



Scientific Excellence • Resource Protection & Conservation • Benefits for Canadians
Excellence scientifique • Protection et conservation des ressources • Bénéfices aux Canadiens

146210

AN ACCURATE EFFICIENT NON-LINEAR CODE FOR WAVE MODELLING

William Perrie and Bechara Toulany

Physical and Chemical Sciences Branch
Scotia-Fundy Region
Department of Fisheries and Oceans

Bedford Institute of Oceanography
P.O. Box 1006
Dartmouth, Nova Scotia
Canada B2Y 4A2

1993

Canadian Technical Report of
Hydrography and Ocean Sciences
No. 146

1993-08



Fisheries
and Oceans

Pêches
et Océans

Canada

Canadian Technical Report of Hydrography and Ocean Sciences

Technical reports contain scientific and technical information that contributes to existing knowledge but which is not normally appropriate for primary literature. The subject matter is related generally to programs and interests of the Ocean Science and Surveys (OSS) sector of the Department of Fisheries and Oceans.

Technical reports may be cited as full publications. The correct citation appears above the abstract of each report. Each report is abstracted in *Aquatic Sciences and Fisheries Abstracts* and indexed in the Department's annual index to scientific and technical publications.

Technical reports are produced regionally but are numbered nationally. Requests for individual reports will be filled by the issuing establishment listed on the front cover and title page. Out of stock reports will be supplied for a fee by commercial agents.

Regional and headquarters establishments of Ocean Science and Surveys ceased publication of their various report series as of December 1981. A complete listing of these publications is published in the *Canadian Journal of Fisheries and Aquatic Sciences*, Volume 39: Index to Publications 1982. The current series, which begins with report number 1, was initiated in January 1982.

Rapport technique canadien sur l'hydrographie et les sciences océaniques

Les rapports techniques contiennent des renseignements scientifiques et techniques qui constituent une contribution aux connaissances actuelles, mais qui ne sont pas normalement appropriés pour la publication dans un journal scientifique. Le sujet est généralement lié aux programmes et intérêts du service des Sciences et levés océaniques (SLO) du ministère des Pêches et des Océans.

Les rapports techniques peuvent être cités comme des publications complètes. Le titre exact paraît au-dessus du résumé de chaque rapport. Les rapports techniques sont résumés dans la revue *Résumés des sciences aquatiques et halieutiques*, et ils sont classés dans l'index annuel des publications scientifiques et techniques du Ministère.

Les rapports techniques sont produits à l'échelon régional, mais numérotés à l'échelon national. Les demandes de rapports seront satisfaites par l'établissement auteur dont le nom figure sur la couverture et la page du titre. Les rapports épuisés seront fournis contre rétribution par des agents commerciaux.

Les établissements des Sciences et levés océaniques dans les régions et à l'administration centrale ont cessé de publier leurs diverses séries de rapports en décembre 1981. Une liste complète de ces publications figure dans le volume 39, Index des publications 1982 du *Journal canadien des sciences halieutiques et aquatiques*. La série actuelle a commencé avec la publication du rapport numéro 1 en janvier 1982.

Canadian Technical Report of

Hydrography and Ocean Sciences No. 146

1993

AN ACCURATE EFFICIENT NONLINEAR CODE FOR WAVE MODELLING

by

William Perrie and Bechara Toulany

Physical and Chemical Sciences
Scotia-Fundy Region
Department of Fisheries and Oceans

Bedford Institute of Oceanography
Dartmouth, Nova Scotia, Canada
B2Y 4A2

© Minister of Supply and Services 1993
Cat. No. Fs 97-18/146E ISSN 0711-6764

Correct citation for this publication:

Perrie, W. and Toulany B. 1993. An Accurate Efficient Nonlinear Code for Wave Modelling. Can Tech. Rep. Hydrogr. Ocean Sci. No. 146: iv + 54 pp.

Table of Contents

ABSTRACT/RÉSUMÉ	iv
1. INTRODUCTION	1
2. PROGRAM STRUCTURE	1
3. FLOW CHARTS	3
4. LIST OF VARIABLES AND SUBROUTINES	11
5. THE PROGRAM AND ITS FUNCTIONAL ELEMENTS	28
6. INPUT EXAMPLE WITH OUTPUT RESULTS	51
7. PRESENT MODEL STATUS AND FUTURE PLANS	55
8. ACKNOWLEDGEMENT	55
9. REFERENCES	55
10. CAPTION FOR FIGURE 6	56

Abstract

Perrie, W. and Toulany B. 1993. An Accurate Efficient Nonlinear Code for Wave Modelling. Can Tech. Rep. Hydrogr. Ocean Sci. No. 146: iv + 54 pp.

We present documentation and code for a new 'fully' nonlinear wave model. The model is accurate and relatively economical to run. The basis for the simulation of the nonlinear wave-wave interactions are the Boltzmann integrals of Hasselmann (1961), using the algorithms which implement the symmetry rules and coordinate transformations documented by Tracy and Resio (1982) and recently exploited by Resio and Perrie (1991) and Perrie and Resio (1993).

Résumé

Perrie, W. and Toulany B. 1993. An Accurate Efficient Nonlinear Code for Wave Modelling. Can Tech. Rep. Hydrogr. Ocean Sci. No. 146: iv + 54 pp.

Nous présentons ici des explications et le code relatifs à un nouveau modèle de vagues, «entièrement» non linéaire. Ce modèle est exact et d'utilisation relativement économique. La simulation des interactions vagues-vagues non linéaires est fondée sur les intégrales Boltzmann de Hasselmann (1961) et sur les algorithmes qui appliquent les règles de symétrie et les transformations de coordonnées établies par Tracy et Resio (1982) et récemment exploitées par Resio et Perrie (1991) et par Perrie et Resio (1993).

1. Introduction

It may be asked, why is there a need for yet another 'third' generation (3g) wave model. The WAM model has been presented by Hasselmann et al (1989) and attempts to accurately simulate ocean surface wave spectra. However, it is precisely because of the WAM model that we were motivated in construction of this present attempt at a wave model. The WAM model has its inadequacies, and we attempt in this model to remove some of these inadequacies. Essentially, this model that we present, called 'FULL', has a much more careful treatment of the nonlinear wave-wave interactions than is possible in the so-called direct interaction approximation (DIA) of the WAM model. That is the distinctive of this model. The basis of the model is laid in Tracy and Resio (1982). It is exploited and developed in Resio and Perrie (1991) and Perrie and Resio (1993). We do not repeat the material given in these papers in this report. We only document the code as it presently stands. The main architect of this present code is Don Resio. This document is a description of the code and it is hoped that this description is complete enough to allow unfamiliar users easy access to an understanding and application of wave model FULL.

2. Program Structure

The overall structure of the program is given in this section. Details of the items of the flow chart are given below. Input-output to the 3g model FULL, is of the form

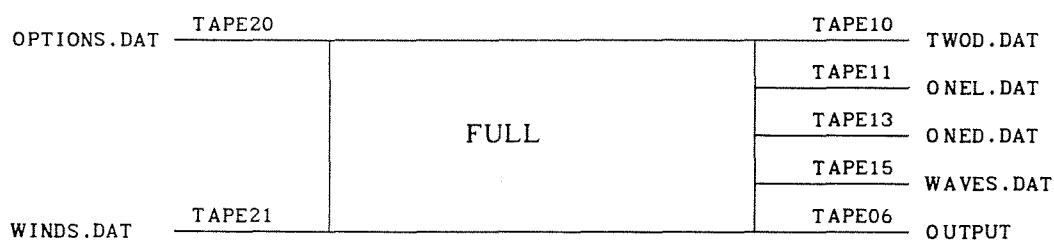


Figure 1. The input-output of the wave model FULL

whereas the detailed structure of FULL, in relation to its subroutines, has the form,

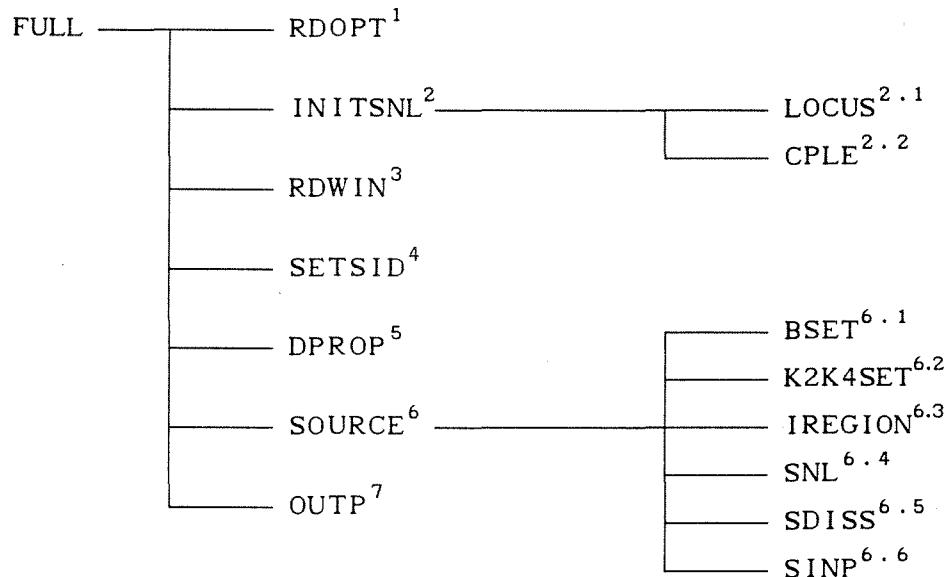


Figure 2. The overall program structure to FULL.

The particular function of the subroutines depicted in Figure 1. may be summarized as follows:

1. RDOPT: reads input options and the grid from TAPE20.
2. INITSNL: initializes SNL- related parameters
 - 2.1 LOCUS: computes the interaction locus for SNL computations
 - 2.2 CPLE: computes the coupling coefficient for SNL
3. RDWIN: reads real winds from input TAPE21, if NORD =0. If NORD = 1, test winds are provided by TAPE20.
4. SETSID: sets boundaries for special test case, if NORD=1.
5. DPRPROP: propagation routine
6. SOURCE: calculates source terms and writes to output tapes 10 and 13.
 - 6.1 BSET: sets the energy in the parametric region and adds it to the

discrete region to form the total energy matrix that is input to the Boltzman integral SNL

- 6.2 K2K4SET: sets the energy in the fine mesh K2-K4 grid
 - 6.3 IREGION: sets the region of K1-K3 space that must be included in the evaluation of SNL
 - 6.4 SNL: uses the information from INITSLN, BSET, K2K4SET and IREGION to estimate the SNL source term
 - 6.5 SDISS: estimates the wave-breaking dissipation term
 - 6.6 SINP: estimates the wind input source term
7. OUTP: writes results to output tapes 11 and 15

3. Flow Charts:

We present the following flow chart for the overall algorithm structure.

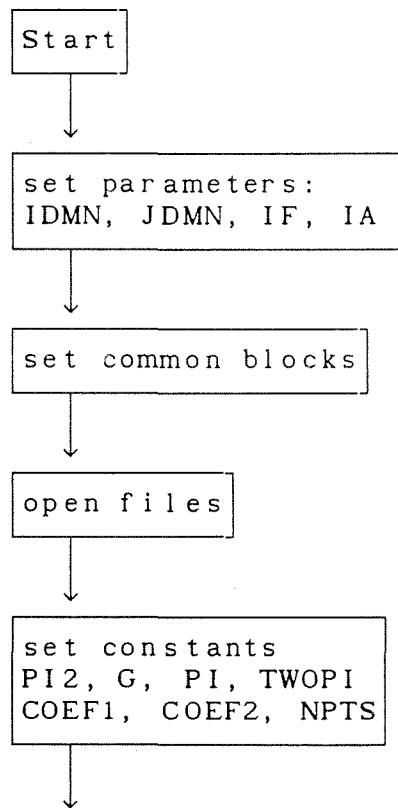


Figure 3. Flow chart for algorithm structure (continued next page).

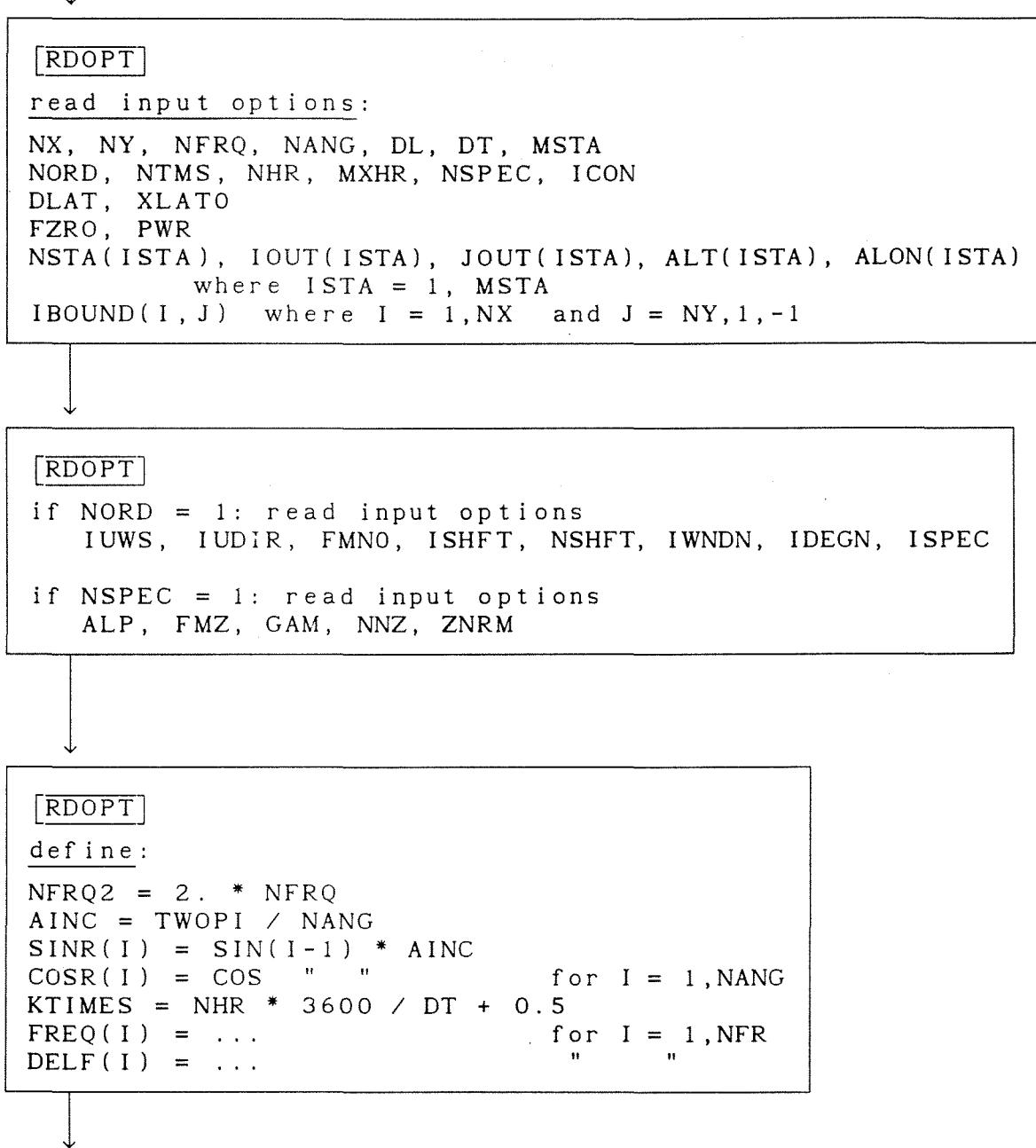


Figure 3. Flow chart for algorithm structure (continued next page).

↓

INITSNL

initialize SNL-related parameters

```

NANP, NANG2, NRNG
COSAN(I), SINAN(I); for I = 1,NANGP
ANGL2(I); for I = 1,NANG2

SCALF(I)
BOLSCL(I)
WKA(I)
OMA(I)
FRQA(I)
PHA(I) } for I = 1, NRNG

TKMIN, TKMAX

AK2(i,j,n)
WK2(i,j,n)
AK4(i,j,n)
WK4(i,j,n)
GRAD(i,j,n) } for i = 1, NPTS
                j = 1, NANGP
                n = 1, NRNG-1

```

LOCUS

```

XLOC2(n)
YLOC2(n)
XLOC4(n)
YLOC4(n)
DS(n) } for n = 1, NPTS

```

CPL

CSQ - the coupling coefficient

↓

INITSNL

define:

```

ADU(j,i,n)
AD(j,i,n)
XDU(j,i,n)
XD(j,i,n)
YDU(i,n)
YD(i,n) } for j = 1, NY
                    i = 1, NFRQ
                    n = NANG

```

Figure 3. Flow chart for algorithm structure (continued next page).

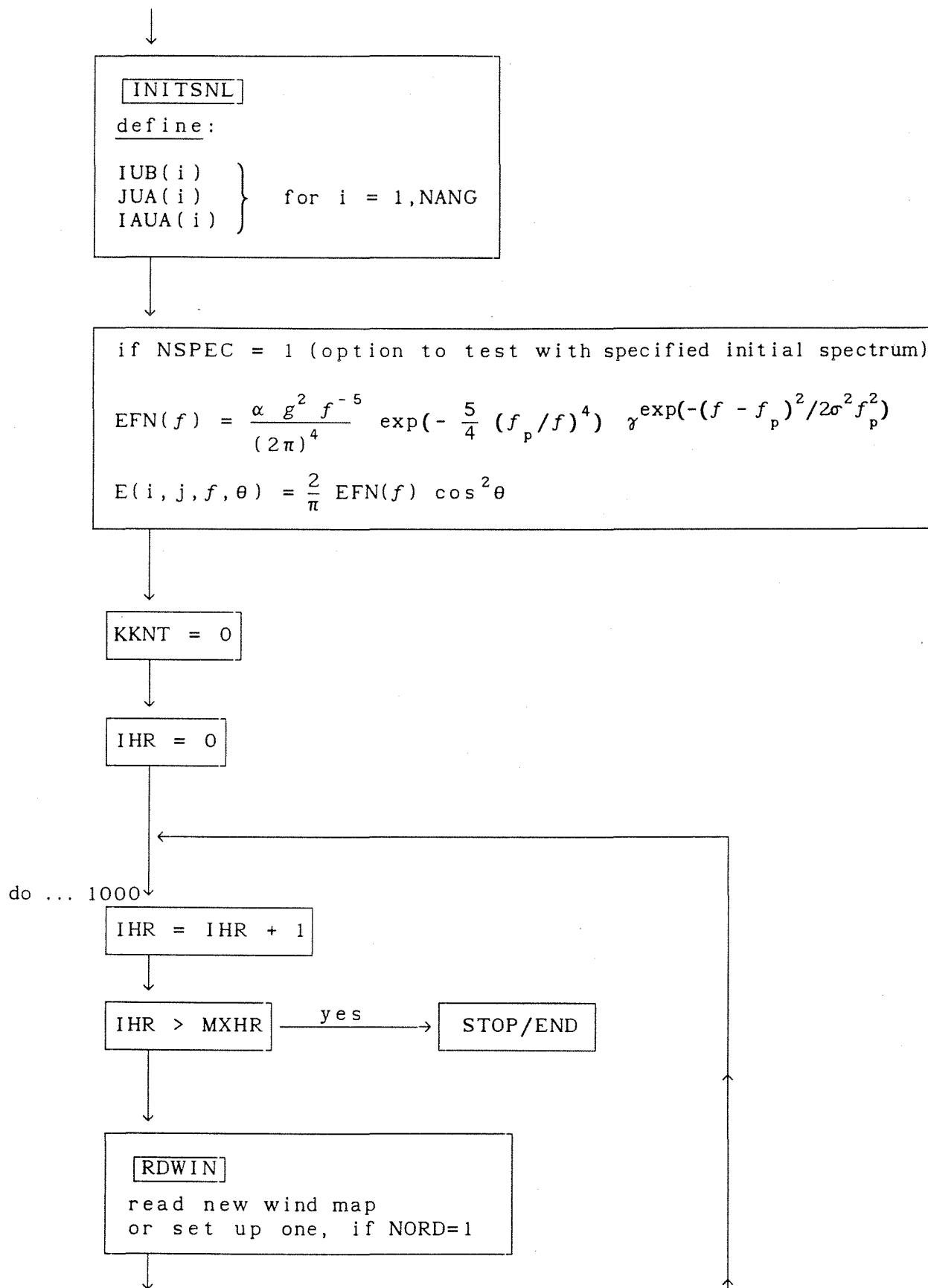


Figure 3. Flow chart for algorithm structure (continued next page).

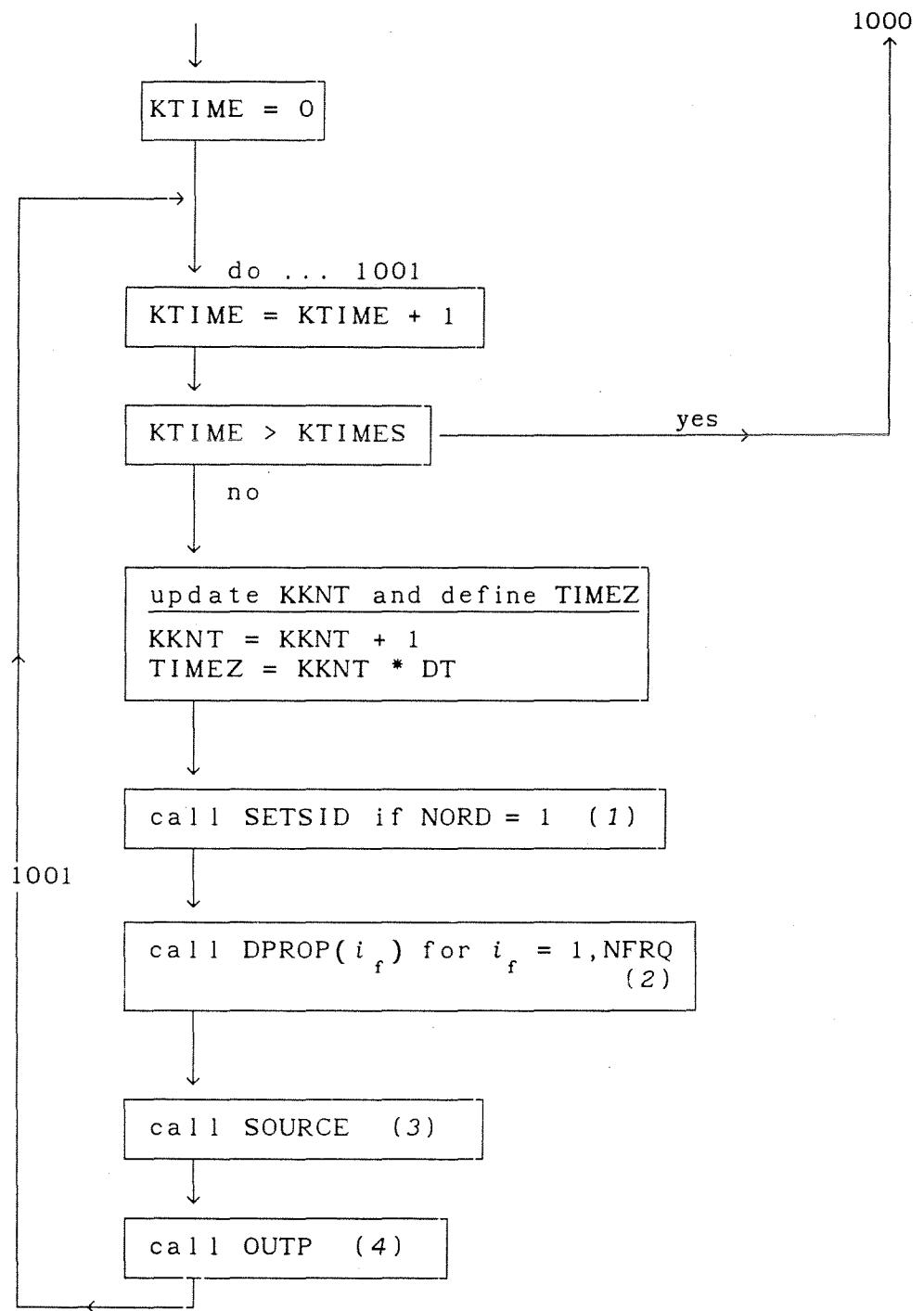


Figure 3. Flow chart for algorithm structure.

Notes:

- (1) sets the boundaries for special test case (NORD = 1), depending on type (ISPEC) specified in the input options: only EMA(NX,NY) and E(NX,NY,NFRQ,NANG) are affected.

(2) propagates energy over water points (if IBOUND(..) =1) using

IUB(..)	XD(..)	XDU(..)
JUA(..)	YD(..)	YDU(..)
IUAU(..)	AD(..)	ADU(..)

using scratch array EN(NX,NY,NANG) for storage and updating
E(NX,NY,NFRQ,NANG).

(3) estimates all source terms S_{nl} , S_{ds} , and S_{in} and performs one time step integration (see flow chart for SOURCE below for details)

(4) outputs results in the form of

- a) 1-D energy at selected output locations and times (tape 13).
- b) significant wave height H_s and spectral parameters at every time step (tape 11).
- c) 2-D energy at selected output locations and times (tape 10).
- d) maps of H_s , mean period T_p and mean direction $\bar{\theta}$ at selected locations and times (tape 15).

The Flow Chart for SOURCE

Because the structure of SOURCE is central to the program, we present its detailed structure. A break-down of the subroutines that fit inside it is also given.

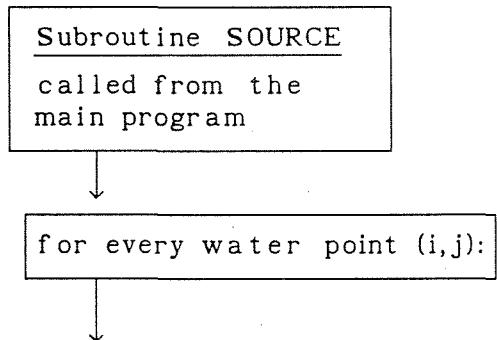


Figure 4. Flow chart for subroutine SOURCE (continued next page).

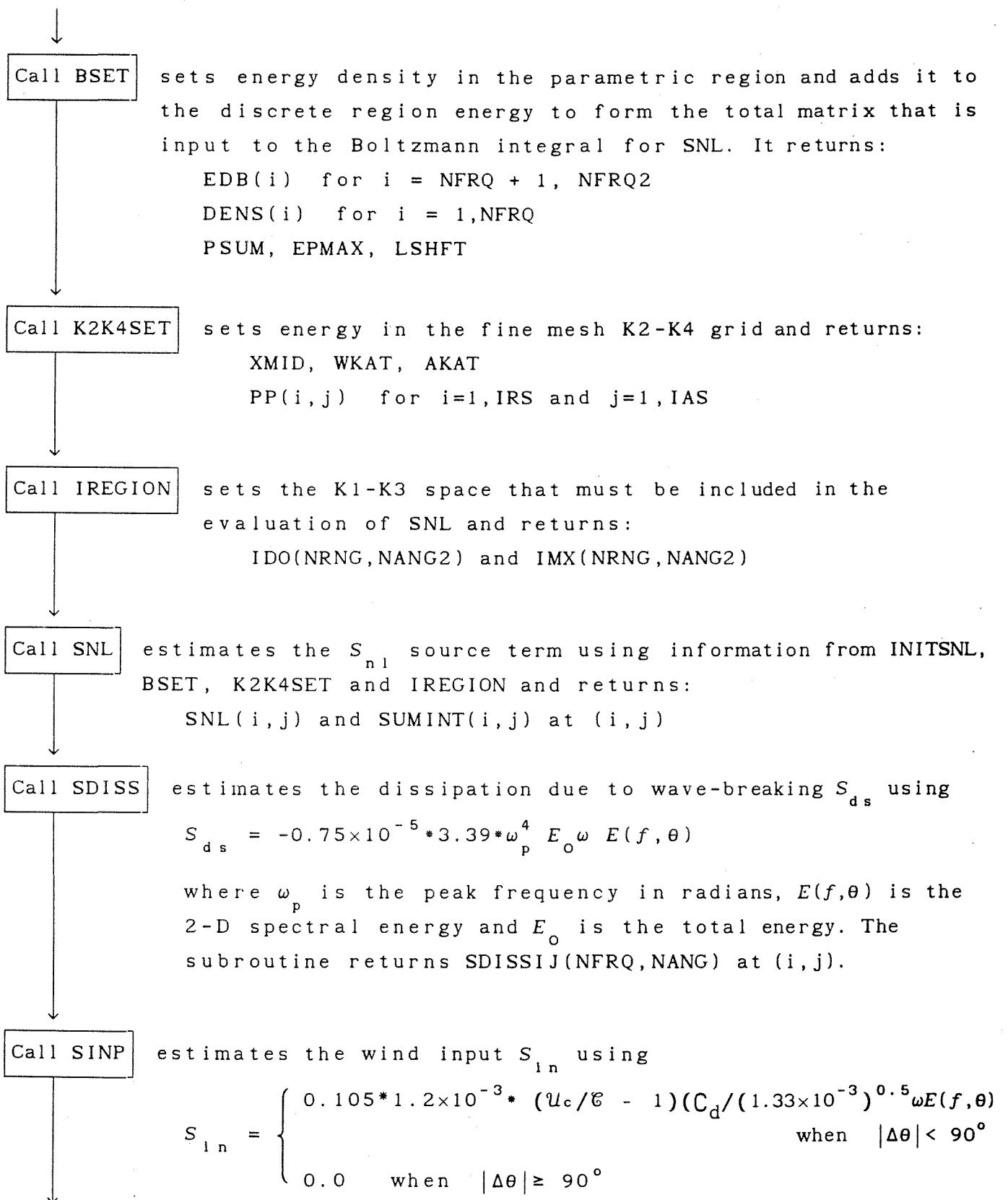


Figure 4. Flow chart for subroutine SOURCE (continued next page).

where U_c is the component of the wind in the direction of the waves, C is the phase velocity of the waves at frequency f and C_d is the drag coefficient. This subroutine returns SINIJ(NFRQ,NANG) at each (i,j) point.

integrate the wave balance equation in time
to update the 2-D energy density:

$$E(x,y,f,\theta) = E(x,y,f,\theta) + [S_{n1}^{1J}(f,\theta) + S_{ds}^{1J}(f,\theta) + S_{in}^{1J}(f,\theta)] * \Delta t$$

Calculate 1-D forms for energy $E(f)$,
and $S_{n1}(f)$, $S_{ds}(f)$ and $S_{in}(f)$

remove possible
negative energies

(option) force the
tail to have an f^{-4} form

return to main
program

Figure 4. Flow chart for subroutine SOURCE (continued next page).

4. List of Variables and Subroutines:

SUBROUTINES

The wave program has 15 subroutines which are, in sequential order,

1. RDOPT: reads all input options and defines the wave grid. It is called only once from the main program.
2. INITSLN: sets and initializes all SNL related variables, and is called once from the main program. It calls subroutines LOCUS and CPLE.
3. LOCUS: returns the locus for allowed nonlinear interactions, and is called once from subroutine INITSLN.
4. CPLE: returns the coupling coefficient for the nonlinear interactions and is called once from subroutine INITSLN.
5. RDWIN: reads new input wind map each 'NHR' hours, and is called once for each new input wind map if NORD=0, (i.e., 'MXHR' times) from the main program.
6. SETSID: sets boundaries for special test case (NORD=1), and is called once each new integration time step (i.e. 'MXHR*KTIMES' times) from the main program.
7. DPROP: propagates energy over water points, and is called 'NFRQ' times for each new integration time step (i.e. 'MXHR*KTIMES*NFRQ' times) from the main program.
8. SOURCE: calculates an estimate for each of the 3 source terms: SNL, SDISS AND SINP. It performs one integration time step and updates all wave variables. It is called once each new integration time step. (i.e. 'MXHR*KTIMES' times) from the main program. It also calls 6 other subroutines 'BSET, K2K4SET,IREGION, SNL, SDISS AND SINP'.

9. BSET: sets the energy densities in the parametric region and estimates the total energy in the parametric region. It is called once from subroutine SOURCE.
10. K2K4SET: sets the energies in the fine mesh K2-K4 grid, and is called once from subroutine SOURCE.
11. IREGION: sets the region of K1-K3 space that must be included in the evaluation of SNL, and is called once from subroutine SOURCE.
12. SNL: uses information from subroutines INITSNL, BSET, K2K4SET, and IREGION to estimate SNL, and is called once from subroutine SOURCE.
13. SDISS: estimates wave-breaking dissipation and is called once from subroutine SOURCE.
14. SINPUT: estimates the wind input, and is called once from subroutine SOURCE.
15. OUTP: outputs results such as 1-D & 2-D energy, Hs (significant wave height) maps, peak period and mean direction at selected output locations and times, and is called once at selected times from the main program.

VARIABLES*

*"(input)" indicates the variable must be read in initially

```

AA2      = AK2(IPT,NREF,NDIF)*SGN + ANG1
AA4      = AK4(IPT,NREF,NDIF)*SGN + ANG1
AD(NY,NFRQ,NANG) = 1 - ADU(NY,NFRQ,NANG), must be .LE. 1.0
ADU(NY,NFRQ,NANG) = the distances travelled along X-axis it cannot be
                     > 1.0 because the energy cannot travel more than one
                     grid in one time step.
AINC     = angle or (directional) increment in radians
           (=2*PI/NANG).

```

AINCQ = angle increment for SNL integration grid
 (= $2 * P120 / (IAS - 1)$)
 AKAT = 1 / AINCQ
 AK2(NPTS,NANGP,NRNG-1) = computed in INITSNL for SNL computations
 AK4(NPTS,NANGP,NRNG-1) = computed in INITSNL for SNL computations
 ALAT = latitude of a specific point (= [(J-1)*DLAT +
 XLATO] * RADC)
 ALON(MSTA) = longitudes of all special output points
 (labelled MSTA) (input).
 ALP = alpha value (=0.01) for optional test with
 specified initial spectrum for input. (input)
 ALP5 = $0.076 * (FMBAR / 3.5)^{**0.66667}$
 ALPCON = $ALP5 * G^*G / TWOPI^{**4}$
 ALPSTR = the 'UNIVERSAL' (not even dependent on
 anything) equilibrium constant (=0.0325)
 ALT(MSTA) = latitudes of all special output points
 (labelled by MSTA) (input)
 ANGO = the median direction between wind and wave
 directions defined as ATAN2(YY,XX) where
 XX = COS(WD(I,J))+COS(AVANG(I,J))
 YY = SIN(WD(I,J))+SIN(AVANG(I,J)) .
 ANGO defines the central band of the $4 * PI / 3$
 region 'LSHFT' used in the SNL computations
 and the smallest angle difference between the
 wind and the shifted wave direction 'ANGDIF'.
 ANG1 = (IANG-1)*AINC - P120
 ANG3 = (KANG-1)*AINC - P120
 ANGDIF = smallest angle difference between the wind
 direction at grid (I,J) and the shifted wave
 direction (= WD(I,J) - (IANG-LMID-1)*AINC)
 ANGLE = the angle in each direction (=(IANG-1)*AINC)
 ANGL2(NANG2) = angles array used in SNL integration
 (=(IANG-1)*AINC-P120: for IANG=1 to NANG2)
 ANGZ = angle position in SNL integration
 = (IIA-1)*AINCQ-P120) when IIA=1,IAS
 AVANG(NX,NY) = average wave direction in radians at all points

AVANGT

= average wave direction in degrees.

BETA1 = the ratio of energy-weighted average phase speed to the phase speed of the spectral peak (along the direction of propagation: = 0.73)
BOLSCL(NRNG) = scaling factor in the nonlinear SNL computation:
(=(PWR** (IRNG-1))**7.5 for IRNG=1 to NRNG)
BRCON = breaking constant (=0.75E-5) used in computing the dissipation 'SDISSIJ(NFRQ,NANG)'.

C37 = 3/7
C73 = 7/3
CC = wave phase velocity in m/s at a given frequency
CDFAC = the assumed drag coefficient for JONSWAP data used to convert wind speed U to wind stress USTAR (=1.2E-3)
CDRAG = drag coefficient at grid point (I,J)
(= (COEF1+COEF2*U10)*0.001)
CFAC = factor (=SQRT(CDRAG/1.33E-3) used in computing the wind input 'SINIJ(NFRQ,NANG)'
CG = group velocity (M/S)
CGG = group velocity (M/S)
COEF1 = parameter for the drag law at 10m height (=1.100)
COEF2 = parameter for the drag law at 10m height (=0.035)
CONJON = the JONSWAP constant for USTAR normalization represented by 'M1' in Resio and Perrie (1989)
(=1.7E-7/CDFAC: the JONSWAP drag coefficient)
CONTI = Q3*BETA1/(ALPSTR*XLAM)
COSAN(NANGP) = cosine array for angle bands from 1 to NANGP
= COS((IANG-1)*AINC) for IANG=1 to NANGP
COSR(NANG) = cosine array for angle bands from 1 to NANG
= COS((IANG-1)*AINC) for IANG=1 to NANG
CMP = phase velocity (m/s)(= G/(TWOPI*FMN))
CSQ = SNL coupling coefficient (computed in CPLE)

D1 = action density DENS(IRNG, IANG) at (IRNG, IANG)
 D2 = action density term PP(IW2, IA2)
 D3 = action density DENS(KRNG, KANG) at (KRNG, KANG)
 D4 = action density term PP(IW4, IA4)
 DE1 = EIJ1D(KMAX) - EIJ1D(KMAX-1)
 DE2 = EIJ1D(KMAX) - EIJ1D(KMAX+1)
 DELF(NFRQ) = frequency increments array (HZ)
 DELFF = frequency increment at each tail frequency
 normalized by the highest discrete frequency
 (= DELF(NFRQ) * SQRT(FRQA(IFRQ)/FREQ(NFRQ))
 for IFRQ = NFRQ+1 to NFRQ2)
 DELX = distance travelled by one wave train in one
 time step (units: m)
 = CGG * DT
 DENS(NFRQ2,NANG) = 2-D energy density spectrum
 = { EFFX*COS(ANGDIF)**2, if ABS(ANGDIF) < PI2
 0.0 otherwise
 DGM1 = sum of all (D4-D2)*GRAD(IPT,NREF,NDIF) over all
 locus points: IPT=1,NPTS
 DGM2 = sum of all D4*D2 *GRAD(IPT,NREF,NDIF) over all
 locus points: IPT=1,NPTS
 DK = wavenumber increment between 2 adjacent rings
 = WKO*(PWR**((IRNG+0.5)-PWR**((IRNG-0.5)))
 DL = distance between grid points in KM (input) then
 converted to meters
 DLAT = latitude increment of the grid (degrees: input)
 DS(NPTS) = incremental distance on the locus of
 interactions: used SNL computations in LOCUS
 DS13 = D3 - D1
 DSUM = total energy in the main frequency regime, not
 including energy in the tail
 DT = time step in seconds (=180) (input)
 DX13 = D1 * D3
 DZRO = the circumference of the earth (=40,000,000 m)

E(NX,NY,NFRQ,NANG)= 2-D wave energy spectrum at all grid points.

Initially it is parametric 2-D energy spectrum given by 'EE*COS(ANGLE)**NNZ'. Note that E has an f^{-4} tail (see RR2) ($=RR2(f)*E(I,J,f,TETA)$) if $f \geq 3.0 * FMA(I,J)$ where $f = FREQ(IFRQ)$, for IANG=1, NANG and TETA=ANGLE(IANG), for IFRQ=1, NFRQ. Directional energies are symmetric around the wind in the tail of the spectrum. direction.

EDB(NFRQ+1:NFRQ2) = parametric JONSWAP 1-D energy spectrum:

(= ALP*G*G/(TWOP1**4) * F5*GAM**EXP(-XX2) * EXP(-1.25*(FM/F)**(4)))

EE = normalized energy
(=EDB(IFRQ)*ZNRM in the high frequency tail)
(=EFN(IFRQ)*ZNRM in the main frequency bands)

EEE = 1-D energy density 'EIJ1D' at frequency FFF.
Used to force an f^{-4} spectral tail variation.

EFF = parametric 1-D energy density in the spectral tail.

If $FMA(I,J) \leq FREQ(NFREQ)$ then EFF is
EFF=ALPCON*F5, for FRQA(IFRQ)>3.5*FMA(I,J)
else
EFF=ALPSTR*(UST**2*CMP)**0.333*G*F4/TP3

If $FMA(I,J) > FREQ(NFREQ)$ then EFF =
FFCE*ALPSTR*(UST**2*CMP)**0.333*G*FQ4/TP3

EFFX = EFF*ZNRM*CG/(TWOP1*OMA(IFRQ)*WKA(IFRQ))

EFN(NFRQ) = parametric JONSWAP 1-D energy spectrum:
(= ALP*G*G/(TWOP1**4) * F5*GAM**EXP(-XX2) * EXP(-1.25*(FM/F)**(4)))

EIJ1D(NFRQ) = 1-D energy spectrum at grid point (I,J) after integrating all source terms (note that EIJ1D conforms to f^{-4} tail)
(= EEE*(FFF/f)**4, if $f \geq 3.0 * FMA(I,J)$: where $f=FREQ(IFRQ)$, for IFRQ=1, NFRQ)

EMAX = 2-D energy maximum (=DENS(NFRQ2,NANG2)-MAX)

EN(NX,NY,NANG) = storage for energy spreading into all directions
 at a fixed frequency at all grid points
 EOLD(NFRQ) = 1-D energy spectrum at grid point (I,J) before
 integrating all source terms
 EZZ(NANG) = COSINE to the power 'NNZ' type of energy spreading
 used with EDB(NFRQ+1,NFRQ2) in the high frequency
 tail regime
 EZED = total energy at grid point (I,J) derived from
 significant wave height (= HSIG(I,J)**2/16.08)

F4 = $F^{**}(-4) \equiv f^{-4}$
 F5 = $F^{**}(-5) \equiv f^{-5}$
 FFCE = EXP(1-ZZ2) , if FRQA(IFRQ) < FMN
 = 1.0 , otherwise
 FFF = closest discrete frequency in (HZ) from the
 'right side to 3 times the peak frequency at
 point (I,J) (FFF \geq 3.0 * FMA(I,J)). Used to
 force an f^{-4} spectral tail.
 FM = peak frequency (HZ)
 FMA(NX,NY) = peak frequency (HZ) at all grid points
 FMBAR = nondimensional peak frequency (=FMN*U/G)
 FMN = peak frequency at grid (I,J) (=1/TMN) or
 (=FMA(I,J) , if FMA(I,J) \leq FREQ(NFRQ)
 =FMPM , if FMA(I,J) > FREQ(NFRQ)
 and FMA(I,J) < FMPM
 = $1 / ((TM^{**}C73 + CONTT * (UST/G)^{**}1.333 * DT)^{**}C37)$
 if FMA(I,J) > FREQ(NFRQ)
 and FMA(I,J) \geq FMPM)
 FMNO = initial peak frequency (HZ) (input)
 FMPM = PIERSON MOSCOWITZ peak frequency
 (= ZCUT*G/(TWOP1*U10))
 FMZ = constant (=0.25) for optional test using specified
 initial spectrum as input. (input)

FP = estimated peak frequency, using information in
 the EIJ1D array (=FREQ(KMAX)
 (=0.5*FREQ(KMAX) +
 (0.5*((DE1/(DE1+DE2))*FREQ(KMAX+1) +
 (DE2/(DE1+DE2))*FREQ(KMAX-1))))
 FQ4 = MAX { FMN, FRQA(IFRQ) } **(-4)
 FREQ(NFRQ) = frequency array (HZ)
 FRQA(NRNG) = frequency array including the high frequency
 tail (=OMA(IRNG)/TWOP1 for IRNG=1 to NRNG)
 FZRO = lowest frequency in array (HZ) (=0.05) (input)
 F5 = F**(-5)

G = gravitational acceleration in m/s**2 (=9.8)
 GSQ = G**2
 GAM = gamma value (=3.3) for optional test with
 specified initial spectrum (input).
 GRAD(NPTS,NANGP,NRNG-1) = is computed in INITSL for nonlinear
 transfer (SNL) computations.

HS = significant wave height (= 4.0*SQRT(SUM))
 HSIG(NX,NY) = significant wave height at all grid points
 (= 4.0*SQRT(DSUM+PSUM))

IA = maximum number of angle bands (=24)
 IA2 = AA2 * AKAT + XMID must be in [1,IAS]
 IA4 = AA4 * AKAT + XMID must be in [1,IAS]
 IANGP = IANG + LSHFT - LMID: kept between 1 and NANG
 IAS = number of angle increments in 240° sector, used
 in PP-matrix (=121)

 IAUA(NANG) = coefficients used to determine upstream point
 locations

IBOUND(NX,NY) = flag for type of all grid points (input)
 = { 0 FOR LAND POINT
 1 FOR WATER POINT

ICON = flag for the type of the tail (input)
 = { 0 FOR UNCONSTRAINED TAIL
 4 FOR F**(-4) TAIL
 5 FOR F**(-5) TAIL

ID = wind date time code (=IHR for test case)

IDEGN = new wind direction after the wind has shifted
 (input)

IDMN = maximum number of grid columns (=21)

IDO(IRNG, IANG) = { 0, if WKA(IRNG)⁴ * DENS(IRNG, IANG)/SMAX < 0.005
 1, otherwise
 for IRNG=1, NRNG and IANG=1, NANG2

IENDWN = error check for ID: END OF WIND FILE FLAG

IF = maximum number of frequency bands (=20)

IIA1 = index of first angle position in SNL integration
 (=ANGZ/AINC + NANGP) must be between 1 and NANG2

IIA2 = index of second angle position in SNL integration
 (=IIA1+1) must be between 1 and NANG2

IIR1 = index of first ring position in SNL integration
 =(ALOG(XK)-ALOG(TKMIN))/ALOG(PWR) + 1.0001
 must be between 1 and NRNG

IIR2 = index of second ring position in SNL integration
 =(IIR1+1) must be between 1 and NRNG

IMX(IRNG, IANG) = { 1 , if at least one surrounding point has IDO=1
 0 , if all surrounding points have IDO=0
 for IRNG=1, NRNG and IANG=1, NANG2

IOUT(MSTA) = the I-coordinates of all special output points
 (input)

IPT = integer index for locus point position (=1, NPTS)

IRS = number of K-increments on the locus, used in
 PP-MATRIX (=501)

ISHFT = option for wind shift (input)
 = 0 for no wind shift
 = 1 for wind shift after NSHFT hours to new wind speed 'IWNDN' in m/s and new wind direction 'IDEGN' in degrees
 ISPEC = flag for boundary type option (input)
 = 0 for normal "solid" boundaries
 = 1 for top and bottom reflective boundaries
 = 2 for left and right reflective boundaries
 IUB(NANG) = coefficients used to determine upstream point locations
 IUDIR = wind direction for test case (input)
 IUWS = wind speed for test case (m/s) (input) assumed at 10 m height
 IW2 = WW2 * WKAT + 0.5 must be in [1,IRS]
 IW4 = WW4 * WKAT + 0.5 must be in [1,IRS]
 IWNDN = new wind speed (m/s) after the wind has shifted (input)

JDMN = maximum number of grid rows (=11)
 JOUT(MSTA) = the J-coordinates of all special output points (input)
 JUA(NANG) = coefficients used to determine upstream point locations

KKNT = number of integration time steps completed
 KMAX = discrete frequency index for the peak energy
 KTIMES = number of time steps between wind inputs
 (= NHR * 3600.0 / DT + 0.5)

LMID = NANG2 / 2
 LSHFT = defines the central band of 4*PI/3 region for SNL computations (=ANGO/AINC + 0.5)

MSTA	= number of special output locations (input) must be ≤ 70
MXHR	= maximum number of wind inputs (input)
<hr/>	
NANG	= number of directional angle bins (=24) (input) = number of radial lines in $[0^\circ, 120^\circ]$ in the grid
NANGP	= number of angles in SNL integration area $(= NANG * 0.333 + 1.5)$ must be $\leq NNA1$
NANG2	= $2 * NANGP - 1$ ($= NNA1$)
NDIF	= KRNG - IRNG
NFRQ	= number of frequency bands (=20) (input)
NFRQ2	= double the number of frequency bands ($= 2 * NFRQ$)
NHR	= number of hours between input winds (input)
NMID	= central angle index on SNL grid ($= IAS/2 + 1$)
NNA1	= maximum number of angles in 120° sector ($= 17$)
NNA2	= maximum number of angles in 240° $= NNA1 * 2 + 1 = 35$
NNR	= maximum number of rings (=50)
NNZ	= power (=2) for cosine spreading for optional test with specified initial spectrum (input)
NORD	= flag for the type of winds input (input) 0 for winds read in 1 for constant winds
NPA	= maximum number of points along the locus (=40), must be even number (input)
NPTS	= number of points used along the locus (input), must be $\leq NPA$ and an even number
NREF	= $IABS(KANG - IANG) + 1$
NRNG	= number of rings for grid ($= NFRQ2$), must be $\leq NNR$ (input)
NSPEC	= option to start from a fixed spectrum (input) 0 for starts with 0 densities 1 for starts with parametric densities 2 for starts with parametric densities with ALP forced to theoretical equilibrium value

NSHFT = number of time increments (IHR level) before
 wind shift to new direction and speed (input)
 NSTA(MSTA) = station numbers of all special output locations
 (input)
 NTMS = number of time steps between outputs (input)
 NX = number of columns in grid (input), must
 be \leq IDMN
 NY = number of rows in grid (input), must be \leq JDMN

OMA(NRNG) = angular frequency array (rad/s) including
 the high frequency tail regime
 ($=\text{SQRT}(G * \text{WKA}(\text{IRNG}))$ for IRNG=1 to NRNG)
 OMMX = peak angular frequency at grid point (I, J)
 ($=2 * \pi * \text{FMA}(I, J)$)

P120 = $2 * \pi / 3$
 PHA(NRNG) = $DK * \text{AINC} * \text{WKA}(\text{IRNG})$ for IRNG=1 to NRNG
 PI = $3.141592654 = \pi$
 PI2 = $\pi / 2$
 PICUT = PI2
 PP(IRS,IAS) = interpolated 2-D energy density DENS(NFRQ,NANG)
 onto (IRS, IAS) SNL grid. At SNL grid (IIR, IIA)
 where IIR=1, IRS and IIA=1, IAS it is given by:

$$= \begin{cases} X1 * \text{WTA1} + X2 * \text{WTA2} & \text{if } IIR1 \& IIR2 \in [1, NRNG] \\ 0.0 & \text{otherwise} \end{cases}$$

 and $X1 = \text{DENS}(IIR1, IIA1) * \text{WTK1} + \text{DENS}(IIR2, IIA1) * \text{WTK2}$
 $X2 = \text{DENS}(IIR1, IIA2) * \text{WTK1} + \text{DENS}(IIR2, IIA2) * \text{WTK2}$
 PSUM = total energy in the tail of the spectrum
 (= sum of all EFF*DELFF in the tail regime)
 PWR = radial power for the wavenumber/frequency grid
 ($=1.177225$) (input)

Q3 = $76.49 * \text{CONJON}$ where $76.49 =$
 $(7/10) * 3 * (2 * \pi) ** (10/3) / (4 * \pi)$

RORAT = 1.2E-3: used in computing the wind input
 'SINIJ(NFRQ,NANG)'
 RR1(F) = factor used for negative 2-D energies at
 each frequency. It multiplies the 2-D energies
 at all frequencies and directions if SUMZ3 >
 1.0E-5, otherwise the 2-D energies remain as
 reset to zero.
 = $\begin{cases} \text{MAX}\{0, (\text{SUMZ3} - \text{SUMNEG})/\text{SUMZ3}\}, & \text{if } \text{SUMZ3} > 1.0\text{E}-5 \\ 0.0 & \text{otherwise} \end{cases}$
 where F=FREQ(IFRQ), for IFRQ=1,NFRQ
 RR2(F) = factor used in operating on the tail of the 2-D
 energy spectrum at each frequency. It
 multiplies the energy at all frequencies and
 directions if the FREQ. > 3*FMA(I,J) other-
 wise the 2-D energies are unchanged.
 = $\begin{cases} ZZ1 * (\text{EEE} * (\text{FFF}/F)^4) / \text{EIJ1D}(F), & \text{if } F \geq 3 \times \text{FMA}(I,J) \\ 1.0 & \text{otherwise} \end{cases}$
 where F=FREQ(IFRQ), for IFRQ=1,NFRQ

SCALE = PWR** (IRNG-1) for IRNG=1 to NRNG
 SCALF(NRNG) = PWR** (IRNG-1) for IRNG=1 to NRNG
 SDISSIJ(NFRQ,NANG) = wave breaking dissipation at point (I,J)
 computed in SDIIS:
 (= -BRCON*OMA(IFRQ)*STP*E(I,J,IFRQ,IANG))
 SDS1D(NFRQ) = integral of dissipation source term SDISSIJ at
 point (I,J) over all directions "times" DT
 SGN = $\begin{cases} -1.0, & \text{if } \text{ANG3} < \text{ANG1} \\ 1.0, & \text{otherwise} \end{cases}$
 SIG = JONSWAP sigma parameter
 $\begin{cases} 0.07 & \text{if } F \leq FM \\ 0.09 & \text{if } F > FM \end{cases}$
 SINAN(NANGP) = SINE array for angle bands from 1 to NANGP
 (= SIN((IANG-1)*AINC) for IANG=1 to NANGP)

SINIJ(NFRQ,NANG) = wind input at grid point (I,J), computed as a function of frequency and direction:

$$= \begin{cases} 0.105 * RORAT * (U10 / CC * \cos(\text{ANGLE}) - 1.0) * \\ OMA(IFRQ) * E(I, J, IFRQ, IANG) * CFAC, & \text{if } \text{ABS(ANGDIF)} < \frac{\pi}{2} \\ 0.0 & \text{if } \text{ABS(ANGDIF)} \geq \frac{\pi}{2} \end{cases}$$

SINR(NANG) = SINE array for angle bands from 1 to NANG
 (= SIN((IANG-1)*AINC) for IANG=1 to NANG)
 SIN1D(NFRQ) = integral of wind input SINIJ at point (I,J)
 over all directions "times" DT
 SMAX = EMAX*WKA(IRNG)**4
 SNL1D(NFRQ) = integral of nonlinear interaction SNL at
 grid point (I,J) over directions "times" DT
 SNLIJ(NFRQ,NANG) = nonlinear wave-wave interaction at
 point (I,J) computed in SNL as a function of
 frequency and direction: (=SUMINT(IRNG,IANG)
 *WKA(IRNG)*OMA(IRNG)*TWOPI/CG
 at (IRNG,IANGP) in [NFRQ,NANG])
 STP = 3.39*EZED*OMMX**4
 SUM = SUM + EE*DELF(IFRQ)
 SUMIN = SUMIN + SINIJ(IFRQ,IANG) * DELF(IFRQ) for
 IFRQ=1, NFRQ and IANG=1, NANG
 SUMINT(IRNG,IANG) = SUMINT(IRNG,IANG) + TR31*PHA(KRNG)
 SUMINT(KRNG,KANG) = SUMINT(KRNG,KANG) - TR31*PHA(IRNG)
 SUMNEG = sum of negative 2-D energies at point (I,J)
 at each frequency
 SUMZ3 = sum over directions of the 2-D energy at
 point (I,J) at each frequency

```

T31      = action density term (=DX13*DGM1+DS13*DGM2)
TIMEZ    = time in seconds at each
           integration time step (=KKNT*DT)
TKMAX    = maximum wavenumber (=WKA(NRNG))
TKMIN    = minimum wavenumber (=WKA(1))
TMAIN    = peak period in seconds (=1/FMA(I,J))

```

TM = peak period at grid (I,J) (=1/FMA(I,J))
 TMN = peak period at grid (I,J)
 = $\begin{cases} 1/FMA(I,J) & \text{if } FMA(I,J) \leq FREQ(NFRQ) \\ 1/FMPM & \text{if } FMA(I,J) \geq FREQ(NFRQ) \& FMA(I,J) < FMPM \\ (TM^{**}C73+CONTT*(UST/G)**1.333*DT)**C37 & \text{if } FMA(I,J) > FREQ(NFRQ) \& FMA(I,J) \geq FMPM \end{cases}$
 TOUT(NX,NY) = peak period in seconds at all points
 TP3 = TWOPI**3 = $2\pi^3$
 TPEAK = peak wave period in seconds (=1/FREQ(KMAX))
 TR31 = SNL term = 2.0 * T31 * BOLSCL(IRNG)
 TSS(NX,NY) = peak wave period in seconds at all points
 TWOPI = 2*PI = 2π

U10 = wind speed at grid point (I,J) (=WS(I,J))
 USTSQ = (USTAR)**2 (=CDRAG*U10**2)
 UST = USTAR (=SQRT(USTSQ))

WD(NX,NY) = wind direction in radians at all points
 WKA(NRNG) = wavenumber array corresponding to FRQA(NRNG),
 including the high frequency tail regime
 (= WK0*PWR**((IRNG-1) for IRNG=1 to NRNG))
 WKAT = 1 / XINC
 WKZ = wavenumber corresponding to each frequency
 WKZO = lowest wavenumber corresponding to FZRO
 (=(TWOPI*FZRO)**2/G)
 WK0 = lowest wavenumber corresponding to the lowest
 frequency (=(TWOPI*FREQ(1))**2/G)
 = WKZO
 WK2(NPTS,NANGP,NRNG-1) = wavenumber term used in SNL and
 computed in INITSNL
 WK4(NPTS,NANGP,NRNG-1) = wavenumber term used in SNL and
 computed in INITSNL
 WS(NX,NY) = wind speed in m/s at all grid points
 WTA1 = 1.0 - WTA2
 WTA2 = (ANGZ-ANGL2(IIA1)) / AINC

WTK1	= (WKA(IIR2)-XK) / (WKA(IIR2)-WKA(IIR1))
WTK2	= 1.0 - WTK1
WW2	= WK2(IPT,NREF,NDIF) * SCALE
WW4	= WK4(IPT,NREF,NDIF) * SCALE

XD(NY,NFRQ,NANG) = 1.0 - XDU(NY,NFRQ,NANG), and must be ≤ 1
 XDU(NY,NFRQ,NANG) = the distances travelled along X-axis: cannot
 be > 1.0 because energy cannot travel more
 than one grid in one time step

XINC	= wavenumber (or ring) increment for SNL integration grid (=TKMAX/IRS)
XK	= wavenumber (or ring) position in SNL integration (for IIR=1, IRS: XK=IIR*XINC)
XLAM	= the factor which converts from FM parameters to total energy. It's the ratio of total energy to the integral of the equilibrium range from FM to ∞ ($=1.76$)
XLATO	= southernmost latitude in degrees (input), which is negative in southern hemisphere
XLOC2(NPTS)	= computed in LOCUS and used in SNL computation
XLOC4(NPTS)	= computed in LOCUS and used in SNL computation
XMID	= NMID + 0.5
XX2	= $(F-FM)^{**2} / (2.0*(SIG*FM)^{**2})$ IF (XX2.GT.2.5) XX2 = 2.5

YD(NFRQ,NANG)	= 1.0 - YDU(NFRQ,NANG), must be .LE. 1.0
YDU(NFRQ,NANG)	= the distances travelled along Y-axis, and it cannot be > 1.0 because energy cannot travel more than one grid in one time step
YLOC2(NPTS)	= computed in LOCUS and used in SNL computation
YLOC4(NPTS)	= computed in LOCUS and used in SNL computation
YY	= ABS(SINR(NANG))

ZCUT = controls the position of the peak frequency
 for fully developed Pierson-Moskowitz (Pierson
 and Moskowitz: (1964)) spectrum (= 0.90)
 ZNRM = normalizing factor for energy (=2/PI) (input)
 ZPAR = { 0.0 , for the tail frequencies
 1.0 , for the main discrete frequencies
 ZZ1 = factor used to force the tail to have an
 F**(-4) form: it is given by
 = { 3.0*FMA(I,J)/F, if F > 3.5*FMA(I,J)
 1.0 otherwise
 ZZ2 = (FMN/FRQA(IFRQ))**4
 if (ZZ2 .GT. 30.0) then ZZ2 = 30.0

5. The Program and its Functional Elements

We present in this section an account of the elements that make up the program as they are called and as they are used in the computation of sea surface wave spectra.

The program

The program begins with the usual setting of parameters, common blocks and open statements.

```
PROGRAM FULL

PARAMETER (IDMN=21, JDMN=11, IF=20, IA=24)
COMMON /A3/      G, PI, DL, NANG, NFRQ, NX, NY, TWOP1, NPTS, DT, PI2,
+                 FDISC, NFRQ2, NANG2, NANGP, NSPEC, FMZ, ALP, GAM, NNZ, ANRM
COMMON /CDR/     COEF1, COEF2, SPDFAC, XLAMZ, ALPSTR, BETA1, EPMAX, FPMAX
COMMON /CEN/     E(IDMN, JDMN, IF, IA)
COMMON /DPR/     IUB(IA), JUA(IA), IAUA(IA), AD(IF, JDMN, IA), ADU(IF, JDMN, IA)
COMMON /OPT/    MSTA, NSTR, NORD, JPRC, NTMS, NHR, IOUT(70), JOUT(70), KTIMES,
+                 MXHR, DLAT, XLATO, DZRO, NSTA(70), ALT(70), ALON7(70), PWR
COMMON /O1/      FREQ(IF), SINR(IA), COSR(IA), DELF(IF), AINC
COMMON /O4/      EN(IDMN, JDMN, IA)
COMMON /O5/      IBOUND(IDMN, JDMN)
COMMON /PMOD/   XD(IF, JDMN, IA), XDU(IF, JDMN, IA), YD(IF, IA), YDU(IF, IA)
COMMON /S2/      FMA(IDMN, JDMN), PSUM, DSUM, KKNT, TIMEZ
COMMON /S4/      HSIG(IDMN, JDMN)
COMMON /S5/      FMAN(IDMN, JDMN), E2(IF, IA), EF(IF), EFN(IF), FKA(IF)
COMMON /S7/      AVANG(IDMN, JDMN), TSS(IDMN, JDMN)
COMMON /TM/      TIMTOT
COMMON /WN/      WS(IDMN, JDMN), WD(IDMN, JDMN)

OPEN (20, FILE='options')
OPEN (21, FILE='winds')
OPEN (10, FILE='TWOD.DAT')
OPEN (11, FILE='ONEL.DAT')
OPEN (13, FILE='ONED.DAT')
OPEN (15, FILE='WAVES.DAT')
```

There follows the setting of constants and coefficients, for example:

```

G      = 9.8
PI     = 3.1415927
TWOPI  = 2.0 * PI
PI2    = PI / 2.0
DZRO   = 40.0E+06
COEF1  = 1.1
COEF2  = 0.035
NPTS   = 30
KKNT   = 0
IENDWN = 0

```

Thereafter subroutine RDOPT is called. This is the subroutine that reads the input options and configures the model run.

SUBROUTINE [RDOPT]

```

COMMON /A3/      G , PI , DL , NANG , NFRQ , NX , NY , TWOPI , NPTS , DT , PI2 ,
+                  FDISC , NFRQ2 , NANG2 , NANGP , NSPEC , FMZ , ALP , GAM , NNZ , ANRM
COMMON /OPT/     MSTA , NSTR , NORD , JPRC , NTMS , NHR , IOUT(70) , JOUT(70) , KTIMES ,
+                  MXHR , DLAT , XLATO , DZRO , NSTA(70) , ALT(70) , ALON(70) , PWR
COMMON /O1/      FREQ(IF) , SINR(IA) , COSR(IA) , DELF(IF) , AINC
COMMON /O5/      IBOUND(IDMN , JDMN)

```

Essential parameters are read in from 'options.dat' at this point:

```

READ(20,*)      NX , NY , NANG , NFRQ , DL , DT , MSTA
READ(20,*)      NORD , NTMS , NHR , MXHR , NSPEC , ICON
READ(20,*)      DLAT , XLATO
READ(20,*)      FZRO , PWR

```

and frequencies, wavenumbers and frequency increments are computed:

```

WKZ0 = (TWOPI*FZRO)**2 / G
DO 668 I=1 , NFRQ
      WKZ      = WKZ0 * PWR** (I-1)
      FREQ(I) = SQRT(G*WKZ) / TWOPI
668 CONTINUE
F1 = 0.5 * (FREQ(2)-FREQ(1))
DO 120 IFRQ=1 , NFRQ-1
      F2      = 0.5 * (FREQ(IFRQ+1)-FREQ(IFRQ))
      DELF(IFRQ) = F1 + F2
      F1      = F2
120 CONTINUE
DELF(NFRQ)      = 2.0 * F2
FDISC = FREQ(NFRQ) + 0.5*DELF(NFRQ)

```

```

IF (MSTA.GT.0) READ(20,*) (NSTA(ISTA),IOUT(ISTA),JOUT(ISTA),
+
ALT(ISTA),ALON(ISTA), ISTA=1,MSTA)
jwrt=(nx-1)/2
do 9190 ista=1,msta
fetch(ista)=dl*ista+dl
9190 continue

```

the number of time steps computed,

```

SECBI=NHR*3600.
KTIMES=SECBI/DT+0.5

```

and arrays defined for sine and cosine functions,

```

DO 180 IANG=1,NANG
ANGLE=(IANG-1)*AINC
SINR(IANG)=SIN(ANGLE)
COSR(IANG)=COS(ANGLE)
180 CONTINUE

```

Finally, the boundary geometry of the ocean is read in. Print statements in the program write out the grid that can be specified here.

```

Grid Orientation is:
1,1 in lower corner
1,NY is upper left corner
To read Top-Down loop over J=NY,1,-1.

DO 295 J=NY,1,-1
READ (20,959) (IBOUND(I,J),I=1,NX)
295      continue

```

The major subroutine that is then called is INITSLN. This initializes coefficients for the nonlinear transfer computation S_{nl} . The latter is a limited region version of the full Boltzmann model. It is valid for arbitrary spectra. The parameter specifications NNR=50, NNA1=17, NPA=40, NNA2=NNA1*2+1 are upper bounds for the parameters for the number of frequency bands, angle bands that can be read in.

SUBROUTINE **INITSNL**

```

PARAMETER (IDMN=21, JDMN=11, IF=20, IA=24)
PARAMETER (NNR=50, NNA1=17, NPA=40, NNA2=NNA1*2+1, IRS=501, IAS=121)
COMMON /A/      XLOC2(NPA), YLOC2(NPA), XLOC4(NPA), YLOC4(NPA), DS(NPA)
COMMON /A3/     G, PI, DL, NANG, NFRQ, NX, NY, TWOPI, NPTS, DT, PI2,
+                FDISC, NFRQ2, NANG2, NANGP, NSPEC, FMZ, ALP, GAM, NNZ, ANRM
COMMON /B/     WKA(NNR), OMA(NNR), FRQA(NNR),
+                COSAN(NNA1), SINAN(NNA1), EDENS(NNR), EDENS2(NNR)
COMMON /C/     SUMINT(NNR, NNA2), PHA(NNR), SCALF(NNR), BOLSCL(NNR),
+                DENS(NNR, NNA2), EDB(NNR)
COMMON /D/     IDO(NNR, NNA2), IMX(NNR, NNA2), TKMAX, TKMIN
COMMON /EFG/    PP(IRS, IAS), ANGL2(NNA2), AKAT, WKAT, XMID,
+                GRAD(NPA, NNA1, NNR), WK2(NPA, NNA1, NNR), AK2(NPA, NNA1, NNR),
+                WK4(NPA, NNA1, NNR), AK4(NPA, NNA1, NNR)
COMMON /OPT/   MSTA, NSTR, NORD, JPRC, NTMS, NHR, IOUT(70), JOUT(70), KTIMES,
+                MXHR, DLAT, XLATO, DZRO, NSTA(70), ALT(70), ALON(70), PWR
COMMON /01/    FREQ(IF), SINR(IA), COSR(IA), DELF(IF), AINC
COMMON /04/    EN(IDMN, JDMN, IA)
COMMON /05/    IBOUND(IDMN, JDMN)

```

Our limited-region computation is the angle wedge:

C NANGP IS THE NUMBER OF ANGLE BANDS IN THE SNL INTEGRATION AREA
 $NANGP = NANG * 0.333 + 1.5$

We first compute the angular-radial grid, as discussed in Tracy and Resio (1982). The radial variation is of the form $k_o \lambda^i$, where k_o is the lowest wavenumber and λ the factor relating to higher wavenumbers:

```

WKO      = (TWOPI*FREQ(1))**2/G
DO 1 IRNG=1, NRNG
  SCALE      = PWR**(IRNG-1)
  SCALF(IRNG) = SCALE
  BOLSCL(IRNG) = SCALE**7.5
  WKA(IRNG)   = WKO * SCALE
  OMA(IRNG)   = SQRT(G*WKA(IRNG))
  FRQA(IRNG)  = OMA(IRNG) / TWOPI
  DK          = WKO * (PWR**(IRNG+0.5) - PWR**(IRNG-0.5))
  PHA(IRNG)   = DK * AINC * WKA(IRNG)
1 CONTINUE
  TKMIN = WKA(1)
  TKMAX = WKA(NRNG)

```

```

DO 2 IANG=1,NANGP
    ANGLE      = (IANG-1)*AINC
    COSAN(IANG) = COS(ANGLE)
    SINAN(IANG) = SIN(ANGLE)
2 CONTINUE

```

Then we compute wavenumbers k_1 and k_3 and their components IRNG and IANG are K1 parameters. KRNG and KANG are K3 parameters. components are X1, Y1, X3 and Y3.

```

IRNG=1
NST=IRNG+1
X1=WKA(1)
Y1=0.0
DO 41 KRNG=NST, NRNG
    NDIF=KRNG-IRNG
    DO 4 KANG=1, NANGP
        X3=WKA(KRNG)*COSAN(KANG)
        Y3=WKA(KRNG)*SINAN(KANG)

```

Then we call LOCUS, which computes the locus of the other 2 wavenumbers k_2 and k_4 , necessary to make up the resonant 4-wave interaction. LOCUS is described in Tracy and Resio (1982), and is not given here. LOCUS also computes the line increment element DS(ip), which is necessary to do the line-integration around the locus-of-interaction.

```
call LOCUS(X1, Y1, X3, Y3, NPTS)
```

Having the components of k_2 and k_4 , which are X2, Y2, X4 and Y4, we need the coupling coefficient for the wave-wave interactions. We call CPLE which returns CSQ, which we do not present as it has been documented in Tracy and Resio (1982),

```
CALL CPLE(X1, Y1, X2, Y2, X3, Y3, X4, Y4, CSQ)
```

We compute the factors that make up the SNL integral:

```

DO 5 IPT=1,NPTS
XK2SQ=X2*X2+Y2*Y2
WK2(IPT,KANG,NDIF)=SQRT(XK2SQ)
AK2(IPT,KANG,NDIF)=ATAN2(Y2,X2)
XK4SQ=X4*X4+Y4*Y4
WK4(IPT,KANG,NDIF)=SQRT(XK4SQ)
AK4(IPT,KANG,NDIF)=ATAN2(Y4,X4)
Z2=XK2SQ**(-0.75)
Z4=XK4SQ**(-0.75)
ZZX=X2*Z2-X4*Z4
ZZY=Y2*Z2-Y4*Z4

```

and present the increment to nonlinear transfer (in Resio and Perrie (1991) this is $\epsilon^2 \left| \frac{\partial W}{\partial n} \right| ds$; equations (2.3)-(2.4)).

```

C      GRAD HERE IS 1/DWDN * DS * COUPLING COEFFICIENT
ZZSUM=ZZX*ZZX+ZZY*ZZY
ZZSUM=ZZSUM*G
DIF14=(X1-X4)**2 + (Y1-Y4)**2
DIF13=(X1-X3)**2 + (Y1-Y3)**2
THFNC=1.0
IF (DIF13.GE.DIF14) THFNC=0.
GRAD(IPT,KANG,NDIF)=2.0*DS(IPT)*CSQ /SQRT(ZZSUM)*THFNC
5      CONTINUE
4      CONTINUE
41     CONTINUE

```

Parameter NORD is the flag to determine if we read real winds or use hypothetical constant SWAMP-type winds:

```

IF (NORD.EQ.0) GO TO 1347
READ(20,*) IUWS,IUDIR,FMNO,ISHFT,NSHFT,IWNDN,IDEGN,ISPEC
DO 1360 I=1,NX
DO 1360 J=1,NY
WS(I,J) = IUWS * 1.0
WD(I,J) = IUDIR * TWOPI/360.0
FMA(I,J) = FMNO
1360    CONTINUE
1347    ID=1

```

We compute coefficients for deep water propagation. These are used later in the propagation subroutine DPRP. XDU and YDU are X- and Y- distances and thus must be > 1 , because energy cannot go more than 1 grid spacing in 1 time step

```

C DZRO IS THE CICUMFERENCE OF THE EARTH IN METRES
C CGG = GROUP VELOCITY OF WAVE TRAIN, (M/S)
C DS = DISTANCE TRAVELED BY ONE WAVE TRAIN
C IN ONE TIME STEP, (M)
TINC = DT
DO 778 IFRQ=1,NFRQ
  CGG = 0.5 * G / (TWOPI*FREQ(IFRQ))
  DS = CGG * TINC
C IUB AND JUA ARE COEFFICIENTS USED TO DETERMINE UPSTREAM POINT LOCATIONS
DO 777 J=2,NY-1
  DO 777 IIA=1,NANG
    XX = ABS(COSR(IIA))
    YY = ABS(SINR(IIA))
    ALAT = ((J-1)*DLAT + XLATO) * (TWOPI/360.0)
    XNPTS = COS(ALAT)
    ADU(IFRQ,J,IIA) = DS*TWOPI/(DZRO*AINC+1.0E-10) * SIN(ALAT)*XX
    IF (ADU(IFRQ,J,IIA).GT.1.0) ADU(IFRQ,J,IIA) = 1.0
    XDU(IFRQ,J,IIA) = XX*DS/(DL*XNPTS)
    IF (XDU(IFRQ,J,IIA).GT.1.0) XDU(IFRQ,J,IIA) = 1.0
    YDU(IFRQ,IIA) = YY*DS/DL
    IF (YDU(IFRQ,IIA).GT.1.0) YDU(IFRQ,IIA) = 1.0
    XD(IFRQ,J,IIA) = 1.0 - XDU(IFRQ,J,IIA)
    YD(IFRQ,IIA) = 1.0 - YDU(IFRQ,IIA)
    AD(IFRQ,J,IIA) = 1.0 - ADU(IFRQ,J,IIA)
777 CONTINUE
778 CONTINUE
  DO 776 IIA=1,NANG
    X = COSR(IIA)
    IF (ABS(X).LT.0.001) GOTO 7761
    IUB(IIA) = (ABS(X)+0.001) / X
    IAUA(IIA) = IIA + IUB(IIA)
    IF (IAUA(IIA).GT.NANG) IAUA(IIA) = 1
    GOTO 7762
7761 IUB(IIA) = 0
    IAUA(IIA) = IIA
7762 Y = SINR(IIA)
    IF (ABS(Y).LT.0.001) GOTO 7763
    JUA(IIA) = (ABS(Y)+0.001) / Y
    GOTO 776
7763 JUA(IIA)=0
776 CONTINUE

```

To complete the initialization, if winds are read (NORD=0), we set the following parameters:

```

DO 250 I=1,NX
DO 250 J=1,NY
  IF (NORD.EQ.0) FMA(I,J) = 1.0
  HSIG(I,J) = 0.0
  AVANG(I,J) = 0.0
  DO 255 L=1,NANG

```

```

EN(I,J,L) = 0.0
DO 254 K=1,NFRQ
    E(I,J,K,L) = 0.0
254     CONTINUE
255     CONTINUE
250 CONTINUE
NNZ = 2
ZNRM = 2.0 / PI
ANRM = ZNRM
GAM = 3.3

```

Concomitantly, we start from a parameteric spectrum (if NSPEC > 0), with JONSWAP form and COSINE-type spreading (Hasselmann et al: 1973). Significant wave height HS is computed, even here,

```

READ(20,*) ALP,FMZ,GAM,NNZ,ZNRM
FM = FMZ
ANRM = ZNRM
DO 290 I=1,NX
    DO 280 J=1,NY
        IF (IBOUND(I,J).NE.1) GOTO 280
        SUM = 0.0
        FMA(I,J) = FM
        DO 281 K=1,NFRQ
            IF (FREQ(K)/FM.LT.0.4) GOTO 281
            F5 = FREQ(K)**(-5)
            XX = (FREQ(K)-FM)**2
            SIG = 0.07
            IF (FREQ(K).GT.FM) SIG = 0.09
            XX2 = XX/(2.* (SIG*FM)**2)
            IF (XX2.GT.2.5) XX2 = 2.5
            EE=ALP*G*G/(TWOPI**4)*F5*GAM**EXP(-XX2)*
+                               EXP(-1.25*(FM/FREQ(K))**(4))
            EFN(K) = EE
            SUM = SUM + EE*DELF(K)
            EE = EE*ZNRM
        DO 282 L=1,NANG
            ANG = (L-1) * AINC
            XX = COS(ANG)
            E(I,J,K,L) = EE * XX**NNZ
            IF (XX.LE.1.0E-2) E(I,J,K,L) = 0.0
282     CONTINUE
283     FORMAT(8E9.2)
281     CONTINUE
        HS = 4.0*SQRT(SUM)
280     CONTINUE
290 CONTINUE

```

Begin the time loop. If NORD=1, use constant winds,

```

DO 1000 IHR=1,MXHR
IF (NORD.EQ.1) GO TO 1100

```

otherwise, read in wind maps by calling subroutine RDWIN(ID,IENDWN). Reads in sets of wind input, including wind speed and wind direction. Wind speeds are in units of m/sec and are assumed to be at a 10-meter height. Wind directions are vector degrees.

```

SUBROUTINE RDWIN(ID, IENDWN)

C      IDMN      = MAXIMUM NUMBER OF GRID COLUMNS
C      JDMN      = MAXIMUM NUMBER OF GRID ROWS
C      IF         = MAXIMUM NUMBER OF FREQUENCY BANDS
C      IA         = MAXIMUM NUMBER OF ANGLE BANDS
C      ID         = DATE TIME CODE
C      IENDWN    = ERROR CHECK PARAMETER FOR ID
C      ID         = DATE TIME CODE
C      WSN(I,J)  = WIND SPEED IN M/SEC AT POINT I,J
C      WDN(I,J)  = WIND DIRECTION VECTOR DEGREES AT POINT I,J
COMMON /A3/      G, PI, DL, NANG, NFRQ, NX, NY, TWOPI, NPTS, DT, PI2,
+                  FDISC, NFRQ2, NANG2, NANGP, NSPEC, FMZ, ALP, GAM, NNZ, ANRM
COMMON /OPT/     MSTA, NSTR, NORD, JPRC, NTMS, NHR, IOUT(70), JOUT(70), KTIMES,
+                  MXHR, DLAT, XLATO, DZRO, NSTA(70), ALT(70), ALON(70), PWR
COMMON /WN/       WS (IDMN, JDMN), WD (IDMN, JDMN)
READ(21,210,END=98) ID
210 FORMAT(I10)
DO 120 J=NY,1,-1
  READ(21,*) (WS(I,J), I=1,NX)
120 CONTINUE
DO 121 J=NY,1,-1
  READ(21,*) (WD(I,J), I=1,NX)
121 CONTINUE

```

The loop for the number of time steps between wind inputs is now begun,

```

DO 1001 KTIME=1,KTIMES
KKNT=KKNT+1

```

It is necessary at this point to set boundaries for special SWAMP-type winds (if NORD=1), and to therefore call SETSID, setting reflection conditions (specified by ISPEC). This affects FMA(NX,NY) and E(NX,NY,NFRQ,NANG).

```

SUBROUTINE SETSID(NX,NY,NF,NA,ISPEC)
COMMON /CEN/   E (IDMN, JDMN, IF, IA)

```

```

COMMON /05/    IBOUND(IDMN,JDMN)
COMMON /S2/    FMA(IDMN,JDMN),PSUM,DSUM,KKNT,TIMEZ
IF (ISPEC.EQ.0) RETURN
IF (ISPEC.NE.1) GOTO 10
DO 1 I=2,NX-1
  IF (IBOUND(I,J).NE.1) GOTO 1
  FMA(I,1) = FMA(I,2)
  FMA(I,NY) = FMA(I,NY-1)
  DO 2 IAA=1,NA
  DO 2 ITF=1,NF
    E(I,1,ITF,IAA) = E(I,2,ITF,IAA)
    E(I,NY,ITF,IAA) = E(I,NY-1,ITF,IAA)
2      CONTINUE
1      CONTINUE
      RETURN
10     CONTINUE
      IF (ISPEC.NE.2) GOTO 20
      DO 3 J=2,NY-1
        IF (IBOUND(I,J).NE.1) GOTO 3
        FMA(1,J) = FMA(2,J)
        FMA(NX,J) = FMA(NX-1,J)
        DO 4 IAA=1,NA
        DO 4 ITF=1,NF
          E(1,J,ITF,IAA) = E(2,J,ITF,IAA)
          E(NX,J,ITF,IAA) = E(NX-1,J,ITF,IAA)
4      CONTINUE
3      CONTINUE
      RETURN
20     CONTINUE
      DO 5 I=2,NX-1
        IF (IBOUND(I,J).NE.1) GOTO 5
        FMA(I,1) = FMA(I,2)
        FMA(I,NY) = FMA(I,NY-1)
        DO 6 IAA=1,NA
        DO 6 ITF=1,NF
          E(I,1,ITF,IAA) = E(I,2,ITF,IAA)
          E(I,NY,ITF,IAA) = E(I,NY-1,ITF,IAA)
6      CONTINUE
5      CONTINUE
      DO 7 J=2,NY-1
        IF (IBOUND(I,J).NE.1) GOTO 7
        FMA(1,J) = FMA(2,J)
        FMA(NX,J) = FMA(NX-1,J)
        DO 8 IAA=1,NA
        DO 8 ITF=1,NF
          E(1,J,ITF,IAA) = E(2,J,ITF,IAA)
          E(NX,J,ITF,IAA) = E(NX-1,J,ITF,IAA)
8      CONTINUE
7      CONTINUE

```

Propagation of energy now occurs by calling subroutine DPRPROP, one frequency at a time. This is only *first* order propagation, with a curvature estimate:

```

SUBROUTINE DPROP(K)
COMMON /A3/   G,PI,DL,NANG,NFRC,NX,NY,TWOP1,NPTS,DT,PI2,
+             FDISC,NFRQ2,NANG2,NANGP,NSPEC,FMZ,ALP,GAM,NNZ,ANRM
COMMON /CEN/   E(IDMN,JDMN,IF,IA)
COMMON /DPR/   IUB(IA),JUA(IA),IAUA(IA),AD(IF,JDMN,IA),ADU(IF,JDMN,IA)
COMMON /OS/    IBOUND(IDMN,JDMN)
COMMON /O4/    EN(IDMN,JDMN,IA)
COMMON /PMOD/  XD(IF,JDMN,IA),XDU(IF,JDMN,IA),YD(IF,IA),YDU(IF,IA)

NXM1 = NX - 1
NYM1 = NY - 1
C LOOPS FOR THE FIRST ORDER PROPAGATION
DO 100 I=2,NXM1
DO 100 J=2,NYM1
C ONLY PROPAGATE WATER POINTS
IF (IBOUND(I,J).NE.1) GOTO 100
DO 101 IIA=1,NANG
IU           = I - IUB(IIA)
EN(I,J,IIA) = XD(K,J,IIA)*E(I,J,K,IIA)+XDU(K,J,IIA)*E(IU,J,K,IIA)
101 CONTINUE
100 CONTINUE
DO 200 I=2,NXM1
DO 200 J=2,NYM1
IF (IBOUND(I,J).NE.1) GOTO 200
DO 201 IIA=1,NANG
JU           = J - JUA(IIA)
E(I,J,K,IIA) = YD(K,IIA)*EN(I,J,IIA)+YDU(K,IIA)*EN(I,JU,IIA)
201 CONTINUE
200 CONTINUE
C BEGIN CURVATURE ESTIMATION IN PROPAGATION
DO 300 I=2,NXM1
DO 300 J=2,NYM1
IF (IBOUND(I,J).NE.1) GOTO 300
DO 301 IIA=1,NANG
EN(I,J,IIA) = E(I,J,K,IIA)
301 CONTINUE
300 CONTINUE
DO 400 I=2,NXM1
DO 400 J=2,NYM1
IF (IBOUND(I,J).NE.1) GOTO 400
DO 401 IIA=1,NANG
IAU          = IAUA(IIA)
E(I,J,K,IIA) = AD(K,J,IIA)*EN(I,J,IIA)+ADU(K,J,IIA)*EN(I,J,IAU)
401 CONTINUE
400 CONTINUE

```

We call the SOURCE subroutine to update S_{nl} , S_{ds} and S_{in} at all points, frequencies and angles. This is the central subroutine of the program. Its structure is to loop on i-j spatial coordinates. All information from subroutines called here is returned in single-point (f, θ) matrices in common blocks /SN/, /SD/ AND /SI/.

```

SUBROUTINE SOURCE
COMMON /A3/    G,PI,DL,NANG,NFRQ,NX,NY,TWOP1,NPTS,DT,PI2,
+                FDISC,NFRQ2,NANG2,NANGP,NSPEC,FMZ,ALP,GAM,NNZ,ANRM
COMMON /B/      WKA(NNR),OMA(NNR),FRQA(NNR),
+                COSAN(NNA1),SINAN(NNA1),EDENS(NNR),EDENS2(NNR)
COMMON /C/      SUMINT(NNR,NNA2),PHA(NNR),SCALF(NNR),BOLSCL(NNR),
+                DENS(NNR,NNA2),EDB(NNR)
COMMON /CDR/    COEF1,COEF2,SPDFAC,XLAMZ,ALPSTR,BETA1,EPMAX,FPMAX
COMMON /CEN/    E(IDMN,JDMN,IF,IA)
COMMON /DPR/    IUB(IA),JUA(IA),IAUA(IA),AD(IF,JDMN,IA),ADU(IF,JDMN,IA)
COMMON /OPT/    MSTA,NSTR,NORD,JPRC,NTMS,NHR,IOUT(70),JOUT(70),KTIMES,
+                MXHR,DLAT,XLATO,DZRO,NSTA(70),ALT(70),ALON(70),PWR
COMMON /01/     FREQ(IF),SINR(IA),COSR(IA),DELF(IF),AINC
COMMON /04/     EN(IDMN,JDMN,IA)
COMMON /05/     IBOUND(IDMN,JDMN)
COMMON /PMOD/   XD(IF,JDMN,IA),XDU(IF,JDMN,IA),YD(IF,IA),YDU(IF,IA)
COMMON /SD/     SDIJJ(IF,IA)
COMMON /SI/     SINIJ(IF,IA)
COMMON /SN/     SNLIJ(IF,IA),LSHFT
COMMON /S2/     FMA(IDMN,JDMN),PSUM,DSUM,KKNT,TIMEZ
COMMON /S4/     HSIG(IDMN,JDMN)
COMMON /S5/     FMAN(IDMN,JDMN),E2(IF,IA),EF(IF),EFN(IF),FKA(IF)
COMMON /S7/     AVANG(IDMN,JDMN),TSS(IDMN,JDMN)
COMMON /TM/     TIMTOT
COMMON /WN/     WS(IDMN,JDMN),WD(IDMN,JDMN)
DIMENSION EIJ1D(IF),EOLD(IF),SNL1D(IF),SIN1D(IF),SDS1D(IF)

```

The first thing to do is set the energy densities in the parametric region and estimate the total energy in the parametric region. We call BSET for each point : I=1,NX and J=1,NY. BSET takes energy from the parametric region, adds it to the discrete region to form a complete energy matrix which is needed to input to the Boltzmann integral of SNL.

```

SUBROUTINE BSET(I,J)
COMMON /A3/    G,PI,DL,NANG,NFRQ,NX,NY,TWOP1,NPTS,DT,PI2,
+                FDISC,NFRQ2,NANG2,NANGP,NSPEC,FMZ,ALP,GAM,NNZ,ANRM
COMMON /B/      WKA(NNR),OMA(NNR),FRQA(NNR),
+                COSAN(NNA1),SINAN(NNA1),EDENS(NNR),EDENS2(NNR)
COMMON /C/      SUMINT(NNR,NNA2),PHA(NNR),SCALF(NNR),BOLSCL(NNR),
+                DENS(NNR,NNA2),EDB(NNR)
COMMON /CDR/    COEF1,COEF2,SPDFAC,XLAMZ,ALPSTR,BETA1,EPMAX,FPMAX
COMMON /CEN/    E(IDMN,JDMN,IF,IA)
COMMON /OPT/    MSTA,NSTR,NORD,JPRC,NTMS,NHR,IOUT(70),JOUT(70),KTIMES,
+                MXHR,DLAT,XLATO,DZRO,NSTA(70),ALT(70),ALON(70),PWR
COMMON /01/     FREQ(IF),SINR(IA),COSR(IA),DELF(IF),AINC
COMMON /SN/     SNLIJ(IF,IA),LSHFT
COMMON /S2/     FMA(IDMN,JDMN),PSUM,DSUM,KKNT,TIMEZ
COMMON /S7/     AVANG(IDMN,JDMN),TSS(IDMN,JDMN)
COMMON /WN/     WS(IDMN,JDMN),WD(IDMN,JDMN)
DIMENSION EZZ(NNA2)

```

The parametric formulation is taken from the second generation model.

Definition of 4 primary parameters affecting wave growth is given

```
C ALPSTR = IS THE 'UNIVERSAL' (NOT EVEN DEPENDENT ON ANYTHING)
C          EQUILIBRIUM CONSTANT
C BETA1   = IS THE RATIO OF THE ENERGY-WEIGHTED AVERAGE PHASE SPEED
C          TO THE PHASE SPEED OF THE SPECTRAL PEAK (ALONG THE
C          MEAN DIRECTION OF PROPAGATION)
C XLAM    = IS THE FACTOR WHICH CONVERTS FROM FM PARAMETERS TO TOTAL
C          ENERGY. IT'S THE RATIO OF THE TOTAL ENERGY TO THE INTEGRAL
C          OF THE EQUILIBRIUM RANGE FROM FM TO INFINITY.
C CONJON   = IS THE JONSWAP CONSTANT FOR USTAR NORMALIZATION
C          REPRESENTED BY M1 IN RESIO & PERRIE.
C CDFAC   = IS THE ASSUMED COEFFICIENT OF DRAG FOR JONSWAP DATA SET
C          USED IN CONVERTING FROM PARAMETERS NORMALIZED BY U TO USTAR
C SIGB    = IS THE JONSWAP PARAMETER FOR F > FM
```

Note that in this region of the spectrum, directional energies are always symmetric about the wind direction. We assume an f^{-4} in the equilibrium range for simplicity.

```
1100 ZNRMH = 2.0/PI
      FPMAX = FMN
      LMID  = NANG2/2
      EPMAX = 0.0
      DO 85 K=NFRQ+1,NFRQ2
      :
      :
      :
      :
      :
      :
      85 CONTINUE
      FMA(I,J) = FMN
      RETURN
600 FM    = FMZ
      ZNRM = ANRM
      DO 281 K=NFRQ+1,NFRQ2
      F5      = FRQA(K)**(-5)
      :
      :
      :
      281 CONTINUE
```

Now we call K2K4SET to set the energy in the fine-mesh K2-K4 grid. This is one of the essential new elements of the model.

```
SUBROUTINE K2K4SET
```

Here we interpolate the density matrix onto the PP matrix.

```
NRNG = NFRQ2
P120 = TWOP1 / 3.0
XINC = WKA(NRNG) / IRS
AINCQ = 2.0*P120 / (IAS-1)
NMID = IAS/2 + 1
XMID = NMID + 0.5
WKAT = 1.0 / XINC
AKAT = 1.0 / AINCQ
DO 1300 I=1,IRS
    XK = I * XINC
    IF (XK.LT.TKMIN) GOTO 1299
    K1 = (LOG(XK)-LOG(TKMIN)) / LOG(PWR) + 1.0001
    IF (K1.LT.1) GOTO 1299
    IF (K1.GT.NRNG) GOTO 1299
    K2 = K1 + 1
    IF (K2.GT.NRNG) GOTO 1299
    WTK1 = (WKA(K2)-XK) / (WKA(K2)-WKA(K1))
    WTK2 = 1.0 - WTK1
    DO 1301 J=1,IAS
        ANGZ = (J-1)*AINCQ - P120
        J1 = ANGZ/AINC + NANGP
        IF (J1.LT.1) J1 = 1
        IF (J1.GT.NANG2) J1 = NANG2
        J2 = J1 + 1
        IF (J2.GT.NANG2) J2 = NANG2
        WTA2 = (ANGZ-ANGL2(J1)) / AINC
        WTA1 = 1.0 - WTA2
        IF ((WTK1.LT.0.0).OR.(WTK2.LT.0.0)) THEN
            PRINT *, 'K2K4SET: ***** ', WTK1,WTK2
            STOP
        ENDIF
        X1 = DENS(K1,J1)*WTK1 + DENS(K2,J1)*WTK2
        X2 = DENS(K1,J2)*WTK1 + DENS(K2,J2)*WTK2
        PP(I,J) = X1*WTA1 + X2*WTA2
1301    CONTINUE
        GOTO 1300
1299    CONTINUE
        DO 1302 J=1,IAS
            PP(I,J)=0.
1302    CONTINUE
1300    CONTINUE
```

We call subroutine IREGION to set the region of K1-K3 space that must be include in the evaluation of SNL (also denoted S_{nl}).

SUBROUTINE IREGION

```
COMMON /A3/ G,PI,DL,NANG,NFRQ,NX,NY,TWOP,NPTS,DT,PI2,  
+ FDISC,NFRQ2,NANG2,NANGP,NSPEC,FMZ,ALP,GAM,NNZ,ANRM
```

```
COMMON /B/ WKA(NNR),OMA(NNR),FRQA(NNR),  
+ COSAN(NNA1),SINAN(NNA1),EDENS(NNR),EDENS2(NNR)  
COMMON /C/ SUMINT(NNR,NNA2),PHA(NNR),SCALF(NNR),BOLSCL(NNR),  
+ DENS(NNR,NNA2),EDB(NNR)  
COMMON /D/ IDO(NNR,NNA2),IMX(NNR,NNA2),TKMAX,TKMIN  
COMMON /OPT/ MSTA,NSTR,NORD,JPRC,NTMS,NHR,IOUT(70),JOUT(70),KTIMES,  
+ MXHR,DLAT,XLATO,DZRO,NSTA(70),ALT(70),ALON(70),PWR
```

The task in IREGION is to set up the matrix of points for which S_{nl} must be computed. We first find the maximum value of the density matrix.

```
DO 400 IRNG=1,NRNG  
DO 400 IANG=1,NANG2  
:  
400 CONTINUE  
DO 401 IRNG=1,NRNG  
DO 401 IANG=1,NANG2  
:  
401 CONTINUE  
DO 402 IRNG=1,NRNG  
DO 402 IANG=1,NANG2  
:  
:  
:  
IF (ISUM.GT.0) IMX(IRNG,IANG) = 1  
402 CONTINUE
```

Now we put the results of INITSL, BSET, K2K4SET and IREGION together and proceed to compute SNL. We call SNL to compute S_{nl} source term. This is in principle the same as is documented in Tracy and Resio (1982).

SUBROUTINE **SNL(IX,JY)**

```
COMMON /A/      XLOC2(NPA),YLOC2(NPA),XLOC4(NPA),YLOC4(NPA),DS(NPA)
COMMON /A3/     G,PI,DL,NANG,NFRQ,NX,NY,TWOPI,NPTS,DT,PI2,
+                  FDISC,NFRQ2,NANG2,NANGP,NSPEC,FMZ,ALP,GAM,NNZ,ANRM
COMMON /B/      WKA(NNR),OMA(NNR),FRQA(NNR),
+                  COSAN(NNA1),SINAN(NNA1),EDENS(NNR),EDENS2(NNR)
COMMON /C/      SUMINT(NNR,NNA2),PHA(NNR),SCALF(NNR),BOLSCL(NNR),
+                  DENS(NNR,NNA2),EDB(NNR)
COMMON /D/      IDO(NNR,NNA2),IMX(NNR,NNA2),TKMAX,TKMIN
COMMON /EFG/    PP(IJS,IAS),ANGL2(NNA2),AKAT,WKAT,XMID,
+                  GRAD(NPA,NNA1,NNR),WK2(NPA,NNA1,NNR),AK2(NPA,NNA1,NNR),
+                  WK4(NPA,NNA1,NNR),AK4(NPA,NNA1,NNR)
COMMON /OPT/   MSTA,NSTR,NORD,JPRC,NTMS,NHR,IOUT(70),JOUT(70),KTIMES,
+                  MXHR,DLAT,XLATO,DZRO,NSTA(70),ALT(70),ALON(70),PWR
COMMON /01/     FREQ(IF),SINR(IA),COSR(IA),DELF(IF),AINC
COMMON /04/     EN(IDMN,JDMN,IA)
COMMON /05/     IBOUND(IDMN,JDMN)
COMMON /SN/     SNLIJ(IF,IA),LSHFT
COMMON /S2/     FMA(IDMN,JDMN),PSUM,DSUM,KKNT,TIMEZ

DO 5 K=1,NRNG
DO 5 IANG=1,NANG2
  SUMINT(K,IANG) = 0.0
5 CONTINUE
DO 1 K=1,NFRQ
DO 1 L=1,NANG
  SNLIJ(K,L) = 0.0
1 CONTINUE
DO 50 IRNG=1,NFRQ
  SCALE = SCALF(IRNG)
  NST = IRNG + 1
  DO 56 IANG=1,NANG2
    IF (IMX(IRNG,IANG).EQ.0) GOTO 56
    ANG1 = (IANG-1)*AINC - P120
    D1 = DENS(IRNG,IANG)
    DO 60 KRNG=NST,NRNG
      NDIF = KRNG - IRNG
      DO 61 KANG=1,NANG2
        IF (IMX(KRNG,KANG).EQ.0)      GOTO 61
        ANG3 = (KANG-1)*AINC - P120
        IF (ABS(ANG1-ANG3).GT.PICUT) GOTO 61
        D3 = DENS(KRNG,KANG)
        SGN = 1.0
        IF (ANG3.LT.ANG1) SGN = -1.0
        T31 = 0.0
        DGM1 = 0.0
        DGM2 = 0.0
        DX13 = D1 * D3
        DS13 = D3 - D1
        NREF = IABS(KANG-IANG) + 1
```

This loop computes the contribution of a single contour integral for a fixed K1-K3 combination. K2 and K4 vary around the contour which is computed via the locus equation in the set-up section of this program. THFNC eliminates insignificant contributions.

```

DO 70 IPT=1,NPTS
  IF (GRAD(IPT,NREF,NDIF).LT.1.0E-13) GOTO 70
  WW2 = WK2(IPT,NREF,NDIF) * SCALE
  AA2 = AK2(IPT,NREF,NDIF) * SGN + ANG1
  WW4 = WK4(IPT,NREF,NDIF) * SCALE
  AA4 = AK4(IPT,NREF,NDIF) * SGN + ANG1
  IW2 = WW2*WKAT+0.5
  IA2 = AA2*AKAT+XMid
  IW4 = WW4*WKAT+0.5
  IA4 = AA4*AKAT+XMid
  IF (IW2.LT.1) GOTO 70
  IF (IW2.GT.IRS) GOTO 70
  IF (IA2.LT.1) GOTO 70
  IF (IA2.GT.IAS) GOTO 70
  IF (IW4.LT.1) GOTO 70
  IF (IW4.GT.IRS) GOTO 70
  IF (IA4.LT.1) GOTO 70
  IF (IA4.GT.IAS) GOTO 70
  D2 = PP(IW2,IA2)
  D4 = PP(IW4,IA4)
  DGM1 = DGM1 + (D4-D2) * GRAD(IPT,NREF,NDIF)
  DGM2 = DGM2 + D4*D2 * GRAD(IPT,NREF,NDIF)
70      CONTINUE
  T31 = DX13 * DGM1 + DS13*DGM2
  TR31 = 2.0 * T31 * BOLSCL(IRNG)
  SUMINT(IRNG,IANG) = SUMINT(IRNG,IANG)+TR31*PHA(KRNG)
  SUMINT(KRNG,KANG) = SUMINT(KRNG,KANG)-TR31*PHA(IRNG)
61      CONTINUE
60      CONTINUE
56      CONTINUE
  CG = G / (2.0*OMA(IRNG))
  SNL1D = 0.0
  LMID = NANG2 / 2
  DO 180 IANG=1,NANG2
    IANGP = IANG + LSHFT - LMID
    IF (IANGP.LT.1) IANGP = IANGP + NANG
    IF (IANGP.GT.NANG) IANGP = IANGP - NANG
    SNLIJ(IRNG,IANGP)=SUMINT(IRNG,IANG)*WKA(IRNG)*OMA(IRNG)*TWOPI/CG
    SNL1D = SNL1D + SNLIJ(IRNG,IANGP)
180      CONTINUE
  SNL1D = SNL1D * AINC
50      CONTINUE

```

We now compute the contribution due to wave-breaking dissipation S_{ds} . We call SDISS(I,J).

```

SUBROUTINE SDISS(I,J)
COMMON /A3/ G, PI, DL, NANG, NFRQ, NX, NY, TWOP1, NPTS, DT, PI2,
+ FDISC, NFRQ2, NANG2, NANGP, NSPEC, FMZ, ALP, GAM, NNZ, ANRM
COMMON /B/ WKA(NNR), OMA(NNR), FRQA(NNR),
+ COSAN(NNA1), SINAN(NNA1), EDENS(NNR), EDENS2(NNR)
COMMON /CEN/ E( IDMN, JDMN, IF, IA)
COMMON /OPT/ MSTA, NSTR, NORD, JPRC, NTMS, NHR, IOUT(70), JOUT(70), K TIMES,
+ MXHR, DLAT, XLATO, DZRO, NSTA(70), ALT(70), ALON(70), PWR
COMMON /O1/ FREQ(IF), SINR(IA), COSR(IA), DELF(IF), AINC
COMMON /O4/ EN(IDMN, JDMN, IA)
COMMON /O5/ IBOUND(IDMN, JDMN)
COMMON /SD/ SDISIJJ(IF, IA)
COMMON /S2/ FMA(IDMN, JDMN), PSUM, DSUM, KKNT, TIMEZ
COMMON /S4/ HSIG(IDMN, JDMN)
COMMON /WN/ WS(IDMN, JDMN), WD(IDMN, JDMN)
OMMX = TWOP1 * FMA(I, J)
EZED = HSIG(I, J)**2/16.0
STP = 3.39*EZED*(TWOP1*FMA(I, J))**4
BRCON = 0.75E-5
DO 1 K=1, NFRQ
DO 1 L=1, NANG
SDISIJJ(K, L) = -BRCON * OMA(K) * STP * E(I, J, K, L)
1 CONTINUE

```

We compute the energy input by the wind S_{in} . We call SINPUT

```

SUBROUTINE SINP(I, J)
COMMON /A3/ G, PI, DL, NANG, NFRQ, NX, NY, TWOP1, NPTS, DT, PI2,
+ FDISC, NFRQ2, NANG2, NANGP, NSPEC, FMZ, ALP, GAM, NNZ, ANRM
COMMON /B/ WKA(NNR), OMA(NNR), FRQA(NNR),
+ COSAN(NNA1), SINAN(NNA1), EDENS(NNR), EDENS2(NNR)
COMMON /CDR/ COEF1, COEF2, SPDFAC, XLAMZ, ALPSTR, BETA1, EPMAX, FPMAX
COMMON /CEN/ E(IDMN, JDMN, IF, IA)
COMMON /OPT/ MSTA, NSTR, NORD, JPRC, NTMS, NHR, IOUT(70), JOUT(70), K TIMES,
+ MXHR, DLAT, XLATO, DZRO, NSTA(70), ALT(70), ALON(70), PWR
COMMON /O1/ FREQ(IF), SINR(IA), COSR(IA), DELF(IF), AINC
COMMON /O4/ EN(IDMN, JDMN, IA)
COMMON /O5/ IBOUND(IDMN, JDMN)
COMMON /SI/ SINIJ(IF, IA)
COMMON /S2/ FMA(IDMN, JDMN), PSUM, DSUM, KKNT, TIMEZ
COMMON /WN/ WS(IDMN, JDMN), WD(IDMN, JDMN)
OMMX = FMA(I, J)*TWOP1
RORAT = 0.0012
U10 = WS(I, J)

```

```

CDRAG = (COEF1+COEF2*U10)*0.001
UST = SQRT(CDRAG) * U10
CFAC = SQRT(CDRAG/1.33E-3)
SUMIN = 0.0
DO 5 K=1,NFRQ
    CC = G / OMA(K)
    DO 1 L=1,NANG
        ANGLE = WD(I,J) - (L-1)*AINC
        IF (ANGLE.GT.PI) ANGLE = TWOPI - ANGLE
        IF (ANGLE.LT.-PI) ANGLE = TWOPI + ANGLE
        IF (ABS(ANGLE).GT.PI2) GOTO 2
        SINIJ(K,L) = 0.105*RORAT * (U10/CC*COS(ANGLE)-1.0) *
            OMA(K)*E(I,J,K,L)*CFAC
+
        IF (SINIJ(K,L).LT.0.) SINIJ(K,L) = 0.0
        SUMIN = SUMIN + SINIJ(K,L) * DELF(K)
        GOTO 1
2     SINIJ(K,L) = 0.0
1     CONTINUE
5 CONTINUE

```

Finally, we add the contributions of these source terms to form the new energy density at this time step. We do sums on each of the source terms and get ready to consider the problem of negative energies.

```

DO 101 K=1,NFRQ
    DO 1028 L=1,NANG
        SUMZ = SUMZ + E(I,J,K,L)
1028 CONTINUE
        EOLD(K) = SUMZ*AINC
        DO 102 L=1,NANG
            E(I,J,K,L) = E(I,J,K,L)+(SNLIJ(K,L)+SDISSIJ(K,L)+SINIJ(K,L))*DT
            IF (E(I,J,K,L).LT.0.0) SUMNEG = SUMNEG + E(I,J,K,L)
            IF (E(I,J,K,L).LT.1.0E-6) E(I,J,K,L) = 0.0
            SUMZ1 = SUMZ1 + SINIJ(K,L)
            SUMZ2 = SUMZ2 + SNLIJ(K,L)
            SUMZ3 = SUMZ3 + E(I,J,K,L)
            SUMZ4 = SUMZ4 + SDISSIJ(K,L)
102     CONTINUE

```

Negative energy is set to zero.

```

    RR = (SUMZ3-SUMNEG)/SUMZ3
    IF (RR.LT.0.) RR = 0.0
    DO 4040 L=1,NANG
        E(I,J,K,L) = RR * E(I,J,K,L)
4040 CONTINUE

```

We complete the sums, multiplying by the correct increments, and re-calculate the maximum energy.

```

4041    SNL1D(K) = SUMZ2 * AINC * DT
        SIN1D(K) = SUMZ1 * AINC * DT
        SDS1D(K) = SUMZ4 * AINC * DT
        EIJ1D(K) = SUMZ3 * AINC
        DSUM = DSUM + EIJ1D(K)*DELF(K)
        IF (EIJ1D(K).LT.EMAX) GOTO 101
        EMAX = EIJ1D(K)
        KMAX = K
101    CONTINUE

```

We (OPTIONALLY) force the high frequency tail to be f^{-4} variation. This is a big assumption!

```

IKK = 0
DO 105 K=1,NFRQ
    IF (IKK.EQ.0) FFF = FREQ(K)
    IF (IKK.EQ.0) EEE = EIJ1D(K)
    IF (FREQ(K).LT.3.0*FMA(I,J)) GOTO 105
    IKK = 1
    ZZ = 1.0
    IF (FREQ(K).GT.3.5*FMA(I,J)) ZZ = 3.0*FMA(I,J)/FREQ(K)
    RR = (EEE*(FFF/FREQ(K))**4)/EIJ1D(K) * ZZ
    EIJ1D(K) = EEE*(FFF/FREQ(K))**4
    DO 106 L=1,NANG
        E(I,J,K,L) = RR * E(I,J,K,L)
106    CONTINUE
105    CONTINUE

```

We re-consider the maximum energy and peak frequency f_p , and we compute the significant wave height H_s ,

```

DO 3000 K=1,NFRQ
    IF (EIJ1D(K).LT.EMAX) GOTO 3000
    EMAX = EIJ1D(K)
    KO = K
3000 CONTINUE
    FP = FREQ(KO)
    IF (KO.EQ.NFRQ.OR.KO.EQ.1) GOTO 3005

```

```

DF1      = EIJ1D(K0) - EIJ1D(K0-1)
DF2      = EIJ1D(K0) - EIJ1D(K0+1)
DFTOT   = DF1 + DF2 + 1.0E-10
FP       = 0.5*FP + 0.5*((DF1/DFTOT)*FREQ(K0+1) +(DF2/DFTOT)*FREQ(K0-1))
3005 IF (EMAX.GT.EPMAX) FMA(I,J) = FP
HSIG(I,J) = 4.01*SQRT(DSUM+PSUM)
100 CONTINUE

```

This ends subroutine SOURCE. We now call subroutine OUTP to output variables that we have computed.

```

SUBROUTINE OUTP(IHR, IDN)
COMMON /A3/    G, PI, DL, NANG, NFRQ, NX, NY, TWOPI, NPTS, DT, PI2,
+                  FDISC, NFRQ2, NANG2, NANGP, NSPEC, FMZ, ALP, GAM, NNZ, ANRM
COMMON /CDR/   COEF1, COEF2, SPDFAC, XLAMZ, ALPSTR, BETA1, EPMAX, FPMAX
COMMON /CEN/   E(IDMN, JDMN, IF, IA)
COMMON /OPT/   MSTA, NSTR, NORD, JPRC, NTMS, NHR, IOUT(70), JOUT(70), KTIMES,
+                  MXHR, DLAT, XLATO, DZRO, NSTA(70), ALT(70), ALON(70), PWR
COMMON /O1/     FREQ(IF), SINR(IA), COSR(IA), DELF(IF), AINC
COMMON /O5/     IBOUND(IDMN, JDMN)
COMMON /S4/     HSIG(IDMN, JDMN)
COMMON /S2/     FMA(IDMN, JDMN), PSUM, DSUM, KKNT, TIMEZ
COMMON /S7/     AVANG(IDMN, JDMN), TSS(IDMN, JDMN)
COMMON /TM/     TIMTOT
COMMON /WN/     WS(IDMN, JDMN), WD(IDMN, JDMN)
DIMENSION ETMP1(IF), TOUT(IDMN, JDMN)

```

The output is done by looping through the number of output stations, writing requested dynamical variables.

```

DO 100 ISTA=1,MSTA
I = IOUT(ISTA)
J = JOUT(ISTA)
WNDSPD = WS(I,J)
IWDT = WD(I,J)
HT = HSIG(I,J)

```

We sum energy over all directions at each frequency to get 1-D energy values,

```

DO 159 K=1,NFRQ
ETMP1(K) = 0.0

```

```

      DO 153 IANG=1,NANG
      ETMP1(K) = ETMP1(K) + E(I,J,K,IANG)*AINC
153   CONTINUE
      IF (ETMP1(K).LT.EMAX) GOTO 159
      EMAX = ETMP1(K)
      KMAX = K
159   CONTINUE

```

define peak spectral period and output 2-D spectral data for each observation station, to separate files, including 1-D energy values for each discrete frequency. NORD = 1 means no wind field was read in and therefore skip the following write statement.

```

TPEAK = 1.0 / FREQ(KMAX)
WRITE(11,103) TIMEZ,NSTA(ISTA),IDN,ALT(ISTA),ALON(ISTA),
+           I,J,HT,TPEAK,AVANGT,WNDSPD,IWDT

```

Loop on i,j to find the maximum energy and peak frequency,

```

DO 501 I=1,NX
DO 501 J=1,NY
      TMAIN = 1.0 / FMA(I,J)
      IF (TMAIN.GE.TSS(I,J)) GOTO 500
      EMAX = 0.0
      KMAX = 1
      DO 809 K=1,NFRQ
          SUM = 0.0
          DO 808 MA=1,NANG
              SUM = SUM + E(I,J,K,MA)
808       CONTINUE
          IF (SUM.LT.EMAX) GOTO 809
          EMAX = SUM
          KMAX = K
809       CONTINUE
      TMAIN      = 1.0 / FREQ(KMAX)
500       TOUT(I,J) = TMAIN
501       CONTINUE

```

and write files containing matrices of wave height, peak period, and average wave direction for all grid points.

```
COO = (360.0/TWOP1)
WRITE(15,*) 'SIG. WAVE HEIGHT (M)'
DO 200 J=NY,1,-1
    WRITE(15,229) (HSIG(I,J), I=1,NX)
200 CONTINUE
WRITE(15,*) 'PEAK WAVE PERIOD (SEC)'
DO 201 J=NY,1,-1
    WRITE(15,229) (TOUT(I,J)*IBOUND(I,J), I=1,NX)
201 CONTINUE
WRITE(15,*) 'AVERAGE WAVE DIRECTION (DEG)'
DO 202 J=NY,1,-1
    WRITE(15,229) (AVANG(I,J)*COO, I=1,NX)
202 CONTINUE
203 continue
END IF
```

This ends the loops that encompass subroutines RDWIN, SETSID, DPRP and SOURCE.

```
1001 CONTINUE
1000 CONTINUE
```

End of the simulation

6. INPUT EXAMPLE WITH OUTPUT RESULTS.

For a simple fetch limited growth run for a square box SWAMP ocean (SWAMP test #2), input file WINDS.DAT would be blank, as wind would be specified in OPTIONS.DAT. The latter would have the form:

```
13 13 24 20 100. 180. 11
 1   1   1 24     1      0
 1.   14.
.05    1.1772250
1    2    7    20.  25.
2    3    7    21.  25.
3    4    7    22.  25.
4    5    7    23.  25.
5    6    7    24.  25.
6    7    7    25.  25.
7    8    7    26.  25.
8    9    7    27.  25.
9   10    7    28.  25.
10   11    7    29.  25.
11   12    7    30.  25.
00000000000000
0111111111110
0111111111110
0111111111110
0111111111110
0111111111110
0111111111110
0111111111110
0111111111110
0111111111110
0111111111110
0111111111110
0111111111110
0111111111110
0111111111110
0111111111110
00000000000000
20    0    1.5    0    0    0    0    1
0.01  0.25  3.3    2   .6366    0
```

Figure 5. Example OPTIONS.DAT for SWAMP fetch limited growth test.

To explain the numbers in this file:

a). line 1 reads 'OPTIONS RECORD 1'

READ(20,*) NX,NY,NANG,NFRQ,DL,DT,MSTA

specifying a square ocean with NX=NY=13, the number of angles on 180° ; NANG=24, the number of frequency bands in the discrete grid; NFRQ=20, the space grid is 100km; DL=100, the time step is 180s; DT=180 and the number of output points is 11; MSTA=11.

b). line 2 reads 'OPTIONS RECORD 2'

```
READ(20,*)      NORD,NTMS,NHR,MXHR,NSPEC,ICON
```

specifying constant winds; NORD=1, 1 timestep between outputs; NTMS=1, 24hr run; NHR=24, initialization from parametric spectra; NSPEC=1 and unconstrained high frequency tail (observations suggest f^{-4} and f^{-5} dependencies and some models use either variation); ICON=0.

c). line 3 reads 'OPTIONS RECORD 3'

```
READ(20,*)      DLAT,XLAT0
```

specifying the latitude increment of the space grid on this square ocean in degrees; DLAT=1, and the southernmost latitude of the grid in degrees; XLAT0=14.

d). line 4 reads 'OPTIONS RECORD 4'

```
READ(20,*) FZRO,PWR
```

specifying the lowest frequency of the frequency grid; FZRO=0.05, and the radial power for spacing of the frequency and wavenumber grids; PWR= 1.1772250.

e). lines 5-15 read 'OPTIONS RECORD 5'

```
IF ( MSTA .GT. 0 ) READ(20,*) (NSTA(ISTA),IOUT(ISTA),JOUT(ISTA),
+                               ALT(ISTA),ALON(ISTA), ISTA=1,MSTA)
```

specifying each station number NSTA(ISTA): ISTA=1 to 11, its (x,y) coordinates on the square ocean grid (IOUT(ISTA),JOUT(ISTA)), and the corresponding latitude-longitude coordinates (ALT(ISTA), ALON(ISTA)). These are here aligned across the ocean at fixed latitude parallel to the wind direction.

f). lines 16-28 read 'OPTIONS RECORD 6',

```
DO 295 J=NY,1,-1
    READ(20,959) (IBOUND(I,J),I=1,NX)
295 CONTINUE
```

which specifies the land-sea matrix with orientation;

GRID ORIENTATION IS:

1,1 IN LOWER CORNER
1,NY IS UPPER LEFT CORNER
TO READ TOP-DOWN LOOP OVER J=NY,1,-1.

g). line 29 reads constant wind test parameters,

```
READ(20,*) IUWS,IUDIR,FMNO,ISHFT,NSHFT,IWNDN,IDEGN,ISPEC
```

where

```
IUWS = WIND SPEED FOR TEST CASE (M/S)
IUDIR = WIND DIRECTION FOR TEST CASE
FMNO = INITIAL PEAK FREQUENCY ?
ISHFT = OPTION FOR SHIFT (0 FOR NO SHIFT 1 FOR SHIFT)
NSHFT = NUMBER OF TIME INCREMENTS (IHR LEVEL) BEFORE SHIFT
IWNDN = NEW WIND SPEED (M/S)
IDEGN = NEW DIRECTION OF WIND
ISPEC = BOUNDARY TYPE OPTION
    = 0 FOR NORMAL "SOLID" BOUNDARIES
    = 1 FOR TOP AND BOTTOM REFLECTIVE BOUNDARIES
    = 2 FOR LEFT AND RIGHT REFLECTIVE BOUNDARIES
```

h). line 30 reads

```
IF ( NSPEC .EQ. 1 ) THEN
    READ(20,*) ALP,FMZ,GAM,NNZ,ZNRM
```

which are parameters required for an initial spectrum.

With the proper format, ONEL.DAT could then produce significant wave height as a function of time and increasing in fetch (the 11 output stations positions). An example, at time step 370 and time 18.50 hr is as follows:

wind direction, windspeed, TIME, kknt					
0	20.00	18.50	370		
station coords, station number, wave height, wave period, mean wave dir					
2	7	1	0.4364E+01	0.8846E+01	0.0000E+00
3	7	2	0.5755E+01	0.1041E+02	0.0000E+00
4	7	3	0.6672E+01	0.1041E+02	0.0000E+00
5	7	4	0.7290E+01	0.1130E+02	0.0000E+00
6	7	5	0.7682E+01	0.1130E+02	0.0000E+00
7	7	6	0.7915E+01	0.1130E+02	0.0000E+00
8	7	7	0.8046E+01	0.1130E+02	0.0000E+00
9	7	8	0.8116E+01	0.1226E+02	0.0000E+00
10	7	9	0.8152E+01	0.1226E+02	0.0000E+00
11	7	10	0.8169E+01	0.1226E+02	0.0000E+00
12	7	11	0.8175E+01	0.1226E+02	0.0000E+00

This is in good agreement with SWAMP-type results. Corresponding output for the entire space grid from WAVES.DAT at the same time 18.50 hr, could have the form,

SIG. WAVE HEIGHT (M)															
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	4.0	5.2	6.0	6.5	6.8	7.0	7.1	7.2	7.2	7.3	7.3	7.3	0.0		
0.0	4.2	5.5	6.3	6.9	7.3	7.5	7.6	7.7	7.7	7.7	7.8	7.8	0.0		
0.0	4.3	5.6	6.5	7.1	7.5	7.7	7.9	7.9	8.0	8.0	8.0	8.0	0.0		
0.0	4.3	5.7	6.6	7.2	7.6	7.8	8.0	8.0	8.1	8.1	8.1	8.1	0.0		
0.0	4.3	5.7	6.6	7.3	7.7	7.9	8.0	8.1	8.1	8.2	8.2	8.2	0.0		
0.0	4.3	5.7	6.6	7.3	7.7	7.9	8.0	8.1	8.1	8.2	8.2	8.2	0.0		
0.0	4.4	5.8	6.7	7.3	7.7	7.9	8.0	8.1	8.1	8.2	8.2	8.2	0.0		
0.0	4.4	5.8	6.7	7.3	7.7	7.9	8.0	8.1	8.1	8.2	8.2	8.2	0.0		
0.0	4.4	5.7	6.6	7.3	7.6	7.9	8.0	8.1	8.1	8.1	8.1	8.1	0.0		
0.0	4.3	5.7	6.6	7.2	7.5	7.8	7.9	7.9	8.0	8.0	8.0	8.0	0.0		
0.0	4.3	5.6	6.4	7.0	7.3	7.5	7.7	7.7	7.8	7.8	7.8	7.8	0.0		
0.0	4.1	5.3	6.1	6.6	6.9	7.1	7.2	7.2	7.3	7.3	7.3	7.3	0.0		
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0		

We plot the fetch growth curves for a SWAMP-type square-box ocean in Figure 6 using a variation of the source terms for wind input and dissipation. These are compared to the JONSWAP growth relation found by Hasselmann et al (1973). It is clear that an appropriate choice of source terms allows the model to be easily tuned to JONSWAP or any other fetch relations.

7. PRESENT MODEL STATUS AND FUTURE PLANS

The model is presently undergoing a number of tests to expose its characteristics and to help us know where bugs exist in the code and where errors have been made in the structure of the model. These include SWAMP-type tests to allow comparison with other models and with data collected from systems with simple geometry. This phase of the work will continue until there is a reasonable degree of confidence in the integrity of the model.

Furthermore, the present propagation scheme within the model is only first order. This is inadequate for many sophisticated applications. Therefore an effort is also being made to upgrade the propagation scheme to a Lax-Wendroff scheme. At the conclusion of this upgrading and these tests, the code will generally be available for application to ocean wave modelling in all its diversity.

8. Acknowledgement

The Federal Panel on Energy Research and Development (PERD) provided funding for this project.

9. References

Hasselmann, K. (1961) On the nonlinear energy transfer in a gravity-wave spectrum. Part I. General Theory. *J. Fluid Mech.* 12, 481-500.

Hasselmann, K., Barnett, T. P., Bouws, E., Carlson, H., Cartwright, D. E., Enke, K., Ewing, J. A., Gienapp, H., Hasselmann, D. E., Kruseman, P., Meerburg, A., Müller, P., Olbers, D. J., Richter, K., Sell, W., Walden, H., (1973) Measurements of wind-wave growth and swell decay during the Joint North Sea Wave Project (JONSWAP). *Deutsche Hydrograph. Zeit.*, Erganzungsheft Reihe A (8°), Nr. 12.

Hasselmann, S., Hasselmann, K., Komen, G. K., Janssen, P., Ewing, J. A. and Cardone, V. (1989) The WAM model- A third generation ocean wave prediction model. *J. Phys. Oceanogr.*, 18, 1775-1810.

Perrie, W. and Resio, D.(1993) A numerical study of nonlinear energy fluxes due to wave-wave interactions Part II: Equilibrium range constraints. Submitted to *J. Phys. Oceanogr.*

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.* 69, 5181-5190.

Resio D. T. and Perrie W. A. (1989) Implications of an f^{-4} equilibrium range for wind-generated waves. *J. Phys. Oceanogr.* 19, No.2, 193-204.

Resio, D. and Perrie, W. (1991) A numerical study of nonlinear energy fluxes due to wave-wave interactions Part I: Methodology and basic results. *J.Fluid Mech.* 223, 603-629.

Tracy, B.A. and Resio, D.T. (1982) Theory and calculations of the nonlinear energy transfer between sea waves in deep water. *WES Rep.* 11. US Army Engineer Waterways Experiment Station, Vicksburg, MS.

SWAMP group. (1985) Ocean Wave Modeling. Plenum, New York. 256 pp.

10. Caption for Figure 6.

The fetch growth curves for a SWAMP-type square-box ocean using a variation of the source terms for wind input and dissipation, in comparison with the JONSWAP growth relation found by Hasselmann et al (1973).

