3,F4)
      GO TO 777
     END IF

     IF(ISP.EQ.1.AND.IGM.EQ.4) THEN
      C1 = CGM5
      C2 = CGM7
      F1 = C1*FD03
      F2 = C2*FD04
      FACTD = DEFUZ(C1,C2,C3,C4,F1,F2,F3,F4)
      GO TO 777
     END IF

     IF(ISP.EQ.1.AND.IGM.EQ.5) THEN
      FACTD = FD04
      GO TO 777
     END IF

C Second combination

     IF(ISP.EQ.2.AND.IGM.EQ.1) THEN
      C1 = CSP1
      C2 = CSP2
      C3 = 0.
      C4 = 0.
      F1 = C1*FD01
      F2 = C2*FD05
      F3 = 0.
      F4 = 0.
      FACTD = DEFUZ(C1,C2,C3,C4,F1,F2,F3,F4)
      GO TO 777
     END IF

     IF(ISP.EQ.2.AND.IGM.EQ.2) THEN
      C1 = MIN(CSP1,CGM1)
      C2 = MIN(CSP2,CGM1)
```

```
 C3 = MIN(CSP1,CGM3)
 C4 = MIN(CSP2,CGM3)
 F1 = C1*FD01
 F2 = C2*FD05
 F3 = C3*FD02
 F4 = C4*FD06
 FACTD = DEFUZ(C1,C2,C3,C4,F1,F2,F3,F4)
 GO TO 777
END IF

IF(ISP.EQ.2.AND.IGM.EQ.3) THEN
 C1 = MIN(CSP1,CGM3)
 C2 = MIN(CSP2,CGM3)
 C3 = MIN(CSP1,CGM5)
 C4 = MIN(CSP2,CGM5)
 F1 = C1*FD02
 F2 = C2*FD06
 F3 = C3*FD03
 F4 = C4*FD07
 FACTD = DEFUZ(C1,C2,C3,C4,F1,F2,F3,F4)
 GO TO 777
END IF

IF(ISP.EQ.2.AND.IGM.EQ.4) THEN
 C1 = MIN(CSP1,CGM5)
 C2 = MIN(CSP2,CGM5)
 C3 = MIN(CSP1,CGM7)
 C4 = MIN(CSP2,CGM7)
 F1 = C1*FD03
 F2 = C2*FD07
 F3 = C3*FD04
 F4 = C4*FD08
 FACTD = DEFUZ(C1,C2,C3,C4,F1,F2,F3,F4)
 GO TO 777
END IF

IF(ISP.EQ.2.AND.IGM.EQ.5) THEN
 C1 = CSP1
 C2 = CSP2
 C3 = 0.
```

```
      C4 = 0.
      F1 = C1*FD04
      F2 = C2*FD08
      F3 = 0.
      F4 = 0.
      FACTD = DEFUZ(C1,C2,C3,C4,F1,F2,F3,F4)
      GO TO 777
     END IF

C Third combination

     IF(ISP.EQ.3.AND.IGM.EQ.1) THEN
      C1 = CSP2
      C2 = CSP3
      C3 = 0.
      C4 = 0.
      F1 = C1*FD05
      F2 = C2*FD09
      F3 = 0.
      F4 = 0.
      FACTD = DEFUZ(C1,C2,C3,C4,F1,F2,F3,F4)
      GO TO 777
     END IF

     IF(ISP.EQ.3.AND.IGM.EQ.2) THEN
      C1 = MIN(CSP2,CGM1)
      C2 = MIN(CSP3,CGM1)
      C3 = MIN(CSP2,CGM3)
      C4 = MIN(CSP3,CGM3)
      F1 = C1*FD05
      F2 = C2*FD09
      F3 = C3*FD06
      F4 = C4*FD10
      FACTD = DEFUZ(C1,C2,C3,C4,F1,F2,F3,F4)
      GO TO 777
     END IF

     IF(ISP.EQ.3.AND.IGM.EQ.3) THEN
      C1 = MIN(CSP2,CGM3)
      C2 = MIN(CSP3,CGM3)
```

```
      C3 = MIN(CSP2,CGM5)
      C4 = MIN(CSP3,CGM5)
      F1 = C1*FD06
      F2 = C2*FD10
      F3 = C3*FD07
      F4 = C4*FD11
      FACTD = DEFUZ(C1,C2,C3,C4,F1,F2,F3,F4)
      GO TO 777
     END IF

     IF(ISP.EQ.3.AND.IGM.EQ.4) THEN
      C1 = MIN(CSP2,CGM5)
      C2 = MIN(CSP3,CGM5)
      C3 = MIN(CSP2,CGM7)
      C4 = MIN(CSP3,CGM7)
      F1 = C1*FD07
      F2 = C2*FD11
      F3 = C3*FD08
      F4 = C4*FD12
      FACTD = DEFUZ(C1,C2,C3,C4,F1,F2,F3,F4)
      GO TO 777
     END IF

     IF(ISP.EQ.3.AND.IGM.EQ.5) THEN
      C1 = CSP2
      C2 = CSP3
      C3 = 0.
      C4 = 0.
      F1 = C1*FD08
      F2 = C2*FD12
      F3 = 0.
      F4 = 0.
      FACTD = DEFUZ(C1,C2,C3,C4,F1,F2,F3,F4)
      GO TO 777
     END IF

C Fourth combination

     IF(ISP.EQ.4.AND.IGM.EQ.1) THEN
      FACTD = FD09
```

```
 GO TO 777
END IF

 C3 = 0.
 C4 = 0.
 F3 = 0.
 F4 = 0.

IF(ISP.EQ.4.AND.IGM.EQ.2) THEN
 C1 = CGM1
 C2 = CGM3
 F1 = C1*FD09
 F2 = C2*FD10
 FACTD = DEFUZ(C1,C2,C3,C4,F1,F2,F3,F4)
 GO TO 777
END IF

IF(ISP.EQ.4.AND.IGM.EQ.3) THEN
 C1 = CGM3
 C2 = CGM5
 F1 = C1*FD10
 F2 = C2*FD11
 FACTD = DEFUZ(C1,C2,C3,C4,F1,F2,F3,F4)
 GO TO 777
END IF

IF(ISP.EQ.4.AND.IGM.EQ.4) THEN
 C1 = CGM5
 C2 = CGM7
 F1 = C1*FD11
 F2 = C2*FD12
 FACTD = DEFUZ(C1,C2,C3,C4,F1,F2,F3,F4)
 GO TO 777
END IF

IF(ISP.EQ.4.AND.IGM.EQ.5) THEN
 FACTD = FD12
 GO TO 777
END IF
```

```
 777  CONTINUE
*
C/ End of FFDF ---------------------------------------------------- /
C/
      END
```

5. fgeob.ftn

```
C/ ------------------------------------------------------------ /
      SUBROUTINE FGEOB
##docb
C/                  +-----------------------------------+
C/                  |     FISHERIES & OCEANS CANADA     |
C/                  | BEDFORD INSTITUTE OF OCEANOGRAPHY |
C/                  |   Dartmouth, Nova Scotia, Canada  |
C/                  |                &                  |
C/                  |       DALHOUSIE UNIVERSITY        |
C/                  |    Halifax, Nova Scotia, Canada   |
C/                  |                                   |
C/                  |           Adhi Susilo             |
C/                  |                         FORTRAN 77 |
C/                  | Last update :         16-Mar-2006 |
C/                  |                       25-Dec-2005 |
C/                  +-----------------------------------+
C/
C  1. Purpose :
C
C     To computes the geometry term, Geom(x,y,p).
C
C  2. Method :
C
C     Use a special grid proposed by Tracy-Resio.
C
C  3. Parameters :
C
C     Parameter list
C     ----------------------------------------------------------------
C       Xn,Yn   Real   Components of wave number in x-y axis.
C
C     Local parameters
C     ----------------------------------------------------------------
C     See source code.
C
C     Common /CW3PHS/ Physical and algebraic constants.(##aux1.cmn)
C     ----------------------------------------------------------------
C       TPI     Real  2pi.
```

```
C       GRAV    Real  Acc. of gravity
C
C                                                          (##dims.cmn)
C                                                          (##nl4p.cmn)
C     -------------------------------------------------------------
C       NTH,NNA Int.  Number of directions.
C       NK,NNR  Int.  Number of wavenumbers.
C       NPA     Int.  Number of points.
C
C     Common /LOCI/   Loci of K2 and K4                  (##loci.cmn)
C     -------------------------------------------------------------
C       XLOCn   R.A.  Component of Kn on x-axis.
C       YLOCn   R.A.  Component of Kn on y-axis.
C       DS      R.A.  Delta S along loci.
C     -------------------------------------------------------------
C
C  4. Subroutines used :
C
C       LOCUS    Find loci of resonance.
C       CPLE     Find the coupling coeffisient.
C       DELW     Find delW.
C
C  5. Called by :
C
C       WW3_GRID Main initial subroutine.
C
C  6. Error messages :
C
C       None.
C
C  7. Remarks :
C
C       None.
C
C  8. Structure :
C
C     See source code.
C
C  9. Switches :
C
```

```
C       None.
C
C 10. Source code :
C
C/ ------------------------------------------------------------ /
      IMPLICIT NONE
C
##dims.cmn
##spar.cmn
##aux1.cmn
##nl4p.cmn
##loci.cmn
##grid.cmn
##geob.cmn
C/
C/ ------------------------------------------------------------ /
C/ Local parameters
C/
      INTEGER IRNG,IRNGM1,IA1,IA3,ISA3,IR3,IPTS,IPA,I
      INTEGER I10(18)
*
      REAL GSQ,GSR,WK0,DK,DW1,
     &     X1,X2,X3,X4,Y1,Y2,Y3,Y4,
     &     DELW,CSQ
*
      REAL DS10(0:NPA)
      EXTERNAL DELW
*
      DATA I10( 1),I10( 2),I10( 3)  /10,12,14/
      DATA I10( 4),I10( 5),I10( 6)  /16,18,20/
      DATA I10( 7),I10( 8),I10( 9)  /22,24,26/
      DATA I10(10),I10(11),I10(12)  /28,30,32/
      DATA I10(13),I10(14),I10(15)  /34,36, 2/
      DATA I10(16),I10(17),I10(18)  / 4, 6, 8/
*
C 1. Set constant variables
*
      GSQ = GRAV*GRAV
      GSR = SQRT(GRAV)
      XLM = XFR*XFR
```

```
*
      WKO = SIG(1)*SIG(1)/GRAV
*
      DO IRNG = 1,NNR
       IRNGM1 = IRNG-1
       WKA(IRNG) = SIG(IRNG)*SIG(IRNG)/GRAV
       DK=WKO*(XLM**(IRNGM1+0.5)-XLM**(IRNGM1-0.5))
       PHA(IRNG)=DK*DTH*WKA(IRNG)
      END DO
*
C 2. Compute the BASIC GEOM
*
      IR3   = 1+I3S
*
      DO IA1 = 1,NNA
       X1 = WKO*ECOS(IA1)
       Y1 = WKO*ESIN(IA1)
*
       DO ISA3 = 1,2
        IPA = IPAP
        IF(ISA3.EQ.2) IPA = IPAN
        IA3 = IA1 + IPA
         IF(IA3.GT.NNA) IA3 = IA3-NNA
         IF(IA3.LT.  1) IA3 = NNA+IA3
*
        X3 = WKA(IR3)*ECOS(IA3)
        Y3 = WKA(IR3)*ESIN(IA3)
        CALL LOCUS (X1,Y1,X3,Y3)
*
        DO ILOCI = 1,NLOCI
         IPTS = I10(ILOCI)
         X2 = XLOC2(IPTS)
         Y2 = YLOC2(IPTS)
         X4 = XLOC4(IPTS)
         Y4 = YLOC4(IPTS)
          X2BL(IA1,ISA3,ILOCI) = X2
          Y2BL(IA1,ISA3,ILOCI) = Y2
          X4BL(IA1,ISA3,ILOCI) = X4
          Y4BL(IA1,ISA3,ILOCI) = Y4
*
```

```
          DW1 = 1/(GSR*DELW(X2,Y2,X4,Y4))
          CALL CPLE(X1,Y1,X2,Y2,X3,Y3,X4,Y4,CSQ)
          IF (CSQ.LT.1.0E-20) CSQ=0.
*
          GEOM(IA1,ISA3,ILOCI) = DW1*CSQ*GSQ
*
        END DO
       END DO
      END DO
*
C 3. Compute averaged ds, H10
*
      DS10(0) = 0.
      DO I = 1,NPA
       DS10(I) = DS(I)+DS10(I-1)
      END DO

      H10 = DS10(NPA)/NLOCI
*
C/ End of FGEOB ------------------------------------------------ /
C/
      END
```

6. fgeoe.ftn

```
C/ ------------------------------------------------------------ /
      SUBROUTINE FGEOE
##docb
C/                  +-----------------------------------+
C/                  |     FISHERIES & OCEANS CANADA     |
C/                  | BEDFORD INSTITUTE OF OCEANOGRAPHY |
C/                  |   Dartmouth, Nova Scotia, Canada  |
C/                  |                &                  |
C/                  |       DALHOUSIE UNIVERSITY        |
C/                  |    Halifax, Nova Scotia, Canada   |
C/                  |                                   |
C/                  |           Adhi Susilo             |
C/                  |                          FORTRAN 77 |
C/                  | Last update :         25-Dec-2005 |
C/                  +-----------------------------------+
C/
C  1. Purpose :
C
C     To computes the geometry term, Geom(x,y,z,p)
C
C  2. Method :
C
C     Use a special grid proposed by Tracy-Resio.
C
C  3. Parameters :
C
C     Parameter list
C     ----------------------------------------------------------
C       Xn,Yn   Real   Components of wave number in x-y axis.
C
C     Local parameters
C     ----------------------------------------------------------
C     See source code.
C
C     Common /CW3PHS/ Physical and algebraic constants.(##aux1.cmn)
C     ----------------------------------------------------------
C       TPI     Real  2pi.
C       GRAV    Real  Acc. of gravity
```

```
C
C                                                          (##dims.cmn)
C                                                          (##nl4p.cmn)
C     ------------------------------------------------------------
C       NTH,NNA Int.  Number of directions.
C       NK,NNR  Int.  Number of wavenumbers.
C       NPA     Int.  Number of points.
C
C     Common /LOCI/   Loci of K2 and K4                    (##loci.cmn)
C     ------------------------------------------------------------
C       XLOCn   R.A.  Component of Kn on x-axis.
C       YLOCn   R.A.  Component of Kn on y-axis.
C       DS      R.A.  Delta S along loci.
C     ------------------------------------------------------------
C
C  4. Subroutines used :
C
C       LOCUS    Find loci of resonance.
C       CPLE     Find the coupling coeffisient.
C       DELW     Find delW.
C
C  5. Called by :
C
C       WW3_GRID Main initial subroutine.
C
C  6. Error messages :
C
C       None.
C
C  7. Remarks :
C
C       None.
C
C  8. Structure :
C
C     See source code.
C
C  9. Switches :
C
C       None.
```

```
C
C 10. Source code :
C
C/ ------------------------------------------------------------- /
      IMPLICIT NONE
C
##dims.cmn
##spar.cmn
##aux1.cmn
##nl4p.cmn
##loci.cmn
##grid.cmn
##geoe.cmn
C/
C/ ------------------------------------------------------------- /
C/ Local parameters
C/
      INTEGER IRNG,IRNGM1,IA1,IA3,IR3,IPTS
*
      REAL GSQ,GSR,WK0,DK,DW1,
     &     X1,X2,X3,X4,Y1,Y2,Y3,Y4,
     &     DELW,CSQ
*
      EXTERNAL DELW
*
C 1. Set constant variables
*
      GSQ = GRAV*GRAV
      GSR = SQRT(GRAV)
      XLM = XFR*XFR
*
      WK0 = SIG(1)*SIG(1)/GRAV
*
      DO IRNG = 1,NNR
       IRNGM1 = IRNG-1
       WKA(IRNG) = SIG(IRNG)*SIG(IRNG)/GRAV
       DK=WK0*(XLM**(IRNGM1+0.5)-XLM**(IRNGM1-0.5))
       PHA(IRNG)=DK*DTH*WKA(IRNG)
      END DO
*
```

```
C 2. Compute BASIC GEOM
*
      DO IA1 = 1,NNA
       X1 = WK0*ECOS(IA1)
       Y1 = WK0*ESIN(IA1)
*
       DO IA3 = 1,NNA
        DO IR3 = 2,NNR
        X3 = WKA(IR3)*ECOS(IA3)
        Y3 = WKA(IR3)*ESIN(IA3)
        CALL LOCUS (X1,Y1,X3,Y3)
*
        DO IPTS = 1,NPA
         X2 = XLOC2(IPTS)
         Y2 = YLOC2(IPTS)
         X4 = XLOC4(IPTS)
         Y4 = YLOC4(IPTS)
          X2BLE(IA1,IA3,IR3,IPTS) = X2
          Y2BLE(IA1,IA3,IR3,IPTS) = Y2
          X4BLE(IA1,IA3,IR3,IPTS) = X4
          Y4BLE(IA1,IA3,IR3,IPTS) = Y4
*
          DW1 = 1/(GSR*DELW(X2,Y2,X4,Y4))
          CALL CPLE(X1,Y1,X2,Y2,X3,Y3,X4,Y4,CSQ)
          IF (CSQ.LT.1.0E-20) CSQ=0.
*
          GEOME(IA1,IA3,IR3,IPTS) = DS(IPTS)*DW1*CSQ*GSQ
*
         END DO
        END DO
       END DO
      END DO
*
C/ End of FGEOE ------------------------------------------------- /
C/
      END
```

7. findn.ftn

```
C/ ------------------------------------------------------------------ /
      SUBROUTINE FINDN(X, Y, DF)
##docb
C/                  +-----------------------------------+
C/                  |     FISHERIES & OCEANS CANADA     |
C/                  | BEDFORD INSTITUTE OF OCEANOGRAPHY |
C/                  |   Dartmouth, Nova Scotia, Canada  |
C/                  |                &                  |
C/                  |       DALHOUSIE UNIVERSITY        |
C/                  |    Halifax, Nova Scotia, Canada   |
C/                  |                                   |
C/                  |           Adhi Susilo             |
C/                  |                          FORTRAN 77 |
C/                  | Last update :         25-Dec-2005 |
C/                  +-----------------------------------+
C/
C  1. Purpose :
C
C     Calculate the n(k) which is outside the grid point.
C
C  2. Method :
C
C     Compute the n(k) using weight factors.
C
C  3. Parameters :
C
C     Parameter list
C     ----------------------------------------------------------------
C       X,Y     Real  Components of wave number in x-y axis.
C       DF      Real  Action density in that point (X,Y).
C
C     Local parameters
C     ----------------------------------------------------------------
C       XLM     Real  Wavenumber multiplication factor.
C
C     Common /CW3PHS/ Physical and algebraic constants. (##aux1.cmn)
C     ----------------------------------------------------------------
C       TPI     Real  2pi.
```

```
C
C     Common /CW3SPR/ Spectral parameters.              (##spar.cmn)
C     ------------------------------------------------------------
C       XFR     Real  Frequency multiplication factor.
C
C                                                       (##dims.cmn)
C                                                       (##nl4p.cmn)
C     ------------------------------------------------------------
C       NTH,NNA Int.  Number of directions.
C       NK,NNR  Int.  Number of wavenumbers.
C
C     Common /GRID/   Physical properties in the grid.  (##grid.cmn)
C     ------------------------------------------------------------
C       DENS13  R.A.  N(k) in the grid.
C       WKA     R.A.  Wavenumber.
C       PHA     R.A.  Unit area ( k dtheta*dk )
C
C  4. Subroutines used :
C
C       None.
C
C  5. Called by :
C
C       W3SNLB, W3SNLE
C                Nonlinear interactions preprocessing program.
C
C  6. Error messages :
C
C       None.
C
C  7. Remarks :
C
C       None.
C
C  8. Structure :
C
C     See source code.
C
C  9. Switches :
C
```

```
C       None.
C
C 10. Source code :
C
C/ ------------------------------------------------------------ /
      IMPLICIT NONE
C
##aux1.cmn
##dims.cmn
##spar.cmn
##nl4p.cmn
##grid.cmn
C
C/ ------------------------------------------------------------ /
C/ Local parameters
C/
*
      INTEGER NDEG,NRING,INID,IFID,INIR,IFIR
*
      REAL X,Y,DF,R,DEG,RWKO,WA1,WA2,WR1,WR2,
     &     RASWF,DHS13,DHE13
*
      XLM = XFR*XFR
      R   = SQRT(X*X+Y*Y)
      DEG = ATAN2(Y,X)
      IF(DEG.LT.0) DEG=TPI+DEG
*
       IF(R.LT.WKA(1)) THEN
       DF = 0.
*
       ELSE IF(R.GT.WKA(NNR)) THEN
           NDEG = DEG/DTH+1
           INID = NDEG
           IFID = NDEG+1
           IF(IFID.EQ.NTH+1) IFID = 1

        WA2 = (DEG-TH(INID))/DTH
        WA1 = 1-WA2
*
        RASWF = (SQRT(R/WKA(NNR)))**(-7)
```

```
        DHS13 = DENS13(NNR,INID)*RASWF
        DHE13 = DENS13(NNR,IFID)*RASWF
        DF = WA1*DHS13+WA2*DHE13
*
       ELSE
           NDEG = DEG/DTH+1
           RWKO = R/WKA(1)
          NRING = LOG(RWKO)/LOG(XLM)+1
*
        INID = NDEG
        IFID = NDEG+1
        IF(IFID.EQ.NTH+1) IFID = 1
        INIR = NRING
        IFIR = NRING+1
*
        WA2 = (DEG-TH(INID))/DTH
        WA1 = 1-WA2
        WR2 = (R-WKA(INIR))/((XLM-1)*WKA(INIR))
        WR1 = 1-WR2
*
        DF = WR1*WA1*DENS13(INIR,INID)+WR1*WA2*DENS13(INIR,IFID)+
     &       WR2*WA1*DENS13(IFIR,INID)+WR2*WA2*DENS13(IFIR,IFID)
*
       END IF
*
C/ End of FINDN ------------------------------------------------- /
C/
       END
```

8. findy.ftn

```
C/ ------------------------------------------------------------ /
      REAL FUNCTION FINDY(X1,X2,XF)
##docb
C/                  +-----------------------------------+
C/                  |     FISHERIES & OCEANS CANADA     |
C/                  | BEDFORD INSTITUTE OF OCEANOGRAPHY |
C/                  |   Darmouth, Nova Scotia, Canada   |
C/                  |                &                  |
C/                  |       DALHOUSIE UNIVERSITY        |
C/                  |    Halifax, Nova Scotia, Canada   |
C/                  |                                   |
C/                  |           Adhi Susilo             |
C/                  |                          FORTRAN 77 |
C/                  | Last update :          20-Feb-2006 |
C/                  +-----------------------------------+
C/
C  1. Purpose :
C
C      To find Y(X) (for Fuzzy Logic)
C
C  2. Method :
C
C      Y(X) = mX + C
C
C  3. Parameters :
C
C     Parameter list
C     ------------------------------------------------------------
C       FINDY      Real   The value of Y(x).
C
C     Local parameters
C     ------------------------------------------------------------
C     See source code.
C
C  4. Subroutines used :
C
C       None.
C
```

```
C  5. Called by :
C
C       FFDFZ
C
C  6. Error messages :
C
C       None.
C
C  7. Remarks :
C
C       None.
C
C  8. Structure :
C
C     See source code.
C
C  9. Switches :
C
C       None.
C
C 10. Source code :
C
C/ ------------------------------------------------------------------ /
*
      REAL   X1,X2,XF,XM,XC
*
      XM = 1/(X1-X2)
      XC = X2/(X2-X1)
      FINDY = XM*XF+XC
C
C/ End of FINDY ---------------------------------------------------- /
C
      END
```

9. locus.ftn

```
C/ ------------------------------------------------------------------ /
      SUBROUTINE LOCUS(X1, Y1, X3, Y3)
##docb
C/                  +-----------------------------------+
C/                  |     FISHERIES & OCEANS CANADA     |
C/                  | BEDFORD INSTITUTE OF OCEANOGRAPHY |
C/                  |   Dartmouth, Nova Scotia, Canada  |
C/                  |                &                  |
C/                  |       DALHOUSIE UNIVERSITY        |
C/                  |    Halifax, Nova Scotia, Canada   |
C/                  |                                   |
C/                  |           Adhi Susilo             |
C/                  |                          FORTRAN 77 |
C/                  | Last update :         25-Dec-2005 |
C/                  +-----------------------------------+
C/
C  1. Purpose :
C
C     Find locus of K2 and K4 as function of (K1,K3).
C
C  2. Method :
C
C     Use NEWTON-RAPHSON iteration for locus solution.
C     This subroutine is from subroutine of a program called
C     windwave3.f, written by Resio.
C
C  3. Parameters :
C
C     Parameter list
C     ----------------------------------------------------------------
C       Xn,Yn   Real   Components of wave number in x-y axis.
C
C     Local parameters
C     ----------------------------------------------------------------
C     See source code.
C
C     Common /CW3PHS/ Physical and algebraic constants.(##aux1.cmn)
C     ----------------------------------------------------------------
```

```
C       TPI     Real  2pi.
C
C                                                          (##dims.cmn)
C                                                          (##nl4p.cmn)
C     ------------------------------------------------------------------
C       NTH,NNA Int.  Number of directions.
C       NK,NNR  Int.  Number of wavenumbers.
C       NPA     Int.  Number of points.
C
C     Common /LOCI/   Loci of K2 and K4                    (##loci.cmn)
C     ------------------------------------------------------------------
C       XLOCn   R.A.  Component of Kn on x-axis.
C       YLOCn   R.A.  Component of Kn on y-axis.
C       DS      R.A.  Delta S along loci.
C
C
C  4. Subroutines used :
C
C       None.
C
C  5. Called by :
C
C       FGEOB,FGEOE
C           Finding the geometry term.
C
C  6. Error messages :
C
C       None.
C
C  7. Remarks :
C
C       None.
C
C  8. Structure :
C
C     See source code.
C
C  9. Switches :
C
C       None.
```

```
C
C 10. Source code :
C
C/ ------------------------------------------------------------ /
      IMPLICIT NONE
C
##aux1.cmn
##dims.cmn
##spar.cmn
##nl4p.cmn
##loci.cmn
C
C/ Local parameters
C/
      INTEGER NPP2,NP2,ILOC,ICOMP
*
      REAL AINC,X1,Y1,X2,Y2,X3,Y3,X4,Y4,WK1,WK3,OM1,OM3,OM13,
     &     Q,QSQ,PX,PY,PSQ,PMAG,PANG,COSP,SINP,
     &     ABSQ,THETA,YY,YYXX,ANGLOC,ANGLOCP,
     &     ANGLOCN,HPOT,WKMX,WKMAX,WKMN,WKMIN,
     &     CNTR,CX,CY,DIAM,RADIUSL,ANGABS,R1,COSA,SINA,
     &     OM2,OM4,WO,RP,RM,X2P,Y2P,X4P,Y4P,
     &     X2M,Y2M,X4M,Y4M,OM2P,OM4P,OM2M,OM4M,
     &     WP,WM,ZZZ,DWDR,RN,DX2,DY2
*
      AINC=TPI/NPA
      NPP2=NPA+2
      NP2=NPA/2
      WK1=SQRT(X1*X1+Y1*Y1)
      WK3=SQRT(X3*X3+Y3*Y3)
      OM1=SQRT(WK1)
      OM3=SQRT(WK3)
      OM13=OM1-OM3
*
      Q=OM13
      QSQ=Q*Q
      PX=X1-X3
      PY=Y1-Y3
      PSQ=PX*PX+PY*PY
      PMAG=SQRT(PSQ)
```

```
      PANG=ATAN2(-PY,-PX)
      COSP=COS(PANG)
      SINP=SIN(PANG)
*
C     Solution is quadratic in sqrt(k) so answer must be squared
*
      ABSQ=ABS(Q)
*
      IF (ABSQ.LT.1.0E-10) THEN
      WKMIN=0.5*PMAG
      THETA=0.
      DO 1001 ILOC=1,NP2
      DS(ILOC)=TAN(THETA) * WKMIN /NP2
      YY=(ILOC-1)*DS(ILOC)
      YYXX=YY/WKMIN
      ANGLOC=ATAN(YYXX)
      ANGLOCP=PANG+ANGLOC
      ANGLOCN=PANG-ANGLOC
      HPOT=WKMIN/COS(ANGLOC)
      XLOC2(ILOC)=HPOT*COS(ANGLOCP)
      YLOC2(ILOC)=HPOT*SIN(ANGLOCP)
      ICOMP=1-ILOC+NPA
      XLOC2(ICOMP)=HPOT*COS(ANGLOCN)
      YLOC2(ICOMP)=HPOT*SIN(ANGLOCN)
      XLOC4(ILOC)=XLOC2(ILOC)+PX
      YLOC4(ILOC)=YLOC2(ILOC)+PY
      XLOC4(ICOMP)=XLOC2(ICOMP)+PX
      YLOC4(ICOMP)=YLOC2(ICOMP)+PY
      DS(ICOMP)=DS(ILOC)
 1001 CONTINUE
*
      ELSE
      WKMX=(-PMAG-QSQ)/(2.*Q)
      WKMAX=WKMX*WKMX
      WKMN=0.5*(-Q+SQRT(2.0*PMAG-QSQ))
      WKMIN=WKMN*WKMN
      XLOC2(1)=WKMIN*COSP
      YLOC2(1)=WKMIN*SINP
      XLOC2(NP2+1)=WKMAX*COSP
      YLOC2(NP2+1)=WKMAX*SINP
```

```
      XLOC4(1)=XLOC2(1)+PX
      YLOC4(1)=YLOC2(1)+PY
      XLOC4(NP2+1)=XLOC2(NP2+1)+PX
      YLOC4(NP2+1)=YLOC2(NP2+1)+PY
      CNTR=0.5*(WKMAX+WKMIN)
      CX=CNTR*COSP
      CY=CNTR*SINP
      DIAM=WKMAX-WKMIN
      RADIUSL=0.5*DIAM
      DS(1)=RADIUSL*AINC
      DS(NP2+1)=RADIUSL*AINC
*
      DO 1 ILOC=2,NP2
      ANGLOC=(ILOC-1)*AINC
      ANGABS=ANGLOC+PANG
      R1=RADIUSL
      COSA=COS(ANGABS)
      SINA=SIN(ANGABS)
   10 X2=CX-R1*COSA
      Y2=CY-R1*SINA
      X4=X2+PX
      Y4=Y2+PY
      OM2=(X2*X2+Y2*Y2)**0.25
      OM4=(X4*X4+Y4*Y4)**0.25
      W0=OM13+OM2-OM4
      IF (ABS(W0).GT.1.0E-6) THEN
       RP=1.001*R1
       RM=0.999*R1
       X2P=CX-RP*COSA
       Y2P=CY-RP*SINA
       X4P=X2P+PX
       Y4P=Y2P+PY
       X2M=CX-RM*COSA
       Y2M=CY-RM*SINA
       X4M=X2M+PX
       Y4M=Y2M+PY
       OM2P=(X2P*X2P+Y2P*Y2P)**0.25
       OM4P=(X4P*X4P+Y4P*Y4P)**0.25
       OM2M=(X2M*X2M+Y2M*Y2M)**0.25
       OM4M=(X4M*X4M+Y4M*Y4M)**0.25
```

```
        WP=OM13+OM2P-OM4P
        WM=OM13+OM2M-OM4M
        ZZZ=(WP-WM)
        DWDR=ZZZ/(RP-RM)
        IF (ABS(DWDR).LT.1.0E-10) DWDR=1.0E-10*ABS(DWDR)/DWDR
        RN=R1-WO/DWDR
        IF (ABS(RN-R1).LT.0.0001) THEN
          CONTINUE
        ELSE
        R1=RN
        GO TO 10
        END IF
       ELSE
        RN=R1
       END IF
*
       DX2=RN*COSA
       DY2=RN*SINA
       XLOC2(ILOC)=CX-DX2
       YLOC2(ILOC)=CY-DY2
       ICOMP=NPP2-ILOC
       XLOC2(ICOMP)=CX-RN*COS(PANG-ANGLOC)
       YLOC2(ICOMP)=CY-RN*SIN(PANG-ANGLOC)
       XLOC4(ILOC)=XLOC2(ILOC)+PX
       YLOC4(ILOC)=YLOC2(ILOC)+PY
       XLOC4(ICOMP)=XLOC2(ICOMP)+PX
       YLOC4(ICOMP)=YLOC2(ICOMP)+PY
       DS(ILOC)=RN*AINC
       DS(ICOMP)=RN*AINC
     1 CONTINUE
*
       END IF
*
C/ End of LOCUS ------------------------------------------------- /
C/
       END
```

10. pten.ftn

```
C/ ------------------------------------------------------------ /
      SUBROUTINE PTEN (HNC, FS10, TRNC, DTRNC)
##docb
C/                  +----------------------------------+
C/                  |     FISHERIES & OCEANS CANADA    |
C/                  | BEDFORD INSTITUTE OF OCEANOGRAPHY |
C/                  |   Darmouth, Nova Scotia, Canada  |
C/                  |                &                 |
C/                  |       DALHOUSIE UNIVERSITY       |
C/                  |    Halifax, Nova Scotia, Canada  |
C/                  |                                  |
C/                  |           Adhi Susilo            |
C/                  |                         FORTRAN 77 |
C/                  | Last update :         11-Apr-2006 |
C/                  +----------------------------------+
C/
C  1. Purpose :
C
C     Compute the integral along loci.
C
C  2. Method :
C
C     Calculate the integral with Newton-Cotes method (10 points).
C
C  3. Parameters :
C
C     Parameter list
C     ------------------------------------------------------------
C       FS10       Real   power of geometry term
C       TRNC,DTRNC Real   Integral of Transfer & Diagonal function
C
C     Local parameters
C     ------------------------------------------------------------
C     See source code.
C
C  4. Subroutines used :
C
C       None.
```

```
C
C  5. Called by :
C
C       W3SNLB
C
C  6. Error messages :
C
C       None.
C
C  7. Remarks :
C
C       None.
C
C  8. Structure :
C
C     See source code.
C
C  9. Switches :
C
C       None.
C
C 10. Source code :
C
C/ ------------------------------------------------------------------ /
      IMPLICIT NONE
C
##dims.cmn
##nl4p.cmn
##nc10.cmn
C/ Local parameters
C
*
      REAL WI0,WI1,WI2,WI3,WI4,WI5,
     &     HNC,FS10,TRNC,DTRNC
*
C/ ------------------------------------------------------------------ /
*
C/ Weight factors
C     WI0 = 9/89600.
```

```
      WI0 = 0.000100446
      WI1 = 2857
      WI2 = 15741
      WI3 = 1080
      WI4 = 19344
      WI5 = 5778
*
Compute the integral using 2x10 points method
*
      TRNC = 0.
      TRNC = TRNC + WI0*FS10*HNC*
     &        ( WI1*(TR10(1)+TR10(10))+
     &          WI2*(TR10(2)+TR10( 9))+
     &          WI3*(TR10(3)+TR10( 8))+
     &          WI4*(TR10(4)+TR10( 7))+
     &          WI5*(TR10(5)+TR10( 6)) )
      TRNC = TRNC + WI0*FS10*HNC*
     &        ( WI1*(TR10(10)+TR10( 1))+
     &          WI2*(TR10(11)+TR10(18))+
     &          WI3*(TR10(12)+TR10(17))+
     &          WI4*(TR10(13)+TR10(16))+
     &          WI5*(TR10(14)+TR10(15)) )
*
      DTRNC = 0.
      DTRNC = DTRNC + WI0*FS10*HNC*
     &        ( WI1*(DG10(1)+DG10(10))+
     &          WI2*(DG10(2)+DG10( 9))+
     &          WI3*(DG10(3)+DG10( 8))+
     &          WI4*(DG10(4)+DG10( 7))+
     &          WI5*(DG10(5)+DG10( 6)) )
      DTRNC = DTRNC + WI0*FS10*HNC*
     &        ( WI1*(DG10(10)+DG10( 1))+
     &          WI2*(DG10(11)+DG10(18))+
     &          WI3*(DG10(12)+DG10(17))+
     &          WI4*(DG10(13)+DG10(16))+
     &          WI5*(DG10(14)+DG10(15)) )
*
C/ End of PTEN ------------------------------------------------- /
C/
      END
```

11. w3snlb.ftn

```
C/ ------------------------------------------------------------------ /
      SUBROUTINE W3SNLB(A, CG, UREF, S, D)
##docb
C/                  +-----------------------------------+
C/                  |     FISHERIES & OCEANS CANADA     |
C/                  | BEDFORD INSTITUTE OF OCEANOGRAPHY |
C/                  |   Dartmouth, Nova Scotia, Canada  |
C/                  |                &                  |
C/                  |       DALHOUSIE UNIVERSITY        |
C/                  |    Halifax, Nova Scotia, Canada   |
C/                  |                                   |
C/                  |           Adhi Susilo             |
C/                  |                          FORTRAN 77 |
C/                  | Last update :          24-Jul-2006 |
C/                  |                        21-Feb-2006 |
C/                  +-----------------------------------+
C/
C  1. Purpose :
C
C     Calculate nonlinear interactions and the diagonal term of
C     its derivative.
C
C  2. Method :
C
C     Boltzmann integral with dominant tranfer and fuzzy logic,
C     AvDI (Advance Dominant Interaction) method.
C
C  3. Parameters :
C
C     Parameter list
C     ----------------------------------------------------------------
C       A       R.A.  I   Action spectrum A(ITH,IK) as a function of
C                         direction (rad)  and wavenumber.
C       CG      R.A.  I   Group velocities (dimension NK).
C       S       R.A.  O   Source term.                            *)
C       D       R.A.  O   Diagonal term of derivative.            *)
C     ----------------------------------------------------------------
C                            *) 1-D array with dimension NTH*NK
```

```
C
C     Common /GRID  / Nonlinear interactions            (##grid.cmn)
C     ------------------------------------------------------------
C       DENS13 R.A.  Action density n(k).
C       WKA    R.A.  Magnitute of wavenumber.
C       PHA    R.A.  kdtheta x dk.
C
C     Common /GEOB  / Nonlinear interactions            (##geob.cmn)
C     ------------------------------------------------------------
C       GEOM   R.A.  Geometry term, G(k).
C       XnBL   R.A.  Length of locus 'n' on X-axis.
C       YnBL   R.A.  Length of locus 'n' on Y-axis.
C     ------------------------------------------------------------
C
C     Local parameters
C     ------------------------------------------------------------
C     See source code.
C
C  4. Subroutines used :
C
C       FINDN    Finding the action density n(k).
C       FFDFZ    Finding the dominant factor by fuzzy logic.
C       PTEN     Finding the integral along loci by fuzzy Newton-Cotes.
C
C  5. Called by :
C
C       W3SRCE   Source term integration.
C
C  6. Error messages :
C
C       None.
C
C  7. Remarks :
C
C       None.
C
C  8. Structure :
C
C     See source code.
C
```

```
C  9. Switches :
C
C       None.
C
C 10. Source code :
C
C/ ------------------------------------------------------------------ /
      IMPLICIT NONE
C
##aux1.cmn
##dims.cmn
##spar.cmn
##nl4p.cmn
##grid.cmn
##geob.cmn
##fzzl.cmn
##nc10.cmn
C/
C/ ------------------------------------------------------------------ /
C/ Local parameters
C/
      INTEGER I3D,I,J,IANG,IRNG,IA1,IA3,IR1,IR3,IETA,IJ,
     &     IR14,IR24,IMAX,JMAX,IMAXP1,JMAXP1,ISA3,IPA,IMIN

*
      REAL X1,Y1,X2,Y2,X3,Y3,X4,Y4,VN1,VN2,VN3,VN4,V1T3,V3M1,
     &    DNDT1(NNR,NNA),DNDT2(NNR,NNA),DIAG1(NNR,NNA),DIAG2(NNR,NNA),
     &    FS,ETAP,DIF13,DIF14,TR,DTR,PUMP,DIFF,DTOTAL,DIATTL,
     &    TRAUX,DGAUX,T13,D13,DNDTAUX1,DNDTAUX2,DIAGAUX1,DIAGAUX2,
     &    A(NTH,NK),S(NSPEC),D(NSPEC),CG(NK),CONV(NK),CONN(NK),
     &    REFMAX,DMAX,FMAG,FGRID,SNL4(NNR,NNA),DNL4(NNR,NNA),
     &    DSL1,DSL2,UREF,EPMREF,ESCALE,MYPEAK,WHMIN
*
C 1. Set constant variables
*
      FGRID = 0.375
      XLM = XFR*XFR
      EPMREF = 3.64e-3*(UREF**4)/(GRAV*GRAV)
*
C 2. Set action desity (n(k))
```

```
*
      DO I = 1,NNR
      CONV(I) = TPIINV / SIG(I) * CG(I)
      CONN(I) = GRAV/(SIG(I)*SIG(I))
       DO J = 1,NNA
        DENS13(I,J) = A(J,I)*CONN(I)
        IF(DENS13(I,J).LT.1.E-20) DENS13(I,J)=0.
       END DO
      END DO

C Finding max 2density
      REFMAX = 0.
      DO I = 1,NNR
       DO J = 1,NNA
        DMAX = MAX(REFMAX,DENS13(I,J))
        IF(DMAX.EQ.DENS13(I,J)) THEN
          IMAX = I
          JMAX = J
          REFMAX = DENS13(I,J)
        END IF
       END DO
      END DO

      ESCALE = A(JMAX,IMAX)/EPMREF
      IF(ESCALE.LT.2.5) THEN
       ESCALE = 1.0
      ELSE
       ESCALE = 0.6
      END IF

C 1 step:
      IMAXP1 = IMAX + 1
      JMAXP1 = JMAX + 1
      IF(IMAX.EQ.NNR) IMAXP1 = NNR
      IF(JMAX.EQ.NNA) JMAXP1 = 1

C Finding the slope for wave number & angle of direction,
C a normalised slope.    (1 step)
      XSP  = (1-DENS13(IMAX,JMAXP1)/DENS13(IMAX,JMAX))
      XSP  = XSP/(DTH*WKA(IMAX))
```

```
      XGAM = (1-DENS13(IMAXP1,JMAX)/DENS13(IMAX,JMAX))
      XGAM = XGAM/((XLM-1)*WKA(IMAX))

C Finding the dominant factor, Fd
      CALL FFDFZ(XSP, XGAM, FACTD)

C Magnifying factor:
      FMAG = FGRID*FACTD*ESCALE

C Initial dndt & diagonal term

      DO IRNG = 1,NNR
       DO IANG = 1,NNA
        DNDT1(IRNG,IANG) = 0.
        DNDT2(IRNG,IANG) = 0.
        DIAG1(IRNG,IANG) = 0.
        DIAG2(IRNG,IANG) = 0.
       END DO
      END DO
*
C 3. Do the integration
*
C 1st loop
      DO IA1 = 1,NNA
*
C 2nd loop
       DO ISA3 = 1,2
        IPA = IPAP
        IF(ISA3.EQ. 2) IPA = IPAN
        IA3 = IA1+IPA
        IF(IA3.GT.NNA) IA3 = IA3-NNA
        IF(IA3.LT.  1) IA3 = NNA+IA3

*
C 3rd loop
        DO IR1 = 1,NNR-I3S
          I3D = IR1+I3S
         IETA = (IR1-1)
*
         X1 = WKA(IR1)*ECOS(IA1)
```

```
          Y1 = WKA(IR1)*ESIN(IA1)
          X3 = WKA(I3D)*ECOS(IA3)
          Y3 = WKA(I3D)*ESIN(IA3)
*
         VN1 = DENS13(IR1,IA1)
         VN3 = DENS13(I3D,IA3)
        V1T3 = VN1*VN3
        V3M1 = VN3-VN1
*
          FS = XLM**IETA
        ETAP = IETA*6.5
       DIF13 = (X1-X3)*(X1-X3)+(Y1-Y3)*(Y1-Y3)
*
C 4th loop, along loci
          DO ILOCI = 1,NLOCI
           X2 = X2BL(IA1,ISA3,ILOCI)*FS
           Y2 = Y2BL(IA1,ISA3,ILOCI)*FS
           X4 = X4BL(IA1,ISA3,ILOCI)*FS
           Y4 = Y4BL(IA1,ISA3,ILOCI)*FS
*
           DIF14 = (X1-X4)*(X1-X4)+(Y1-Y4)*(Y1-Y4)
           IF(DIF13.GE.DIF14) GO TO 1111
*
           CALL FINDN(X2,Y2,VN2)
           CALL FINDN(X4,Y4,VN4)
*
           PUMP = V1T3*(VN4-VN2)
           DIFF = VN2*VN4*V3M1
         DTOTAL = PUMP+DIFF
         DIATTL = VN3*(VN4-VN2)-VN2*VN4
*
          TR10(ILOCI) = DTOTAL*GEOM(IA1,ISA3,ILOCI)*XLM**ETAP
          DG10(ILOCI) = DIATTL*GEOM(IA1,ISA3,ILOCI)*XLM**ETAP
*
 1111 CONTINUE
          END DO
*
          CALL PTEN(H10,FS,TR,DTR)
          T13 = 2*TR*FMAG
          D13 = 2*DTR*FMAG
```

```
*
          DNDTAUX1 = T13*PHA(I3D)
          DNDTAUX2 = T13*PHA(IR1)
          DIAGAUX1 = D13*PHA(I3D)
          DIAGAUX2 = D13*PHA(IR1)
*
          DNDT1(IR1,IA1) = DNDT1(IR1,IA1)+DNDTAUX1
          DNDT2(I3D,IA3) = DNDT2(I3D,IA3)-DNDTAUX2
          DIAG1(IR1,IA1) = DIAG1(IR1,IA1)+DIAGAUX1
          DIAG2(I3D,IA3) = DIAG2(I3D,IA3)-DIAGAUX2
*
         END DO
        END DO
       END DO
*
C 4. Stabilize Snl
*
       WHMIN = 0.
       DO I=1,NNR
        DO J=1,NNA
         SNL4(I,J) = DNDT1(I,J)+DNDT2(I,J)
         DNL4(I,J) = DIAG1(I,J)+DIAG2(I,J)
         WHMIN = MIN(WHMIN,SNL4(I,J))
         IF( WHMIN.EQ.SNL4(I,J) ) THEN
          IMIN = I
          WHMIN = SNL4(I,J)
         END IF
        END DO
       END DO

       DO I=IMIN+1,NNR
        DO J=1,NNA
        IF( SNL4(I,J).GT.0. ) SNL4(I,J) = 0.
        END DO
       END DO
*
C 5. Transfer 2D Snl&D into 1D Snl&D
*
       DO I=1,NNR
        DO J=1,NNA
```

```
          IJ = J + (I-1)*NNA
C S requires m^3, output is m^4, so:
       S(IJ)= SNL4(I,J)/CONN(I)
       D(IJ)= DNL4(I,J)
       END DO
      END DO
*
C/ End of W3SNLB ------------------------------------------------------ /
C/
      END
```

12. w3snle.ftn

```
C/ ------------------------------------------------------------ /
      SUBROUTINE W3SNLE(A, CG, S, D)
##docb
C/                  +----------------------------------+
C/                  |     FISHERIES & OCEANS CANADA    |
C/                  | BEDFORD INSTITUTE OF OCEANOGRAPHY |
C/                  |   Dartmouth, Nova Scotia, Canada  |
C/                  |                &                 |
C/                  |       DALHOUSIE UNIVERSITY       |
C/                  |    Halifax, Nova Scotia, Canada   |
C/                  |                                  |
C/                  |           Adhi Susilo            |
C/                  |                         FORTRAN 77 |
C/                  | Last update :         25-Dec-2005 |
C/                  +----------------------------------+
C/
C  1. Purpose :
C
C     Calculate nonlinear interactions and the diagonal term of
C     its derivative.
C
C  2. Method :
C
C     Boltzmann integral (WTR method).
C
C  3. Parameters :
C
C     Parameter list
C     ----------------------------------------------------------------
C       A       R.A.  I   Action spectrum A(ITH,IK) as a function of
C                         direction (rad)  and wavenumber.
C       CG      R.A.  I   Group velocities (dimension NK).
C       S       R.A.  O   Source term.                            *)
C       D       R.A.  O   Diagonal term of derivative.            *)
C     ----------------------------------------------------------------
C                             *) 1-D array with dimension NTH*NK
C
C     Common /GRID  / Nonlinear interactions            (##grid.cmn)
```

```
C     -------------------------------------------------------------
C       DENS13  R.A.  Action density n(k).
C       WKA     R.A.  Magnitute of wavenumber.
C       PHA     R.A.  kdtheta x dk.
C
C     Common /GEOE  / Nonlinear interactions           (##geoe.cmn)
C     -------------------------------------------------------------
C       GEOME   R.A.  Geometry term, G(k).
C       XnBLE   R.A.  Length of locus 'n' on X-axis.
C       YnBLE   R.A.  Length of locus 'n' on X-axis.
C     -------------------------------------------------------------
C
C     Local parameters
C     -------------------------------------------------------------
C     See source code.
C
C  4. Subroutines used :
C
C       FINDN    Finding the action density n(k).
C
C  5. Called by :
C
C       W3SRCE   Source term integration.
C
C  6. Error messages :
C
C       None.
C
C  7. Remarks :
C
C       None.
C
C  8. Structure :
C
C     See source code.
C
C  9. Switches :
C
C       None.
C
```

```
C 10. Source code :
C
C/ ------------------------------------------------------------ /
      IMPLICIT NONE
C
##aux1.cmn
##dims.cmn
##spar.cmn
##nl4p.cmn
##grid.cmn
##geoe.cmn
C/
C/ ------------------------------------------------------------ /
C/ Local parameters
C/
      INTEGER I3D,I,J,IANG,IRNG,IA1,IA3,IR1,IR3,IETA,IPTS,IJ,
     &  IFINISH,IRD
*
      REAL X1,Y1,X2,Y2,X3,Y3,X4,Y4,VN1,VN2,VN3,VN4,V1T3,V3M1,
     &  DNDT1(NNR,NNA),DNDT2(NNR,NNA),DIAG1(NNR,NNA),DIAG2(NNR,NNA),
     &  FS,ETAP,DIF13,DIF14,TR,DTR,PUMP,DIFF,DTOTAL,DIATTL,
     &  TRAUX,DGAUX,T13,D13,DNDTAUX1,DNDTAUX2,DIAGAUX1,DIAGAUX2,
     &  A(NTH,NK),S(NSPEC),D(NSPEC),CG(NK),CONV(NK),CONN(NK)
*
C 1. Set constant variable
*
      XLM = XFR*XFR
*
C 2. Set action desity (n(k))
*
      DO I = 1,NNR
      CONV(I) = TPIINV / SIG(I) * CG(I)
      CONN(I) = GRAV/(SIG(I)*SIG(I))
       DO J = 1,NNA
        DENS13(I,J) = A(J,I)*CONN(I)
        IF(DENS13(I,J).LT.1.E-20) DENS13(I,J)=0.
       END DO
      END DO
*
C Initial dndt & diagonal term
```

```
      DO IRNG = 1,NNR
       DO IANG = 1,NNA
        DNDT1(IRNG,IANG) = 0.
        DNDT2(IRNG,IANG) = 0.
        DIAG1(IRNG,IANG) = 0.
        DIAG2(IRNG,IANG) = 0.
       END DO
      END DO
*
C 3. Do the integration
*
C 1st loop
      DO IA1 = 1,NNA
*
C 2nd loop
       DO IA3 = 1,NNA
*
C 3rd loop
        DO IR3 = 2,NNR
        IFINISH = NNR+1-IR3
        IRD = IR3-1
*
C 4th loop
        DO IR1 = 1,IFINISH
          I3D = IR1+IRD
         IETA = (IR1-1)
*
         X1 = WKA(IR1)*ECOS(IA1)
         Y1 = WKA(IR1)*ESIN(IA1)
         X3 = WKA(I3D)*ECOS(IA3)
         Y3 = WKA(I3D)*ESIN(IA3)
*
        VN1 = DENS13(IR1,IA1)
        VN3 = DENS13(I3D,IA3)
       V1T3 = VN1*VN3
       V3M1 = VN3-VN1
*
         FS = XLM**IETA
       ETAP = IETA*7.5
```

```
      DIF13 = (X1-X3)*(X1-X3)+(Y1-Y3)*(Y1-Y3)
         TR = 0.
        DTR = 0.
*
C 5th loop, along loci
         DO IPTS = 1,NPA
          X2 = X2BLE(IA1,IA3,IR3,IPTS)*FS
          Y2 = Y2BLE(IA1,IA3,IR3,IPTS)*FS
          X4 = X4BLE(IA1,IA3,IR3,IPTS)*FS
          Y4 = Y4BLE(IA1,IA3,IR3,IPTS)*FS
*
          DIF14 = (X1-X4)*(X1-X4)+(Y1-Y4)*(Y1-Y4)
          IF(DIF13.GE.DIF14) GO TO 1111
*
          CALL FINDN(X2,Y2,VN2)
          CALL FINDN(X4,Y4,VN4)
*
          PUMP = V1T3*(VN4-VN2)
          DIFF = VN2*VN4*V3M1
        DTOTAL = PUMP+DIFF
        DIATTL = VN3*(VN4-VN2)-VN2*VN4
*
         TRAUX = DTOTAL*GEOME(IA1,IA3,IR3,IPTS)*XLM**ETAP
         DGAUX = DIATTL*GEOME(IA1,IA3,IR3,IPTS)*XLM**ETAP
*
            TR = TR + TRAUX
           DTR = DTR + DGAUX
*
 1111 CONTINUE
         END DO
*
         T13 = TR
         D13 = DTR
         DNDTAUX1 = T13*PHA(I3D)
         DNDTAUX2 = T13*PHA(IR1)
         DIAGAUX1 = D13*PHA(I3D)
         DIAGAUX2 = D13*PHA(IR1)
*
         DNDT1(IR1,IA1) = DNDT1(IR1,IA1)+DNDTAUX1
         DNDT2(I3D,IA3) = DNDT2(I3D,IA3)-DNDTAUX2
```

```
          DIAG1(IR1,IA1) = DIAG1(IR1,IA1)+DIAGAUX1
          DIAG2(I3D,IA3) = DIAG2(I3D,IA3)-DIAGAUX2
*
          END DO
         END DO
        END DO
       END DO
*
C 4. Transfer 2D Snl&D into 1D Snl&D
*
       DO I=1,NNR
        DO J=1,NNA
           IJ = J + (I-1)*NNA
         S(IJ)=(DNDT1(I,J)+DNDT2(I,J))/CONN(I)
         D(IJ)= DIAG1(I,J)+DIAG2(I,J)
        END DO
       END DO
*
*
C/ End of W3SNLE ------------------------------------------------ /
C/
       END
```