

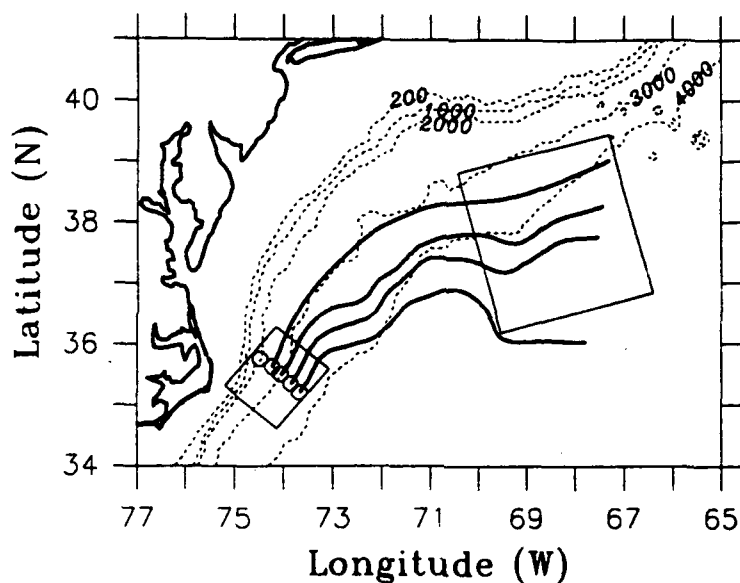
AD-A267 014



GRADUATE SCHOOL OF OCEANOGRAPHY
UNIVERSITY OF RHODE ISLAND
NARRAGANSETT, RHODE ISLAND

3

Ray Tracing on Topographic Rossby Waves



DTIC
ELECTE
JUL 22 1993
S A D

by

Christopher Meinen

Erik Fields

Robert Pickart

D. Randolph Watts

This document has been approved
for public release and sale; its
distribution is unlimited.

GSO Technical Report No. 93-1
May, 1993

This research program has been sponsored by the National Science Foundation under grant number OCE87-17144 and by the Office of Naval Research under contracts N00014-90J-1568 and N00014-92J-4013.

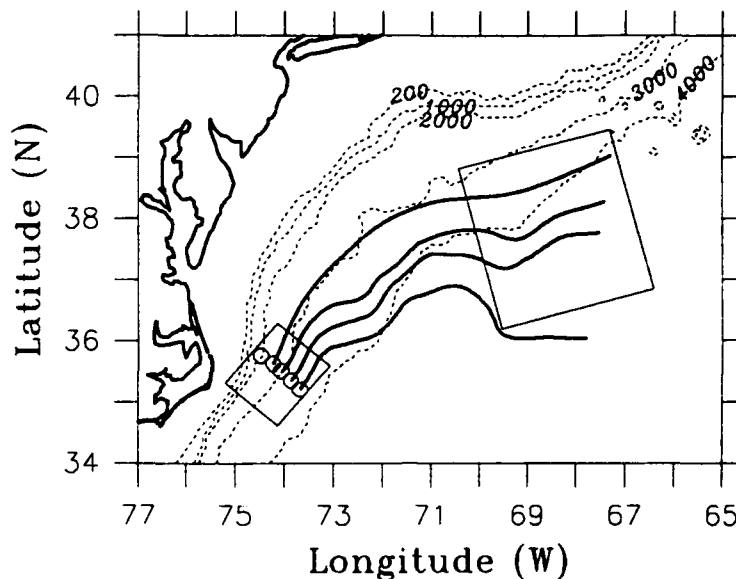
93-16530



93 7 21 03 7

GRADUATE SCHOOL OF OCEANOGRAPHY
UNIVERSITY OF RHODE ISLAND
NARRAGANSETT, RHODE ISLAND

Ray Tracing on Topographic Rossby Waves



DTIC QUALITY INSPECTED 5

by

Christopher Meinen

Erik Fields

Robert Pickart

D. Randolph Watts

Accession For	
NTIS CRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

GSO Technical Report No. 93-1
May, 1993

This research program has been sponsored by the National Science Foundation under grant number OCE87-17144 and by the Office of Naval Research under contracts N00014-90J-1568 and N00014-92J-4013.

Abstract

Topographic Rossby Waves (TRWs) have been identified with the largest variability in deep current meter records along the continental slope in the Mid-Atlantic Bight (MAB). Ray tracing theory is applied to TRWs using the real bottom topography of the MAB and the observed stratification. The dispersion relation for TRWs is derived and various wavenumber limits are discussed. A computational method for tracing the waves is presented, including the necessity of smoothing the bathymetry. In the examples shown, TRWs with periods of 24–48 days generally propagate southwestward, changing their wavelengths from 400 to 100 kilometers in response to the change in bottom slope. TRW paths are shown that connect from the SYNOP Central Array near 68°W to the SYNOP Inlet Array near Cape Hatteras.

Contents

List of Figures	iii
1 Introduction	1
2 Historical Significance of TRWs in the Mid-Atlantic Bight	1
3 Dynamics and Kinematics of TRWs	2
4 Ray Tracing	8
5 Wave Tracing Programs	12
5.1 Preparing the Topography	12
5.1.1 The Raw Bathymetry	13
5.1.2 Remove unwanted features	13
5.1.3 Filtering	14
5.1.4 Compute spline representation	15
5.2 Getting the initial wavenumber	15
5.3 Tracing	16
5.4 Examples	17
6 References	24
7 Acknowledgements	24
8 Appendix A The Dispersion relation derivation	25
9 Appendix B Example Bottom Topography Preparation.	30
10 Appendix C Program Codes	34

List of Figures

1	Location of SYNOP Inlet and Central Arrays	6
2	Bottom-trapped dispersion relation	7
3	Uniform, barotropic dispersion relation	9
4	Numerical solution for $\beta = 0$ dispersion relation.	9
5	Numerical solution for full dispersion relation.	10
6	Numerical solutions for various periods	10
7	B3 launch with $dt = -1$ hrs and $dt = -8$ hrs	19
8	Comparison of $dt = -1$ hrs and $dt = -8$ hrs	20
9	B3 launch for different filter sizes	22
10	Launches from sites B2, B3, and B4	22
11	B3 launch with different periods	23
12	Real bottom topography	31
13	Topography with seamounts removed	31
14	110 km filtered topography	32
15	220 km filtered topography	32
16	330 km filtered topography	33

1 Introduction

This report describes the method of ray tracing to find the path that Topographic Rossby Waves follow for realistic bottom topography and stratification. The report gives some background on Topographic Rossby Waves (TRWs) and explains their kinematics and dynamics. It then documents a set of programs that allow for the development of TRW ray paths.

2 Historical Significance of TRWs in the Mid-Atlantic Bight

TRWs are transverse, quasi-geostrophic waves found in regions of sloping topography. They are very similar to ordinary Rossby Waves in that the sloping topography has a dynamically equivalent effect to the variation in Coriolis parameter. TRWs thus propagate "westward" along the continental slope, traveling with the shallow water on their right. TRWs have been detected by moored arrays of current meters at numerous locations in the Mid-Atlantic Bight. Their strong horizontal current variance overwhelms the signal of the Deep Western Boundary Current and that of deep expressions of Gulf Stream meanders. By measuring the horizontal current variance information about the TRWs is obtained.

The source of the TRWs in the Mid-Atlantic Bight has not been clearly identified yet. One possibility that has been suggested is that the TRWs are created by large Gulf Stream meanders and ring formation [Hogg, 1981]. Although there appears to be no direct coupling between TRWs and the above Gulf Stream near Cape Hatteras [Johns and Watts, 1985], a connection between the presence of TRWs and the existence of troughs and ring formation downstream has been reported by Schultz [1987]. Thus it appears that the TRWs may be remotely forced by these meanders and rings that occur farther to the northeast. This remains a question for later investigation.

3 Dynamics and Kinematics of TRW_s

To derive the dispersion relation for TRWs one solves the equations of motion for a single governing equation in terms of the pressure p . The equations of motion consist of the linearized horizontal momentum equations,

$$\frac{\partial u}{\partial t} - fv = -\frac{1}{\rho_0} \frac{\partial p'}{\partial x} \quad (1)$$

$$\frac{\partial v}{\partial t} + fu = -\frac{1}{\rho_0} \frac{\partial p'}{\partial y} \quad (2)$$

the incompressibility condition,

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \quad (3)$$

and the hydrostatic equation and linearized density equation,

$$-\frac{\partial p'}{\partial z} - \rho' g = 0 \quad (4)$$

$$\frac{\partial \rho'}{\partial t} - \frac{\rho_0 N^2 w}{g} = 0 \quad (5)$$

These are then expressed quasi-geostrophically. Because TRWs are bottom trapped it is possible to follow Hogg [1981] and assume that $N = N_B(x, y)$ the Brunt-Vaisälä frequency near the ocean bottom. The governing equation becomes,

$$\frac{\partial}{\partial t} \left[\nabla^2 p' + \frac{f_0^2}{N_B^2} \frac{\partial^2 p'}{\partial z^2} \right] + \beta \frac{\partial p'}{\partial x} = 0 \quad (6)$$

Detailed derivations of these equations are given in Appendix A. The relevant boundary conditions are no normal flow through either the top or the (sloped) bottom: $w = 0$ at $z = 0$ and $w = \mathbf{u} \cdot -\nabla \mathbf{h}$ at $z = -h(x, y)$. These conditions result in

$$\frac{\partial^2 p'}{\partial t \partial z} = 0 \quad \text{at} \quad z = 0 \quad (7)$$

and

$$\frac{\partial^2 p'}{\partial t \partial z} = \frac{N_B^2}{f_0} \left[\frac{\partial p'}{\partial x} \frac{\partial h}{\partial y} - \frac{\partial p'}{\partial y} \frac{\partial h}{\partial x} \right] \quad \text{at} \quad z = -h(x, y) \quad (8)$$

Plane wave solutions of (6) are sought of the form

$$p = A(z) \exp[i(kx + ly - \omega t)] \quad (9)$$

Substituting (9) into (6) and using the two boundary conditions (7) and (8) it is possible to get the dispersion relation. Specifically, substituting (9) into (6) gives

$$\frac{\partial^2 A}{\partial z^2} - A \left[\frac{N_B^2}{f_0} (k^2 + l^2 + \frac{\beta k}{\omega}) \right] = 0 \quad (10)$$

which can be expressed as

$$\frac{\partial^2 A}{\partial z^2} - \lambda^2 A = 0 \quad (11)$$

where

$$\lambda^2 = \left(\frac{N_B}{f_0} \right)^2 (k^2 + l^2 + \frac{\beta k}{\omega}) \quad (12)$$

Here λ is the bottom-trapping coefficient, not the wavelength. The solution to (11), using boundary condition (7), is

$$A(z) = \cosh(\lambda z) \quad (13)$$

Substituting (9) into the bottom boundary condition (8) and using (13),

$$\lambda \tanh(\lambda h) = \frac{N_B^2}{f_0 \omega} (h_y k - h_x l) \quad (14)$$

Equations (12) and (14) constitute the dispersion relation for TRWs. It is convenient to non-dimensionalize these expressions using the following scales

$$h = H h^*$$

$$\lambda = \frac{\lambda^*}{H}$$

$$\begin{aligned}\omega &= \frac{\beta \omega^* N_B H}{f_0} \\ (k, l) &= \frac{f_0}{N_B H} (k^*, l^*) \\ \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right) &= \frac{f_0}{N_B H} \left(\frac{\partial}{\partial x^*}, \frac{\partial}{\partial y^*} \right)\end{aligned}$$

where the asterisks denote non-dimensional variables. The factor $\frac{N_B H}{f_0} = R_d$ is the internal Rossby radius of deformation. The non-dimensional form of the dispersion relation is

$$\lambda^2 = k^2 + l^2 + \frac{k}{\omega} \quad (15)$$

$$\lambda \tanh(\lambda h) = \frac{f_0}{\beta R_d} \frac{h_y k - h_x l}{\omega} \quad (16)$$

where the asterisks have been dropped. Appendix A shows this derivation in detail.

Note the following properties of TRWs:

- The amplitude, and thus the energy, decays as $\cosh(\lambda z)$ with height above the bottom; i.e., the wave is bottom trapped.
- TRW velocities are in approximate geostrophic balance.
- TRWs are transverse waves, since $\mathbf{k} \cdot \mathbf{u}_g = k u_g + l v_g = \nabla \cdot \mathbf{u}_g = 0$.
- The observed periods of TRWs in the Mid-Atlantic Bight range from 8 to 60 days [e.g. Hogg 1981; Schultz, 1987] for wavelengths of 40 to 300 km.

The (ω, k, l) dispersion relationship is commonly represented either as two-dimensional plots of $\omega(k)$ or as "slowness" curves, which are traces of k, l at a set of constant frequencies in wavenumber space. There are three limits of (12) and (14) that will be examined:

- 1) $\lambda h \rightarrow \infty$ and correspondingly $\tanh(\lambda h) \simeq 1$, which is the strongly bottom-trapped case.
- 2) $\lambda h \rightarrow 0$ and $\tanh(\lambda h) \simeq \lambda h$, which is the uniform "barotropic" case.
- 3) $\beta = 0$, which is the f-plane assumption.

In order to look at these limits it is necessary to choose a location at which to find the parameters that will be used in the dispersion relation. Pickart and Watts [1990] detected TRWs in the SYNOP Inlet Array, so this is the area that will be focused on when discussing the limits. Figure 1 shows the location of the SYNOP Inlet Array and sites B1-B5 which will be used at various parts of the report.

In the first limit, (14) becomes

$$\lambda = \frac{N_B^2}{f_0\omega}(h_y k - h_x l) \quad (17)$$

which, when combined with (12), gives a result of the form

$$Ak^2 + Bkl + Cl^2 + Dk = 0 \quad (18)$$

where

$$A = \omega^2 - N_B^2 h_y^2$$

$$B = 2N_B^2 h_x h_y$$

$$C = -N_B^2 h_x^2$$

$$D = \beta\omega$$

This is an equation for either a rotated hyperbola or a rotated ellipsoid, depending on whether A and C are either the same or opposite signs. For the site used here they are both negative. Figure 2 shows the solution to (18) for a wave with a period of 40 days and $N_B = .001s^{-1}$. The bottom slope was specified as $h_x = -0.018718$ and $h_y = 0.0079937$, and the downslope gradient is $\arctan(\frac{h_y}{h_x}) = -23.125$ deg. These parameters were chosen because they correspond to site B3 of the SYNOP Inlet Array (Figure 1) where TRWs were observed by Pickart and Watts [1990]. This set (ω, N_B, h_x, h_y) is referred to as the "Test B3" values.

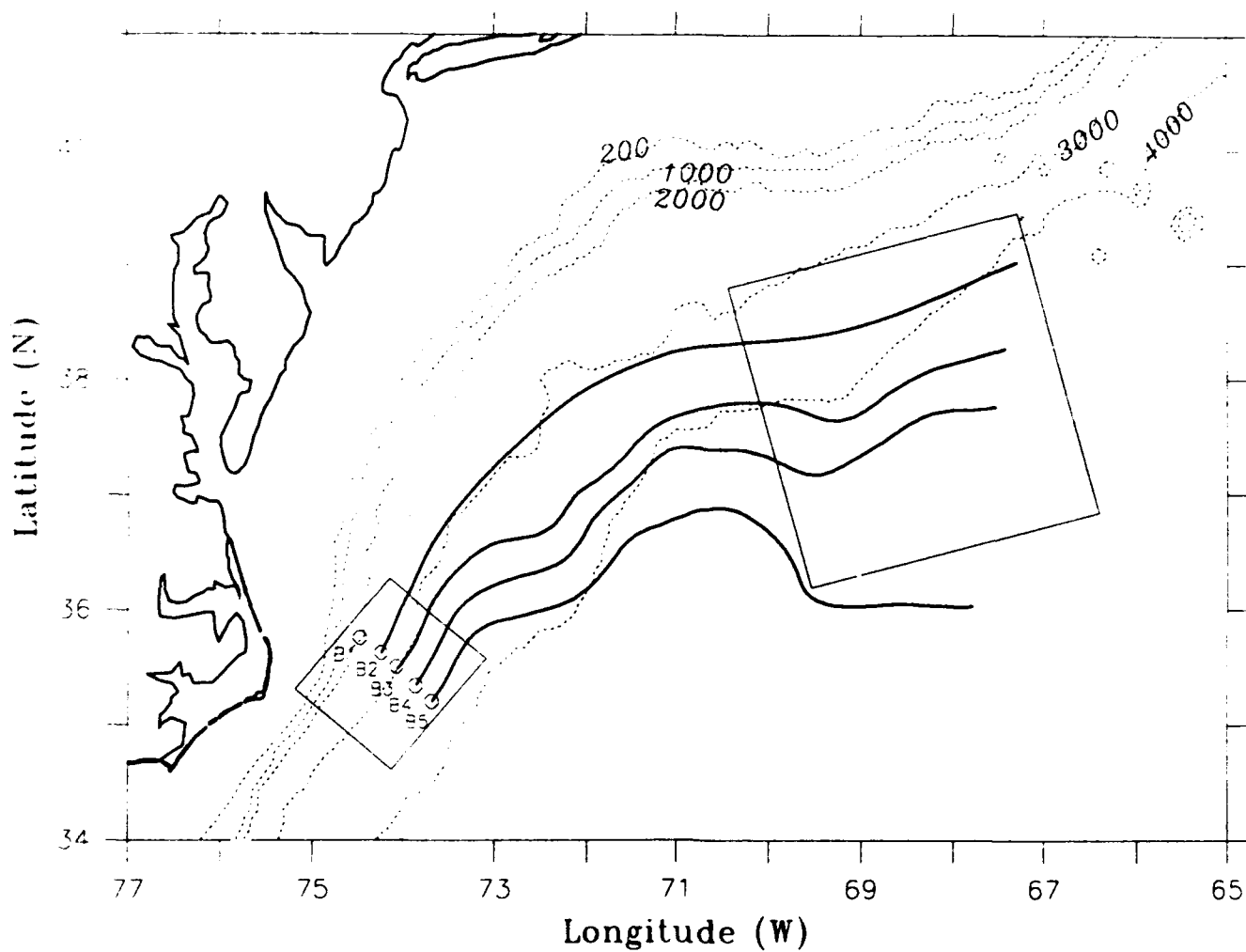


Figure 1: Location of SYNOP Inlet and Central Array. The circles denote the sites B1-B5. The coastline is shown by a dark line. The other dark lines are four TRW traces. The trace leaving B2 was run using splines fit to a 330 km filter, the others were run using splines fit to a 110 km filter. The B2, B3, B4, and B5 runs are 14, 22, 33, and 33 days in length, respectively.

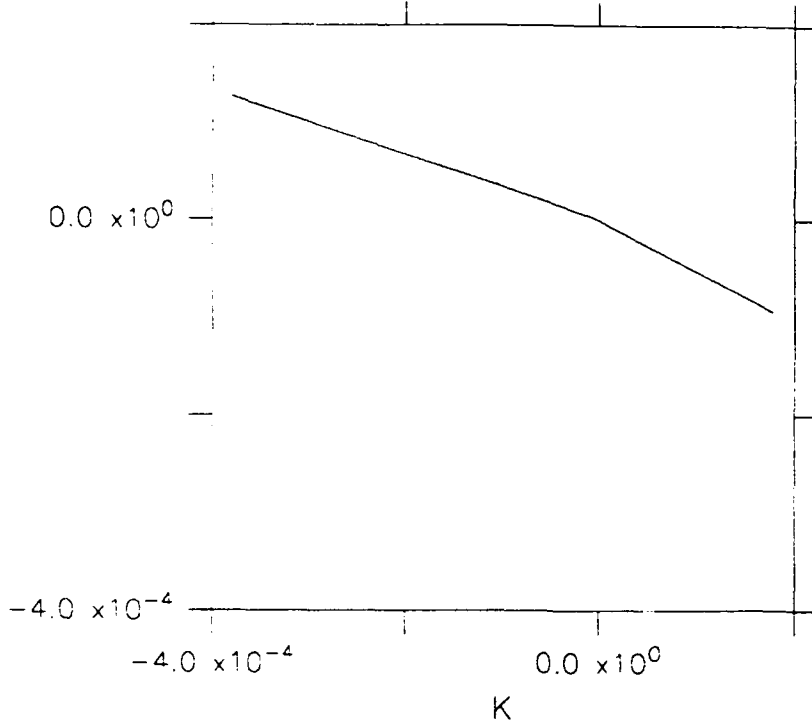


Figure 2: The bottom-trapped dispersion relation for a TRW shown as a function of (k, l) space. Units are m^{-1} .

In the second limit, $\tanh(\lambda h) \rightarrow (\lambda h)$, (14) becomes

$$\lambda^2 h = \left(\frac{NB^2}{f_0 \omega} \right) (h_y k - h_x l)$$

When this is combined with (12) the result is

$$(k + E)^2 + (l + F)^2 = G^2$$

where

$$E = \frac{\beta}{2\omega} - \frac{fh_y}{2\omega h}$$

$$F = \frac{fh_x}{2\omega h}$$

$$G = \left(\left(\frac{\beta}{2\omega} - \frac{fh_y}{2\omega h} \right)^2 + \left(\frac{fh_x}{2\omega h} \right)^2 \right)^{\frac{1}{2}}$$

This is the equation of a circle. The solution for this barotropic case at site B3, assuming the Test B3 values, is shown in Figure 3.

In the third limit, $\beta = 0$, (12) becomes

$$\lambda^2 = \left(\frac{N_B}{f_0} \right)^2 (k^2 + l^2)$$

which, when combined with (14) does not yield a simple analytic solution. Thus this limit is solved using numerical techniques together with the Test B3 values. Figure 4 shows the resulting dispersion relation.

In reality, TRWs should fall somewhere between the first two limits. Figure 5 shows a numerical solution for the dispersion curve using the Test B3 values but without specifying any limits. A comparison of Figures 2 and 5 reveals that the bottom-trapped case, where $\tanh(\lambda h) \rightarrow 1$, is an excellent approximation for the full solution (at least for waves specified by the Test B3 parameters).

Finally, Figure 6 shows the dependence of the full dispersion relation on frequency. Numerical solutions for four different wave periods are shown, 24, 32, 40, and 48 days. All four of these wave vectors essentially point either upslope or downslope; the best alignments were found with the longest period waves. As seen in Figure 6, for wavelengths greater than 75 km ($k < .0001$), k and l vary only slightly for these moderate changes in frequency.

4 Ray Tracing

The following discussion was adapted from LeBlond and Mysak [1978]. There are three steps involved in ray tracing. First, the equations describing the system are developed. This includes the expressions for the dispersion relation, the group velocity, the time variation of the frequency, and the dependence of the frequency on the environmental parameters (such as bottom slope and water depth). Second, initial values for position, wavelength, and frequency are chosen, and the corresponding group velocity, c_g , is computed. Third, the wave packet is progressed with velocity c_g for a time δt , and a new wavelength and frequency are found by integrating the equations from

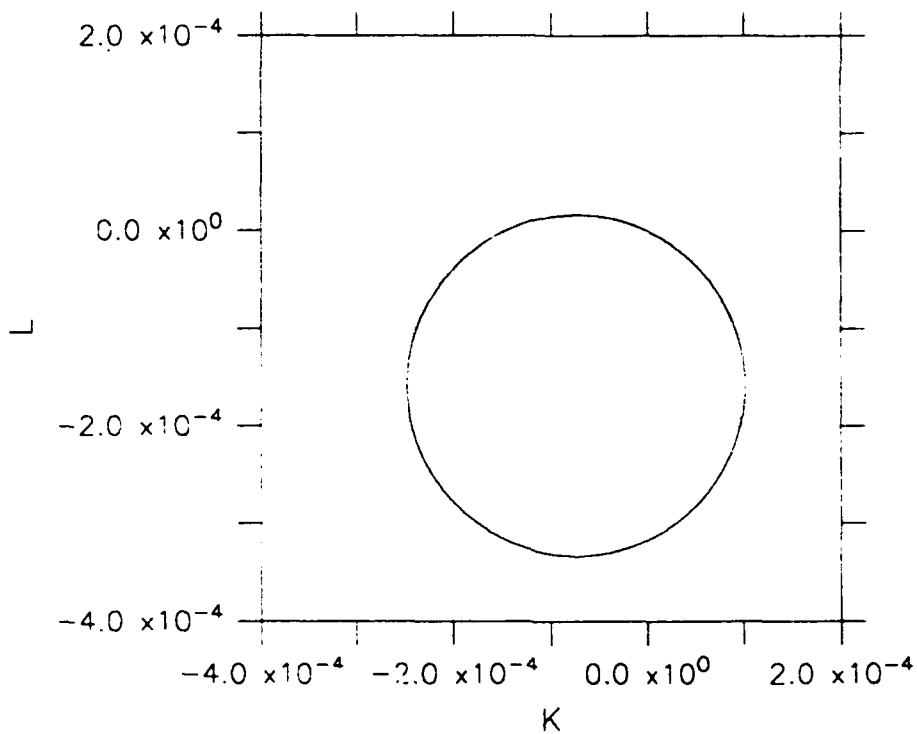


Figure 3: The uniform, barotropic dispersion relation for a TRW assuming the Test B3 values. Units are m^{-1} .

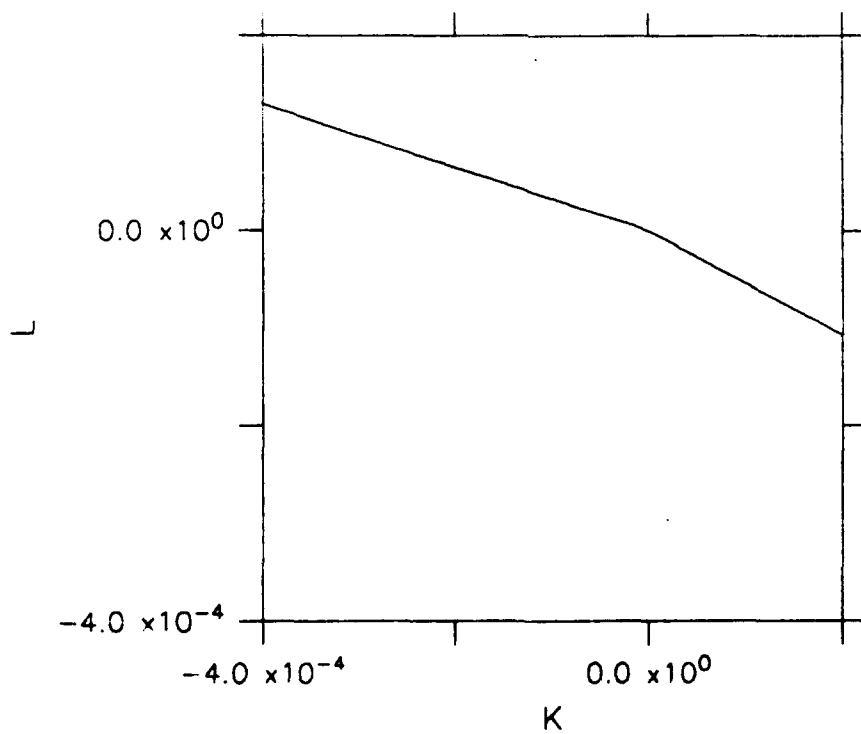


Figure 4: Numerical Solution for the $\beta = 0$ dispersion relation in which the Test B3 values are used. Units are m^{-1} .

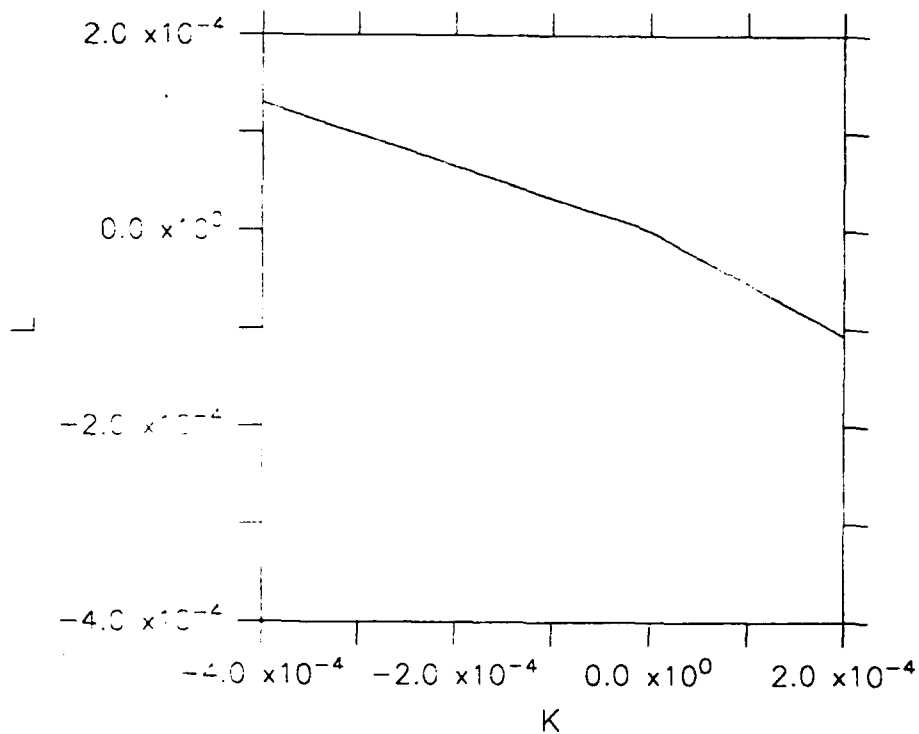


Figure 5: Numerical solution to the full dispersion relation. Units are m^{-1} .

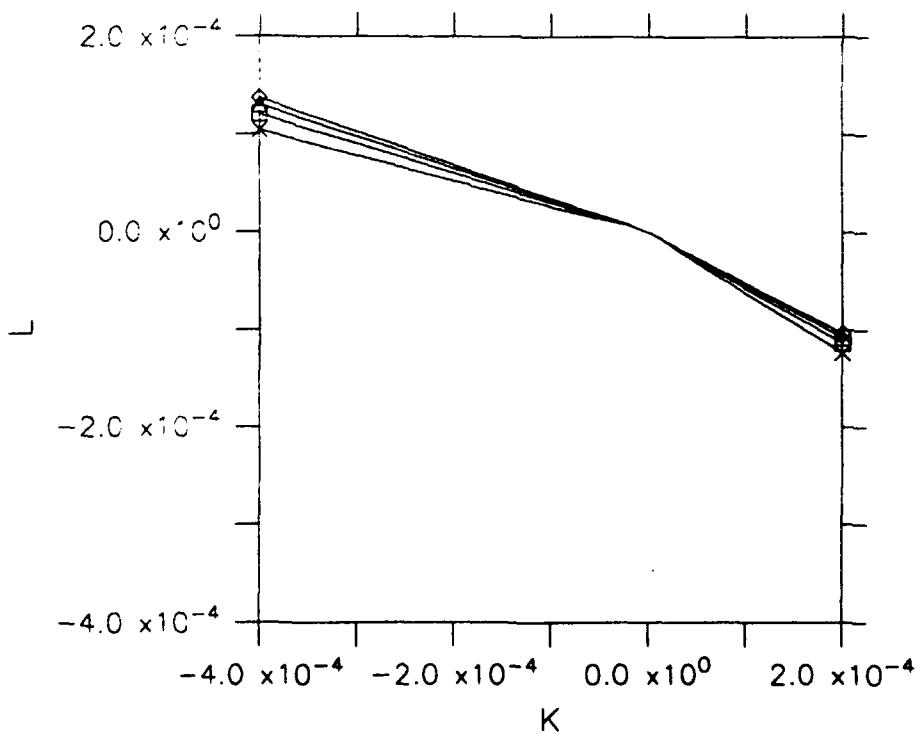


Figure 6: Numerical solutions for periods 24, 32, 40, and 48 days. The crosses denote a period of 24 days, the square is a 32 day period, the triangle denotes the 40 day period, and the diamond is the 48 day period. Units are in m^{-1}

step 1 over the time interval δt . The second and third steps are repeated to obtain a path.

To form a mathematical expression for the ray tracing consider a plane wave, with amplitude $A(\mathbf{x}, t)$ and phase $S(\mathbf{x}, t)$,

$$\phi(\mathbf{x}, t) = A(\mathbf{x}, t) \exp[iS(\mathbf{x}, t)] \quad (19)$$

It is assumed that the amplitude of the wave varies at a much slower rate and over much larger spatial scales than the phase. The local wavenumber $\mathbf{k}$ is defined as

$$\mathbf{k} = \nabla S$$

and the frequency ω as

$$\omega = -\frac{\partial S}{\partial t}$$

Cross-differentiation of these two definitions results in the conservation of wave crests equation

$$\frac{\partial \mathbf{k}}{\partial t} + \nabla \omega = 0 \quad (20)$$

Here $\mathbf{k}$ is the directional density of wave crests and ω is the flux of wave crests past a fixed point.

By considering both the dynamics of the wave type and the environmental parameters, it is possible to relate $\mathbf{k}$ and ω by using the dispersion relation

$$\omega(\mathbf{x}, t) = \sigma[\mathbf{k}(\mathbf{x}, t); \gamma(\mathbf{x}, t)] \quad (21)$$

where γ represents the environmental parameters. If the dispersion relation (21) is substituted into the conservation of wave crests (20), the result in tensor notation is

$$\frac{\partial k_i}{\partial t} + \frac{\partial \sigma}{\partial k_j} \frac{\partial k_j}{\partial x_i} + \frac{\partial \sigma}{\partial \gamma} \frac{\partial \gamma}{\partial x_i} = 0$$

Since the wavenumber is defined as the gradient of the phase, its curl must vanish; thus $\frac{\partial k_i}{\partial x_j} \epsilon_{ijk} = 0$

and so $\frac{\partial k_i}{\partial x_i} = \frac{\partial k_j}{\partial x_j}$. Using this and the definition of the group velocity, $c_{g_i} = \frac{\partial \sigma}{\partial k_i}$, the above equation

may be rewritten in vector notation as

$$\frac{d\mathbf{k}}{dt} \equiv \frac{\partial \mathbf{k}}{\partial t} + \mathbf{c}_g \cdot \nabla \mathbf{k} = -\frac{\partial \sigma}{\partial \gamma} \nabla \gamma \quad (22)$$

where $\frac{d}{dt} \equiv \frac{\partial}{\partial t} + \mathbf{c}_g \cdot \nabla$ is a derivative *moving with the wavepacket*. An expression for the variation of ω can be found by differentiating the dispersion relation (21) with respect to time and using the conservation of wave crests (20) to eliminate $\mathbf{k}$. The result is

$$\frac{\partial \omega}{\partial t} = \frac{\partial \sigma}{\partial k_i} \frac{\partial k_i}{\partial t} + \frac{\partial \sigma}{\partial \gamma} \frac{\partial \gamma}{\partial t} = -\frac{\partial \sigma}{\partial k_i} \frac{\partial \omega}{\partial x_i} + \frac{\partial \sigma}{\partial \gamma} \frac{\partial \gamma}{\partial t}$$

Rewriting in vector notation similar to (22) gives the variation in ω along a ray path

$$\frac{d\omega}{dt} = \frac{\partial \omega}{\partial t} + \mathbf{c}_g \cdot \nabla \omega = \frac{\partial \sigma}{\partial \gamma} \frac{\partial \gamma}{\partial t} \quad (23)$$

Thus the equations which describe the system are

$$\omega(\mathbf{x}, t) = \sigma[\mathbf{k}(\mathbf{x}, t); \gamma(\mathbf{x}, t)] \quad (24)$$

$$\frac{d\mathbf{x}}{dt} = \mathbf{c}_g = \frac{\partial \sigma}{\partial \mathbf{k}} \quad (25)$$

$$\frac{d\mathbf{k}}{dt} = -\frac{\partial \sigma}{\partial \gamma} \nabla \gamma \quad (26)$$

$$\frac{d\omega}{dt} = \frac{\partial \sigma}{\partial \gamma} \frac{\partial \gamma}{\partial t} \quad (27)$$

5 Wave Tracing Programs

This section describes the processing steps for ray tracing. The MATLAB code (or FORTRAN, in the case of the first program **GET_ETOPO_II.FOR**) are given in Appendix C.

5.1 Preparing the Topography

One of the most important features of this method is that real bottom topography is used. In order to remove all small scale topographic fluctuations that would not affect the waves but will influence the resulting spline fit, the topography must be gridded and smoothed. Figures showing the progressive results of this process are presented in Appendix B.

5.1.1 The Raw Bathymetry

The raw bathymetry can be obtained from the NCAR ETOPO5 earth elevation data set which is a gridded topography array on a $\frac{1}{12}^\circ$ grid. The data set is gridded in equal increments of latitude and longitude rather than kilometers so the gridding in the x and y directions is slightly different. Since the area of study will normally encompass only a small fraction of this data, a program called `GET_ETOPO_II.FOR` is used to select just the bathymetry for the area of interest. It is best to choose the area of interest generously, so that boundary effects of the smoothing filter (discussed later) will be minimized in the actual study area.

The program `GET_ETOPO_II.FOR` is based on a program `GET_ETOPO.FOR` (written by John Lillibridge at URI). The program has been modified to remove the z -scaling option and to run on a micro VAX computer. The program is executed by the command file `GET_ETOPO.COM`, which must be modified to contain the appropriate input and output file names.

5.1.2 Remove unwanted features

After windowing the bathymetry, it is necessary to remove some isolated features. In this formulation only waves in regions where WKBJ approximations are valid (i.e. where the topography varies on a much longer scale than the wavelength). Thus features that violate this must be removed or the subsequent smoothing process will spread those features to the region of interest. For example, bathymetric features that are narrow compared to a wavelength would approximately generate Taylor-columns to the waves; thus it is better to eliminate these narrow features. The larger features, such as isolated large-diameter seamounts or chains of seamounts, act as scatterers and must be removed since reflection is not included in the formulation. The problem features are removed by reducing their vertical extent. For example, a seamount that was on a slope of average depth -3500 meters and rose to -2800 meters would cause problems so all grid points in the region

of the mount that had a value of greater than -3400 meters would be set equal to -3400 meters, effectively shortening the seamount.

The MATLAB routine **LANDSCAPE.M** is used to eliminate unwanted features. A mouse is required to run the program. The mouse is used to focus in on the feature to be shortened. A seamount is selected and eliminated by clicking with the mouse, first on the lower left and then on the upper right corner of the seamount. At the 'landscape level?' query give the desired depth of the seamount (e.g. -3500). The program will continue to loop through these steps as long as the user types 'y' to the 'continue' prompt. The bathymetric contours may be selectively labelled. This is accomplished by clicking with the mouse on the desired contour at the 'manual' prompt. After all features have been edited, answer 'n' to the 'continue' query. Save the variable 'mowed_BATH' to a disk file; this is the new bathymetry data set.

5.1.3 Filtering

After removing the unwanted large-scale features, it is necessary to filter out the small-scale features that in reality do not affect the waves, but in this formulation would cause considerable noise in the calculation of the environmental parameters γ and $\nabla\gamma$. The TRWs of interest have wavelengths ranging from about 110 km to 330 km. Therefore a cutoff wavenumber of $\frac{2\pi}{110\text{km}}$ was chosen for the wavenumber filter. MATLAB computes the filter coefficients from the ratio of the cutoff wavenumber to the Nyquist wavenumber.

The program for filtering the topography is a MATLAB routine called **FILTER2.M**. To run the program, the user supplies the coefficients for the spatial gridding and the 'mowed_BATH' data set. The smoothed bathymetry, Y , should be saved in a disk file for later use.

5.1.4 Compute spline representation

The final step in preparing the topography is to fit a series of B-splines to the bottom depths. This is done so that the gradient of the depth, ∇h , can be computed for any given location. Because B-splines are linear processes, the two dimensional grid of bathymetry can be treated one dimension at a time. The B-splines are first fit to lines of constant latitude. Then another set of B-splines is fit as a function of latitude (i.e. along lines of constant longitude), creating a two dimensional B-spline expression for the bathymetry.

The program that fits the B-splines is called **COMPUTE_SPLINES3.M**. The code is commented heavily. Many of the commands are concerned with reducing "undulations" at the edges of the domain. These are reduced by augmenting the series to be interpolated. The arguments of the program are the latitude and longitude of the first point of the filtered bathymetry data set. It should be noted that the constant 'dlat' must be specified prior to executing the program. This constant specifies the grid spacing of the input array of bathymetry. (For example, for a grid spacing of half a degree, set *dlat* = 0.5). The *output* variables 'c, cprime, c2prime, and xknots' should be saved in a disk file. (Use an *output* file name such as **SPLINE_110** to indicate what filter was applied to the *input* bathymetry data.)

The routine **TEST_SPLINES** uses the B-splines variables to solve for the depth, bottom slope, and their gradients at a given location. It is executed in each ray tracing step. In addition to the B-spline coefficients, the latitude and longitude of the point must be supplied.

5.2 Getting the initial wavenumber

As noted previously in step 2 of the Ray Tracing discussion, it is necessary to obtain initial (k_i, l_i) estimates for the wavenumber vector. These values are needed in order to compute the initial group velocity of the wave packet. The MATLAB routine **INITVAL.M** requires the user to supply the

position of the starting point of the TRW, β , f_0 , N_B , H , and ∇h , as well as the wavelength and frequency of the wave. The outputs are k and l as well as the bottom trapping coefficient λ and the propagation angle θ .

Prior to running INITVAL.M, a five element row vector called 'parms' must be created. The five elements of this vector are β , dt , f_0 , H , and N_B . The second element, the time step dt , is not used by this program so it is not crucial to supply its correct value. (However, dt will be used in a subsequent program.) The third and fourth elements are the Coriolis parameter and the average water depth from sea surface to sea floor. The final element is N_B , the Brunt-Vaisälä frequency.

The routine INITVAL.M also requires the depth, h , at the starting point of the TRW as well as the bottom slopes h_x and h_y . To find the value of h , h_x , and h_y at this position the user must run the MATLAB program TEST_SPLINES.M using the B-spline coefficients obtained from the COMPUTE_SPLINES3.M program together with latitude and longitude of that point. The output of the TEST_SPLINES.M program should be placed in variable called 'gradh' for use by the program INITVAL.M to find the wavenumber.

Four additional inputs are required before executing the INITVAL.M program. Set 'omega' to be the angular frequency, ($\frac{2\pi}{T}$), of the TRW in s^{-1} and set 'LAM' to be the wavelength in km. The longitude and latitude of the starting point are specified as 'x' and 'y.' Once all of the variables necessary to run INITVAL.M are specified, the program may be executed. The initial k and l values, as well as the bottom-trapping coefficient λ and the propagation angle θ (measured from east) of the TRW are calculated by the program.

5.3 Tracing

The MATLAB routine called TRWTRACE.M traces the wave path in the manner discussed in the section entitled Ray Tracing. The routine contains a self check to see if the error is becoming

too large. For this test, a new frequency, ω_i , is computed using the new wavenumbers. This is then checked against the initial frequency by $wcheck = \frac{\omega_i - \omega_0}{\omega_0}$. Ideally the frequency should not vary over time. However due to computational error the frequency does change and this test warns if the error is becoming too large. Tests (discussed later) show that for the 40-day period waves, errors in $wcheck$ of as much as 25% will not significantly alter the important wave parameters, including the path, wavenumber, group speed, or the bottom-trapping coefficient.

There are several variables that need to be assigned prior to running **TRWTRACE.M**. First, the time step increment dt (the second element in the 'parms' array) must be set. This program allows the wave to be traced forward (positive time increment) or backward (negative time increment) in time. The time increment is specified in hours. Second, put the longitude and latitude, in that order, into a 2-element row vector called 'loc' where longitude is measured in degrees east (e.g. 73W is -73). Third another row vector called 'k' should contain the initial k_x and k_y values. Finally, the user must specify the name of the file containing the B-spline variables (e.g. **SPLINE.110**).

This is an approximate guide to the execution time of the **TRWTRACE.M** program on a DEC 3100-76 computer. The major time-consuming step is in the calculation of ∇h which requires the computation of new B-spline variables for each new position. The execution time for this step is lengthened greatly if a dense spline set is used. For example, if the B-splines have been fit to a 15 by 25 grid then to run **TRWTRACE.M** for 50 time steps, the total run time is approximately 15 minutes. Denser filter grids significantly slow down the program; for example using splines fit to a 85 by 145 grid, a 50 time step run would take 1.5 hours.

5.4 Examples

Choosing the ray tracing parameters, such as filter size and time step, can be difficult. One recommendation is to conduct test runs, specifying different choices for each parameter and compare

the results. That way the time efficiency can be weighed against the desired accuracy. The following examples (Figures 6-10) show the effects of various choices of these parameters on tracing TRW ray paths, from a detection site in the SYNOP Inlet Array to a potential generation region in the SYNOP Central Array. (Thus here the time steps will be negative).

For the following cases a consistent set of parameters was used and only one parameter was varied at a time. Unless stated otherwise, the bathymetry wavenumber filter size was 330 km, the time step was eight hours, the period was 40 days, the wavelength was 130 km, and the launch site was B3. TRWs of this wavelength and period have been observed in the Inlet Array (Pickart and Watts, 1990). Several parameters were also specified; $\beta = 1.8 \times 10^{-11} \text{ m}^{-1}\text{s}^{-1}$, $f = 9 \times 10^{-5} \text{ s}^{-1}$, and $H = 4000 \text{ m}$. As mentioned earlier, N_B was set equal to a constant value, an appropriate value for the Gulf Stream region being $N_B = 1 \times 10^{-3} \text{ s}^{-1}$ (Pickart, personal communication).

Figure 7, shows the effect of varying the time step (dt) on the wavepath. Time steps of one hour and eight hours were tested. The paths for these two runs are almost indistinguishable; the only difference being that the total wave distance travelled was about 2% shorter for the $dt = -1$ hr case due to a slower group speed. Figure 8 compares how several of the parameters varied over time for these two runs. The plots of the total wavenumber (K), the average bottom slope ($\text{grad}h$), and the bottom trapping coefficient (λ) show little difference between the two runs. However, there is a very large difference in 'wcheck', a measure of the percent variation of the actual frequency in the run compared to the correct frequency. A value of zero for wcheck indicates no error in computed frequency. In Figure 8 it is obvious that the frequency errors are high when $dt = -8$ hours; however this appears to have no effect on the predicted wave path (Figure 6). On the other hand differences in the actual frequency result in differences in the group speed C_g for the two test cases. Figure 8 shows that when $dt = -1$ hr the TRW travels about 2 - 3% slower than when $dt = -8$ hrs.

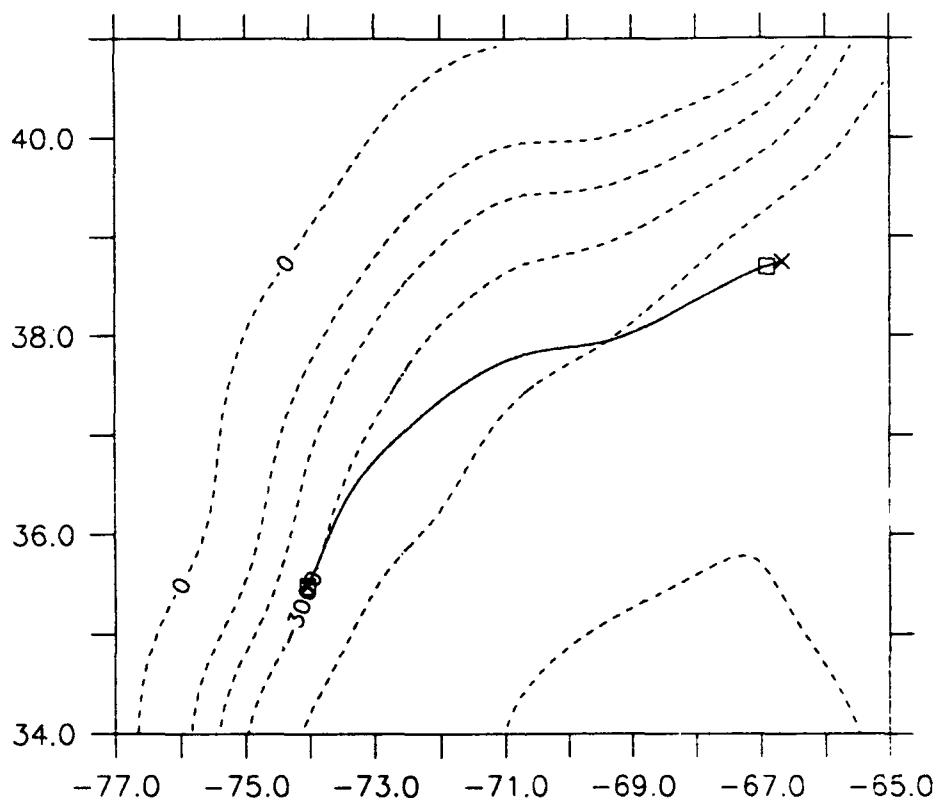


Figure 7: B3 launch with $dt=-1$ hrs and $dt=-8$ hrs for 410 time steps and 52 time steps, respectively. Approximate run times 1.5 hours and 15 minutes, respectively. The cross denotes the eight hour time step and the square is the one hour time step run. The dashed lines are the 330 km filtered bathymetry.

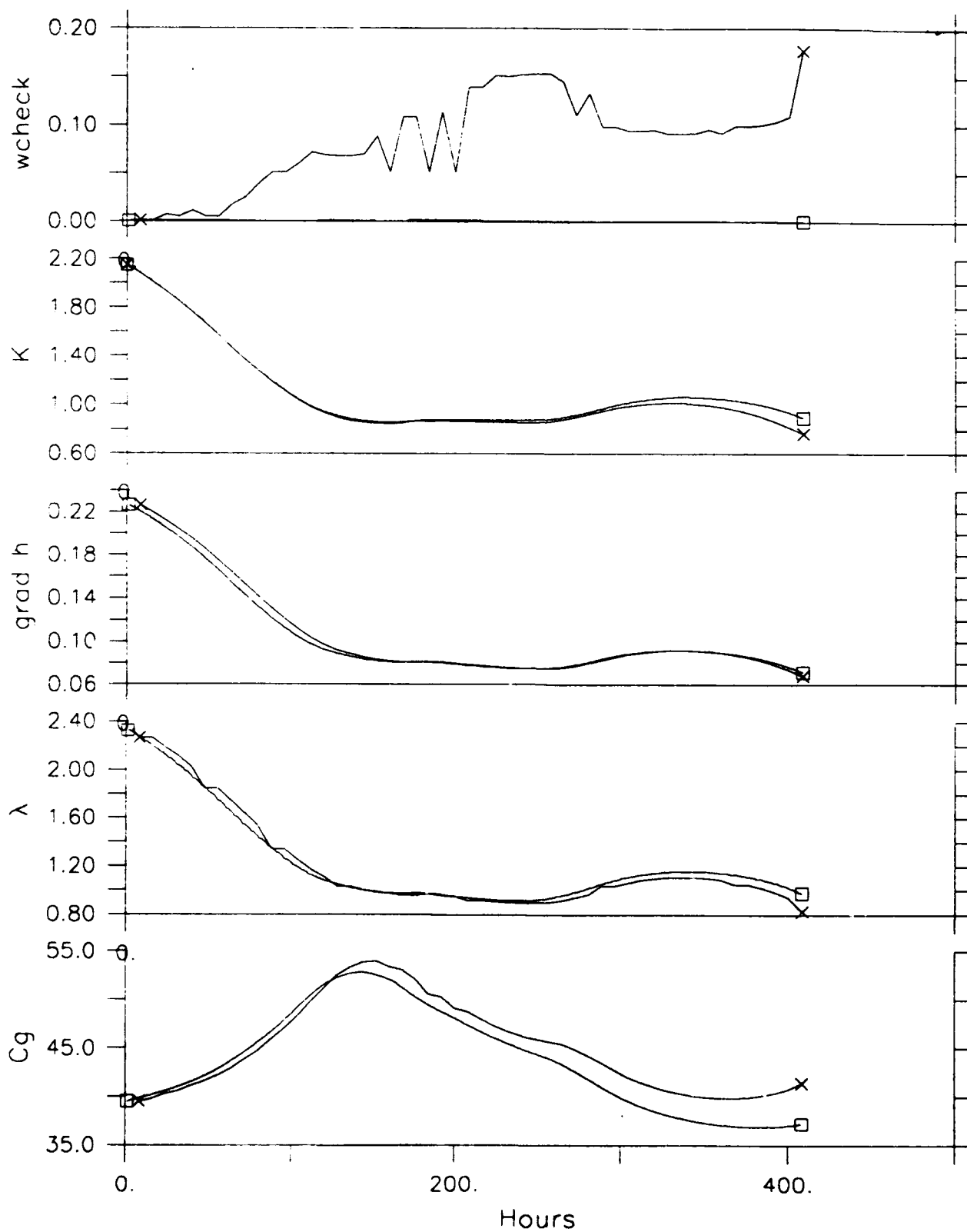


Figure 8: Comparison of $dt = -1$ hrs and $dt = -8$ hrs. The cross denotes the eight hour time step run and the squares are the one hour run.

The effects of varying the degree of smoothing of the bathymetry were also tested. For example, for the area between the SYNOP Inlet and Central Arrays, the 110 km filter gives a more realistic bottom topography than the 330 km filter (see Appendix B). However, in order to fit B-splines to the 110 km filtered bathymetry without aliasing, the filtered data can only be subsampled every $\frac{1}{6}$ degree. This leads to a very dense spline set and considerably slows down the tracing program. Using a 330 km filter for the bathymetry permits subsampling at every $\frac{1}{2}$ degree without aliasing, and in turn leads to a sparser spline set and a shorter run time. Figure 9 shows the predicted wave paths from site B3 for three different filter sizes, 110, 220, and 330 km. Despite the differences in the smoothed bathymetries the end positions of the predicted wave paths are quite close.

The influence of the choice of launch position and wave period was also tested. Figure 10 shows launches from sites B2, B3, and B4 of the Inlet Array, which are about 50 km from each other. All other parameters have been held constant. The three paths appear to parallel one another. The biggest difference between them is the total distance traveled, implying that the group speeds are different. The slowest speeds were obtained for the wave traced from site B4, where $|\nabla h|$ is the smallest.

The predicted wave path also depends on the wave periodicity. Figure 11 shows the traces of three waves launched from site B3 with periods of 32, 40, and 48 days. In this case the waves travel approximately the same distance (There is little variation in longitude of the end points). On the other hand there is a difference about 2° latitude between the trace endpoints for waves with the 32 day periodicity and the 48 day period waves.

These results can be summarized as follows. Firstly, an accuracy of 10 kilometers in the 'launch' site of the wave gives consistent path predictions. Additionally, a 10% error in wave period will also give consistent path predictions. Second, more accurate topography is obtained when a 110 km wavenumber filter is applied. Finally, a time step of $dt = 8$ hours is adequate for most cases;

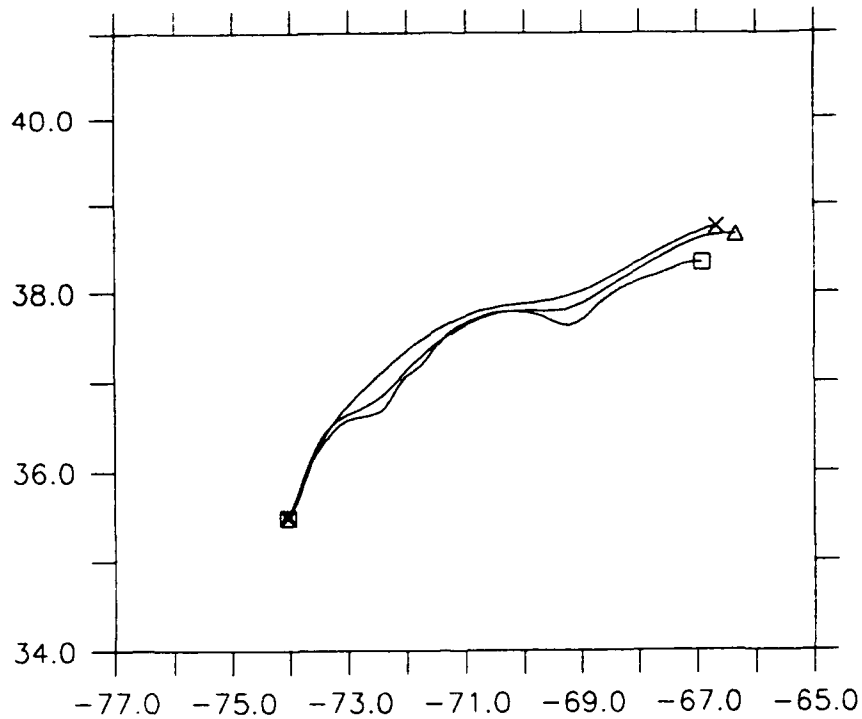


Figure 9: B3 launch with splines on 110, 220, and 330 km filters for 70 time steps of $dt=-8$ hrs on the 110 filter and 52 time steps of $dt=-8$ hrs on the other two filters. Approximate run times 1.5 hrs., 45 min., and 15 min. respectively. The cross denotes the 330 km filter run, the triangle is the 220 km filter run, and the square is the 110 km filter run.

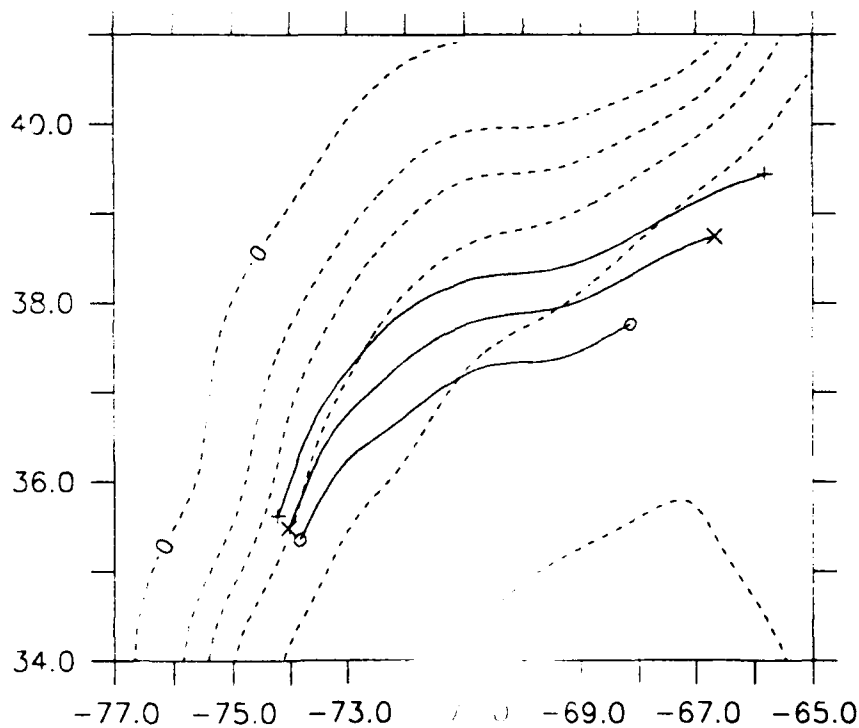


Figure 10: Launches from sites B2, B3, and B4, each for 52 time steps of $dt=-8$ hrs. Approximate run time 15 minutes for all. The plus denotes the B2 launch, the cross is the B3 launch, and the circle shows the B4 launch.

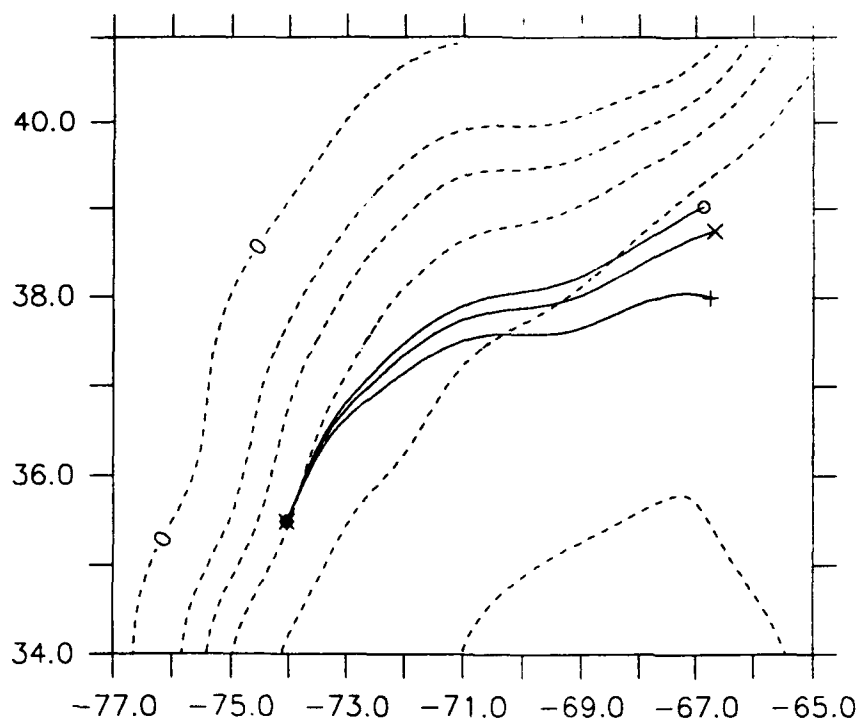


Figure 11: B3 launch with different periods, each for 52 time steps of $dt = -8$ hrs. The plus denotes the 32 day period, the cross is the 40 day period, and the circle shows the 48 day period.

using $dt = -1$ hour is only necessary when extreme accuracy in group speed, c_g , is needed.

6 References

Hogg, N.G. 1981. Topographic waves along 70°W on the Continental Rise. *J. Mar. Res.*, 39, 627-649.

Johns, W.E. and D.R. Watts. 1985. Gulf Stream meanders: observations on the deep currents. *J. Geophys. Res.*, 90, 4819-4832.

LeBlond, P.H. and L.A. Mysak. 1978. Waves in the Ocean. Elsevier, Amsterdam.

Pickart, R.S. and D.R. Watts. 1990. Deep Western Boundary Current variability at Cape Hatteras. *J. Mar. Res.*, 48, 765-791.

Schultz, J.R. 1987. Structure and propagation of topographic Rossby waves northeast of Cape Hatteras. M.S. Thesis, Marine Sciences Program, University of North Carolina, Chapel Hill, NC. 63pp.

7 Acknowledgements

The authors would like to thank Nelson Hogg for sharing an earlier version of his MATLAB ray tracing program and also for his considerable help in adapting this code for our purposes. Also thanks to Karen Tracey for her assistance in the preparation of the this report. The work was supported by ONR contracts N00014-90J-1568 and N00014-92J-4013 and by NSF grant number OCE87-17144.

8 Appendix A The Dispersion relation derivation

This is the derivation of the dispersion relation (equations (15) and (16)) that was shown in the Dynamics and Kinematics of TRWs section. Begin by using the same five equations, the two linearized horizontal motion equations, the incompressibility equation, the hydrostatic equation, and the linearized density equation, which are repeated here.

$$\frac{\partial u}{\partial t} - fv = -\frac{1}{\rho_0} \frac{\partial p'}{\partial x} \quad (28)$$

$$\frac{\partial v}{\partial t} + fu = -\frac{1}{\rho_0} \frac{\partial p'}{\partial y} \quad (29)$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \quad (30)$$

$$-\frac{\partial p'}{\partial z} - \rho'g = 0 \quad (31)$$

$$\frac{\partial \rho'}{\partial t} - \frac{\rho_0 N^2 w}{g} = 0 \quad (32)$$

Now, taking the time derivative of (31) and substituting (32) into it yields

$$\frac{\partial^2 p'}{\partial t \partial z} + \rho_0 N^2 w = 0 \quad (33)$$

Next, take $\frac{\partial}{\partial x}$ of (29) and subtract $\frac{\partial}{\partial y}$ of (28) to give

$$\frac{\partial^2 v}{\partial x \partial t} - \frac{\partial^2 u}{\partial y \partial t} + f \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) + \beta v = 0$$

Substituting (30) for the quantity in paranthesis and moving it over to the right hand side gives

$$\frac{\partial}{\partial t} \left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \right) + \beta v = f \frac{\partial w}{\partial z} \quad (34)$$

which is the linear vorticity equation. Let the quantity in paranthesis be denoted as ζ . Now, by using the leading order geostrophic balance it is possible to get relations for v_g , u_g , and ζ_g

$$v_g = \frac{1}{f_0 \rho_0} \frac{\partial p'}{\partial x} \quad (35)$$

$$u_g = -\frac{1}{f_0 \rho_0} \frac{\partial p'}{\partial y} \quad (36)$$

$$\zeta_g = \frac{1}{f_0 \rho_0} \nabla^2 p' \quad (37)$$

Here $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$. Substituting these into the linear vorticity (34) gives us the Quasi-Geostrophic Vorticity (QGV) Equation

$$\frac{1}{f_0 \rho_0} \left[\frac{\partial}{\partial t} (\nabla^2 p') + \beta \frac{\partial p'}{\partial x} \right] = f_0 \frac{\partial w}{\partial z} \quad (38)$$

Now take $\frac{\partial}{\partial z}$ of (33) to get

$$\frac{\partial}{\partial t} \frac{\partial}{\partial z} \left(\frac{1}{\rho_0 N^2} \frac{\partial p'}{\partial z} \right) + \frac{\partial w}{\partial z} = 0$$

which is substituted into the right hand side of the QGV (38) to give

$$\frac{1}{f_0 \rho_0} \left[\frac{\partial}{\partial t} (\nabla^2 p') + \beta \frac{\partial p'}{\partial x} \right] = -f_0 \frac{\partial}{\partial t} \frac{\partial}{\partial z} \left(\frac{1}{\rho_0 N^2} \frac{\partial p'}{\partial z} \right)$$

Multiplying both sides by ρ_0 and combining the time derivatives gives

$$\frac{\partial}{\partial t} [\nabla^2 p' + \frac{\partial}{\partial z} \left[\left(\frac{f_0}{N} \right)^2 \frac{\partial p'}{\partial z} \right]] + \beta \frac{\partial p'}{\partial x} = 0$$

Using Hogg's [1981] technique approximate the varying N with a constant Brunt-Vaisälä frequency, N_B , at the bottom to obtain a workable equation of one variable.

$$\frac{\partial}{\partial t} [\nabla^2 p' + \left(\frac{f_0}{N_B} \right)^2 \frac{\partial^2 p'}{\partial z^2}] + \beta \frac{\partial p'}{\partial x} = 0 \quad (39)$$

This is the same expression as equation (6). Now for the top boundary condition at $z = 0$ the rigid lid approximation is used to force $w = 0$. So (33) simplifies to

$$\frac{\partial^2 p'}{\partial t \partial z} = 0 \quad (40)$$

Assuming no normal flow at the bottom, where $w = \mathbf{u} \cdot -\nabla \mathbf{h}$

$$w = \frac{1}{f_0 \rho_0} \left[\frac{\partial p'}{\partial y} \frac{\partial h}{\partial x} - \frac{\partial p'}{\partial x} \frac{\partial h}{\partial y} \right]$$

Substituting into (33) gives

$$\frac{\partial^2 p'}{\partial t \partial z} = \frac{N^2}{f_0} \left[\frac{\partial p'}{\partial x} \frac{\partial h}{\partial y} - \frac{\partial p'}{\partial y} \frac{\partial h}{\partial x} \right] \quad (41)$$

where $z = -h(x, y)$. A plane wave solution of the form

$$p' = A(z) \exp[i(kx + ly - \omega t)] \quad (42)$$

is sought. Substituting (42) into (39) gives

$$\frac{\partial}{\partial t} [p'(-k^2 - l^2 + \left(\frac{f_0}{N_B}\right)^2 \frac{\partial^2 A}{\partial z^2} \frac{1}{A})] + \beta i k p' = 0$$

From this point, the prime will be dropped from pressure p , and the subscripts will be dropped from f and N . Taking the time derivative yields

$$i\omega p(k^2 + l^2 - \left(\frac{f}{N}\right)^2 \frac{1}{A} \frac{\partial^2 A}{\partial z^2}) + \beta i k p = 0$$

Divide by both i and p , since they are non-zero, leaving

$$\omega(k^2 + l^2 - \left(\frac{f}{N}\right)^2 \frac{1}{A} \frac{\partial^2 A}{\partial z^2}) + \beta k = 0$$

Dividing by ω and regrouping gives

$$-\left(\frac{f}{N}\right)^2 \frac{1}{A} \frac{\partial^2 A}{\partial z^2} + (k^2 + l^2 + \frac{\beta k}{\omega}) = 0$$

Multiplying by $-(\frac{N}{f})^2 A$ produces

$$\frac{\partial^2 A}{\partial z^2} - A \left[\left(\frac{N}{f}\right)^2 (k^2 + l^2 + \frac{\beta k}{\omega}) \right] = 0$$

If the quantity inside the square brackets is defined as λ^2 the result is

$$\frac{\partial^2 A}{\partial z^2} - \lambda^2 A = 0 \quad (43)$$

where

$$\lambda^2 = \left(\frac{N}{f}\right)^2 (k^2 + l^2 + \frac{\beta k}{\omega}) \quad (44)$$

Using (44) and the top boundary condition (40) it is possible to solve for $A(z)$

$$A(z) = \cosh(\lambda z) \quad (45)$$

Putting this into (42) and then substituting into the bottom boundary condition (41) the left side of (41) becomes

$$\frac{\partial^2 p}{\partial t \partial z} = -\lambda \sinh(-\lambda h) i \omega \exp[i(kx + ly - \omega t)]$$

and the right side of (41) becomes

$$\frac{N^2}{f} \left(\frac{\partial h}{\partial y} \frac{\partial p}{\partial x} - \frac{\partial h}{\partial x} \frac{\partial p}{\partial y} \right) = \frac{N^2}{f} i \cosh(-\lambda h) \exp[i(kx + ly - \omega t)] \left(\frac{\partial h}{\partial y} k - \frac{\partial h}{\partial x} l \right)$$

Setting the two sides equal and cancelling the common i and $\exp[i(kx + ly - \omega t)]$

$$-\lambda \sinh(-\lambda h) \omega = \frac{N^2}{f} \cosh(-\lambda h) (h_y k - h_x l)$$

where the derivatives of h are denoted as h_x and h_y . Dividing through by $\cosh(-\lambda h)$ gives

$$-\lambda \tanh(-\lambda h) = \frac{N^2}{\omega f} (h_y k - h_x l)$$

Since $\tanh(-x) = -\tanh(x)$, this can be written as

$$\lambda \tanh(\lambda h) = \frac{N^2}{\omega f} (h_y k - h_x l) \quad (46)$$

At this point, the equations are easier to work with if they are non-dimensionalized. Specify the following factors

$$\begin{aligned} h &= H h^* \\ \lambda &= \frac{\lambda^*}{H} \\ \omega &= \frac{\beta \omega^* N H}{f} \\ (k, l) &= \frac{f}{N H} (k^*, l^*) \end{aligned}$$

$$\left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right) = \frac{f}{NH} \left(\frac{\partial}{\partial x^*}, \frac{\partial}{\partial y^*}\right)$$

Where the asterisks denote non-dimensional variables. Substituting these into (44) gives the non-dimensional form

$$\frac{\lambda^{*2}}{H^2} = \left(\frac{N}{f}\right)^2 \left(\frac{k^{*2} f^2}{N^2 H^2} + \frac{l^{*2} f^2}{N^2 H^2} + \frac{\beta k^* f^2}{N^2 H^2 \beta \omega^*} \right)$$

This expression can be simplified by multiplying by H^2 , moving $\frac{N^2}{f^2}$ inside the parenthesis, and cancelling the β -terms to obtain

$$\lambda^{*2} = k^{*2} + l^{*2} + \frac{k^*}{\omega^*} \quad (47)$$

The non-dimensional form of (46) is

$$\frac{\lambda^*}{H} \tanh\left(\frac{\lambda^*}{H} H h^*\right) = \frac{N^2}{f} \frac{f}{\beta \omega^* N H} \frac{H f}{N H} \left(h_y^* \frac{k^* f}{N H} - h_x^* \frac{l^* f}{N H} \right)$$

Cancelling the common factors gives

$$\lambda^* \tanh(\lambda^* h^*) = \frac{f^2}{\beta H N} \frac{(h_y^* k^* - h_x^* l^*)}{\omega^*}$$

Finally, introducing the radius of deformation, $R_d = \frac{NH}{f}$ yields

$$\lambda^* \tanh(\lambda^* h^*) = \frac{f}{\beta R_d} \frac{(h_y^* k^* - h_x^* l^*)}{\omega^*} \quad (48)$$

Equations (47) and (48) together comprise the non-dimensional dispersion relation for this system.

9 Appendix B Example Bottom Topography Preparation.

Following are three figures that show the progression of a region of bottom topography as it is prepared for ray tracing. The area shown is 65-77 W and 34-41 N.

The first is the original bottom topography, as taken from the ETOPO5 data set. Figure 12 still has the sea mounts and all of the true topography shown. After the LANDSCAPE.M program is run the seamounts that were in the upper right corner are no longer there, creating a topography such as in Figure 13. After the filtering has been done the result is Figure 14. Here a 110 km filter has been used. Figure 15 shows the 220 km filtered topography. Figure 16 shows the 330 km filtered topography.

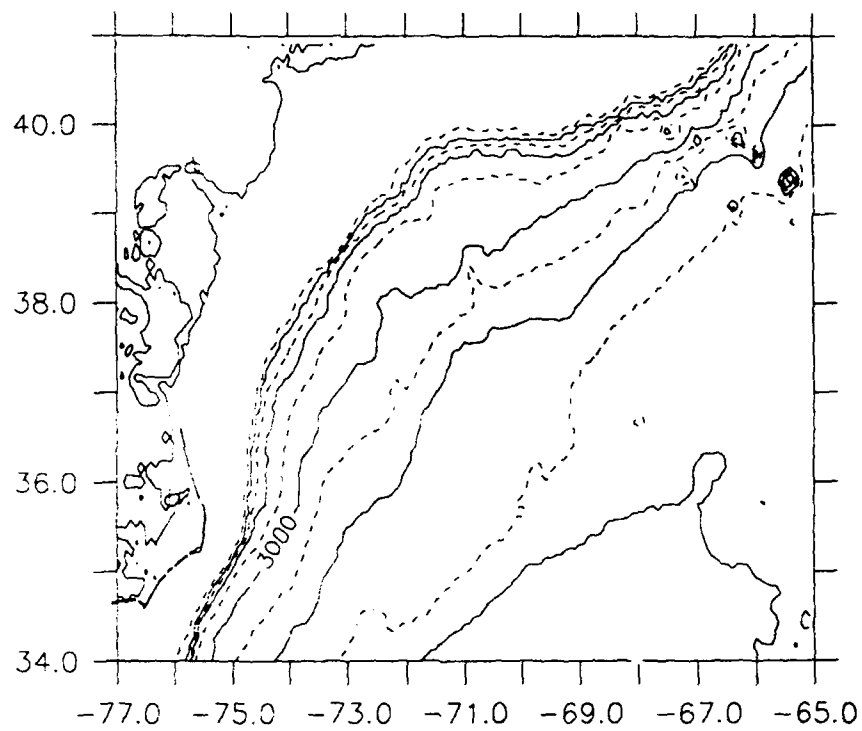


Figure 12: Real bottom topography

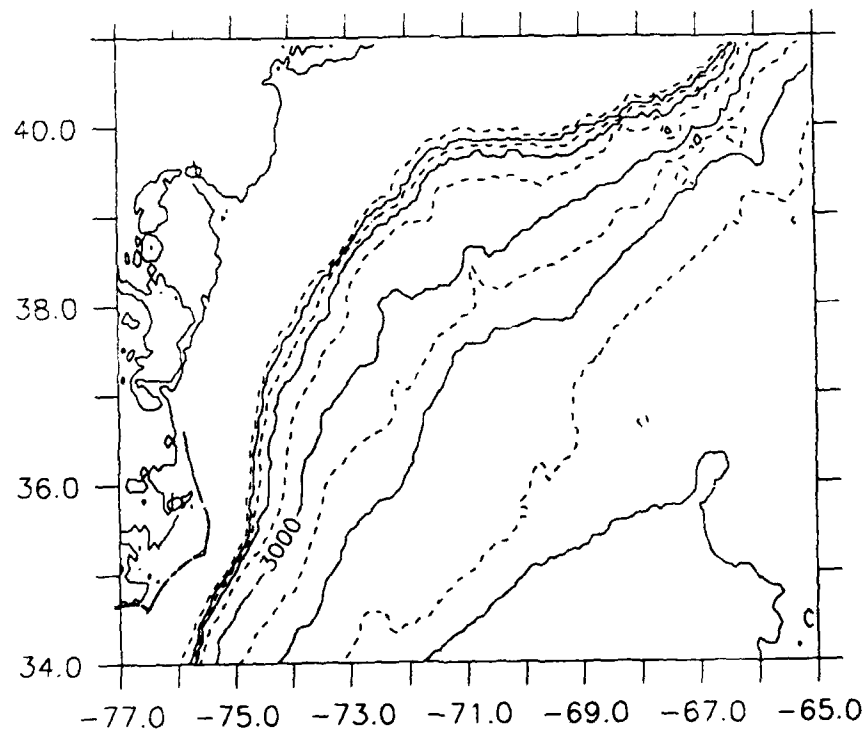


Figure 13: Topography with seamounts removed

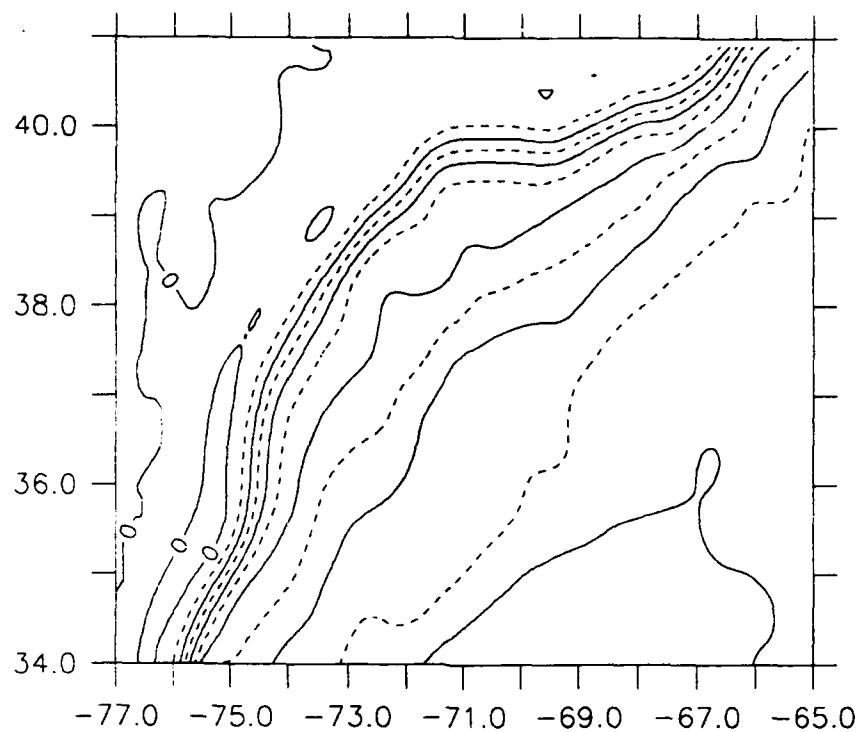


Figure 14: 110 km filtered topography. Note the coastline gets smoothed also and appears as the 0 contour.

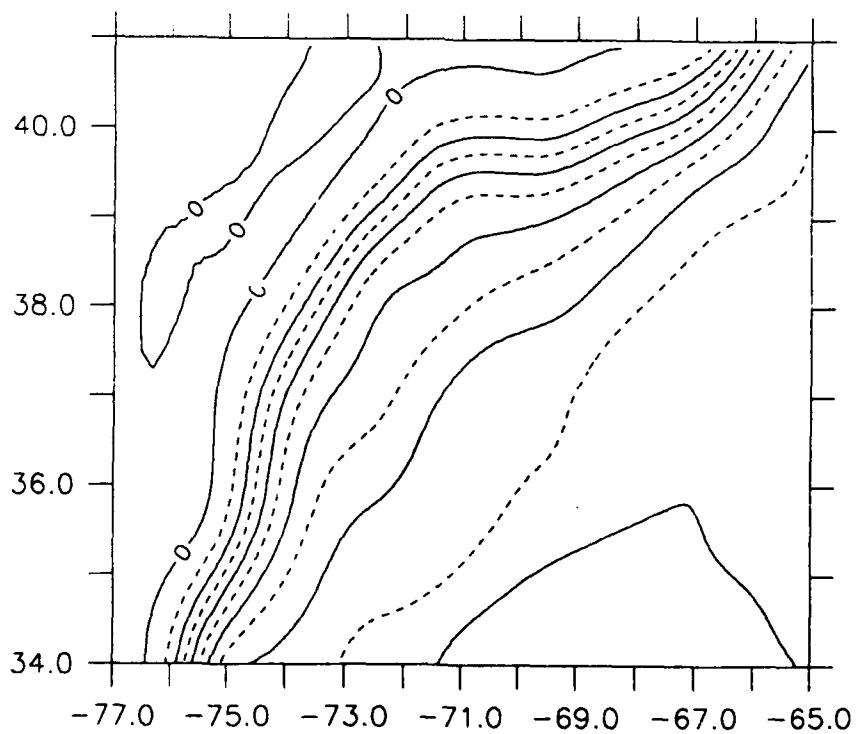


Figure 15: 220 km filtered topography.

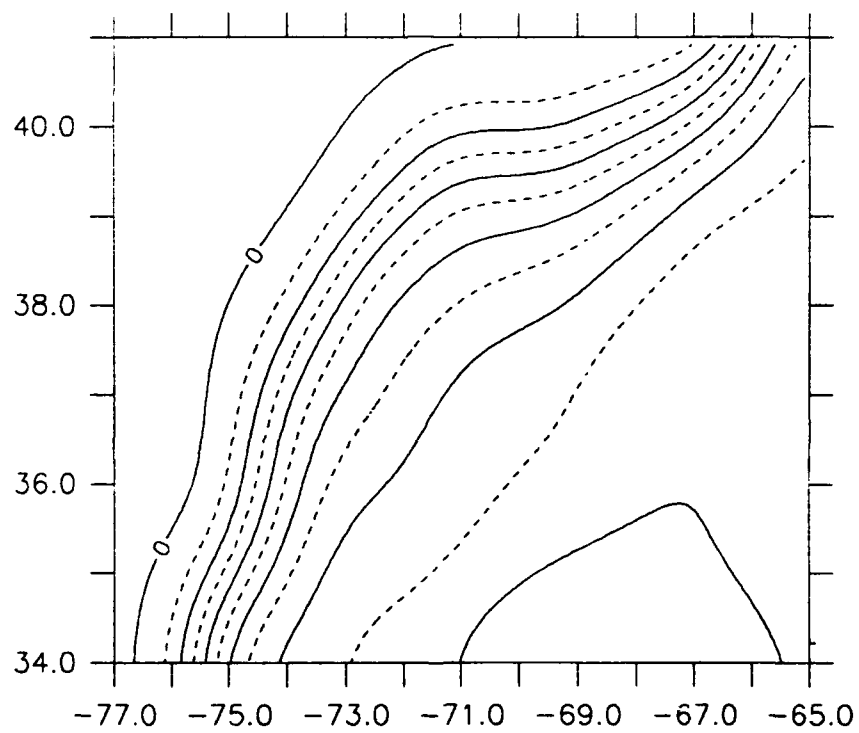


Figure 16: 330 km filtered topography.

10 Appendix C Program Codes

Most of the following program code is written in MATLAB. A good reference for MATLAB is Pro-MatlabTM for VAX/VMS Computers, 1991 by MathWorks Inc. The lone exception, the program **GET_ETOPO_II.FOR** is written in FORTRAN.

```

c This was a program of John Lillibridge. It was modified to
c run on the VAX. Command-line input will be removed.
c
c the output is ascii rather than direct access. The output is in a single
c column with longitude varying faster than lat.
c
c Also for the unformatted input file, the record size is in longwords,
c rather than bytes (Probably VAX-specific, this was not the case for
c the HP9000 John used).
c
c a s-scaling option has been removed. If image processing is your interest
c see the original get_etopo.for code.
c *****
c
c program get_etopo
c
c Get_etopo 1-15-91 J.L. Lillibridge @ NOAA/NOS/CaGS/NGSD/GRDL
c
c This program accesses the ETOPO5 topography/bathymetry database from NCAR.
c The original N & S-hemisphere ASCII files of heights/depths to the nearest
c meter were ingested into a direct-access binary data file with program
c etopo_da.f. The format of the resulting etopo.daf file has each height/depth
c value (>0, <0 respectively) stored as an I*2 integer, thus having a range
c of -32768-+32767. This will handle the range of actual heights on earth
c without losing the 1m resolution of the original data. The grid of values
c covers the earth at 5' lat/long spatial sampling, with strips of longitude
c values at a fixed latitude. Thus long. samples cycle most quickly (columns)
c and each row (record) of the file is for a given latitude which cycles more
c slowly. The total size of the direct-access I*2 file is then:
c 360 long.*12 (5' samples/deg.) * 180 lat.*12 * 2 bytes/number = 17.8 Mbytes.
c
c Note that the original NCAR distribution had the full strip of longitudes
c (360*12=4320 values) at the N. pole, but NO value at the S. pole. That's
c why there are 180*12=2160 records of latitude strips. The file is
c arranged with record #1 at the N. pole (with 4320 columns going from
c 0->359 55' longitude), and record #2160 is at -89 55', again with longitude
c columns from 0->359 55')
c
c The output of the program is a "windowed" direct access file based on user
c defined lat/long. limits. It is the users responsibility to keep track of
c the origin of the x-y values, since these are sent to the screen but not to
c the file itself. The limits are inclusive so that border values are sent
c out as well.
c
c implicit none
c
c integer stdout,nbytes
c parameter (stdout=16)
c integer*2 i,j,n,nrec,nwrd,nwrdn
c real min_lat,max_lat,min_long,max_long
c integer*2 min_rec,max_rec,min_wrd,max_wrd
c integer*2 inbuf(4320),outbuf(4320)
c character*80 fname,fnameo
c
c namelist/control/ fname,fnameo,min_lat,max_lat,min_long,max_long
c
c open(unit=21,status='old',form='formatted',readonly)
c read(21,nml=control)
c close(unit=21)

```

```

c
c open(10,file=fname,access='direct',recl=4320,
c status='old')
c
c Handle crossing of Greenwich below by passing over 360
c
c min_wrd=int((min_long*12+0.5)+1)
c min_long=(min_wrd-1.0)/12.0
c if (max_long.lt.min_long) max_long=max_long+360.
c max_wrd=int((max_long*12+0.5)+1)
c max_long=(max_wrd-1.0)/12.0
c
c values written as integer*2, recl in longword
c nwrd=max_wrd-min_wrd+1
c open(20,file=fnameo,form='formatted',status='new')
c
c min_rec=int((90-max_lat)*12+0.5)+1
c max_lat=90-(min_rec-1.0)/12.0
c min_rec=int((90-min_lat)*12+0.5)+1
c min_lat=90-(max_rec-1.0)/12.0
c nrec=max_rec-min_rec+1
c
c write(stdout,*) max_wrd-min_wrd+1,'lon values'
c write(stdout,*) 'nrec','lines'
c write(stdout,*) 'max_wrd-min_wrd+1','pixels/line'
c
c write(stdout,*) 'Min,Max Latitudes:',min_lat,max_lat
c write(stdout,*) 'Min,Max Longitudes:',min_long,max_long
c
c " Need to check this.... Seems like negative heights (bathymetry) need
c to be pairwise swapped (in longitude), while positive heights (topography)
c are OK as they were originally written to NCAR tape" assume correct first,
c no swapping.
c
c do nrec=min_rec,max_rec
c read(10,rec=nrec)(inbuf(i),i=1,4320)
c nwrdn=1
c do nwrd=min_wrd,max_wrd
c
c Handle Greenwich crossing here based on 360 longs * 12 (5' values/degree)
c
c if (nwrd.gt.4320) then
c n=nwrd-4320
c else
c n=nwrd
c endif
c write(stdout,*) nrec,nwrd
c outbuf(nwrdn)=inbuf(n)
c nwrdn=nwrdn+1
c enddo
c write(20,101) (outbuf(i),i=1,nwrdn-1)
c format(16)
c enddo
c
c close(10)
c close(20)
c stop
c end

```

_BURNSDB100: [CHRIS:RAYTRACE.N_FILES]GET_ETOPO.COM,4 6-MAY-1993 14:13 Page 1

c This command file runs the get_etopo_ii.for program.

c

c

\$ define for021 'pl'.ctrl

\$ define for016 'pl'.log

\$ run drv\$dub1:[etopo5]get_etopo_ii

c

c

c This is a typical control file for starting get_etopo_ii.for

c

c \$control fname1='dump.etopo5',fname2='n.atlan.topo',min_lat=35

max_lat=55, min_long=305, max_long=330 \$end

c

```

% This routine is used to "hand edit" seamounts from
% the etopo5 file BATHY_ETOPO5.mat
%
% BATHY_ETOPO5 contains 5'x5' sea-level elevations (i.e., z positive up)
% The range is from (80-45)N, (30-45)N.
%
% The "landscaped" bathymetry is stored in moved_BATH. This matrix must be
% saved to file after running this routine to permanently store the result
%
% load and initialize
continue='y';
load bathy_etopo5
moved_BATH=BATH;

% The loop is terminated if query to continue is negative.
% The query is located at the end of this loop
while ((continue ~='n') & (continue ~='N'))
    contour(moved_BATH)
    [n,m]=input(2);
    % contour the ocean bottom.
    % select seamount of interest
    % by clicking on the lower left
    % corner and then upper right
    % corner of a box which
    % encompasses the seamount.
    %
    % convert to matrix indices.
    n=round(n);m=flipud(182-round(m));
    h=BATH(m(1):m(2),n(1):n(2));
    cl=contour(h);
    cl=contour(h);
    % use mouse to click on contour
    % lines and manually label ones of
    % interest.
    level=input('landscape level? ');
    % input level to chop seamount off
    % at (include sign, e.g., -4500).
    x=find(h>level);
    h(x)=level*ones(x);
    % set elevation at those locations
    % to level.
    % contour result.
    % apply result to moved_BATH.
    moved_BATH(m(1):m(2),n(1):n(2))=h;
    plot(moved_BATH(m(1):m(2),:))
    % landscaped
    continue=input('continue (y/n)?','s');
    % continue?
end
disp(' Save moved_BATH to file, if run is complete and was successful')
return

```

```
function Y=filter2(bx,ax,by,ay,X);
% function Y=filter2(bx,ax,by,ay,X);
% This function does two dimensional filtering on a matrix of bathymetry
% values. The inputs are the appropriate Butterworth filter coefficients
% bx,ax,by,ay and the bathymetry matrix X.
[n,m]=size(X);
for column=1:m
    Y(:,column)=filtfilt(bx,ay,X(:,column));
end
for row=1:n;
    Y(row,:)=filtfilt(by,ay,Y(row,:));
end
```

```
function [c,cprime,c2prime,xknots]=compute_splines2(bath,lat,lon);
function [c,cprime,c2prime,xknots]=compute_splines2(bath,lat,lon);

% ARGUMENTS:
% bath      - the 5degx5deg tabulated ETOPO5 earth elevation in meters
% lat, lon  - the lat and lon of bath(1,1)

% OUTPUT:
% c, cprime, and c2prime
% - coefficient matrix (explained below)
% xknots    - knot sequence for B_i(x) (explained below)

% IDEA:
% bath(x,y)= a_i(y) B_i(x) summed from i=1,n
% where a_i is the coefficient of the ith B-spline
% a_i(y)= c_ij B_j(y) summed over j=1,m
% This routine fits splines to the bathymetry in the variable bath.
% B-splines are found along rows (lines of latitude) to get the a_i(y).
% For each i (1:n), a B-spline is fit to the m a_i's to get the c_ij.
% c is a matlab representation of the spline a_i(y) e.g. it contains the
% order, knots, values, ... according to spbrk.m
% cprime and c2prime are the first and second derivative of the splines
% set c.
```

```
% length(yknots) = m+k
% length(xknots) = m+k
% NOTE: the m and n will be augmented to reduce undulation at the
% edge of the domain of interest. Two points are padded to each end of x
% and y (lon and lat); corresponding bathymetry values at the padded points
% are found by linear extrapolation.
```

```
[m,n]=size(bath);
dlat=(1/2); dlon=dlat;
y=lat-dlat*(0:m-1); x=lon+dlon*(0:n-1);
yknots=[y(1)+dlat*(4,3,2,1) y(m)+dlat*(-1,-2,-3,-4)];
xknots=[x(1)+dlon*(-4,-3,-2,-1) x(n)+dlon*(1,2,3,4)];
clear x y
```

```
diff=bath(:,2)-bath(:,1);
adleft=[bath(:,1), bath(:,1)] - [2*diff diff];
diff=bath(:,n)-bath(:,n-1);
adright=[bath(:,n), bath(:,n)] + [diff 2*diff];
bath=[adleft bath adright]; n=n+4;
```

```
% for each row (line of lat) compute the spline
```

```
for i=1:m
    a(i,:)=spapi(xknots,xknots(3:n*2),bath(i,:));
end
```

```
% extract the coefficients of each row-fit and transpose so that
% that n-th row contains all the n-th B-spline-coefficients. The
% coefficient "field" is then augmented, as was done to the bathymetry.
```

```
a=a(:,3+[1:n]);
diff=a(:,2)-a(:,1);
adleft=[a(:,1), a(:,1)] - [2*diff diff];
diff=a(:,m)-a(:,m-1);
adright=[a(:,m), a(:,m)] + [diff 2*diff];
```

```
a=[adleft a adright]; m=m+4;
% fit splines to each row...
% convert to co-latitude so knot sequence is non-decreasing
% DON'T FORGET!!!
yknots=90-yknots;
for j=1:n
    c(j,:)=spapi(yknots,yknots(3:n*2),a(j,:));
    cprime(j,:)=fnder(c(j,:));
    c2prime(j,:)=fnder(cprime(j,:),2);
end
return
```

```

function gradh-test_splines(c,cprime,c2prime,xknots,xn,yn)
% 2-d spline interpolation
% function gradh-test_splines(c,cprime,c2prime,xknots,xn,yn)
% first four arguments are calculated using compute_splines.m
% xn,yn are the lon and lat of the location of interest (i.e., -70,35)
% returns gradh=[h h_x h_y h_xx h_yy h_xy]
n=length(xknots)-4;
yn=90-yn; % convert to co-latitude
% evaluate the spline for a_i at the
% specified co-latitude.
for i=1:n
    a(i)=fnval(c(i,:),yn);
    aprime(i)=fnval(cprime(i,:),yn);
    a2prime(i)=fnval(c2prime(i,:),yn);
end
d=spmak(xknots,a); % make the spline of the depth along the
% line of constant latitude, yn from the
% sum over i=1:n, a_i B_i(x)
ddy=spmak(xknots,aprine); % do a spline for d/dy depth
ddydy=spmak(xknots,a2prime); % do a spline for d/dydy depth
depth=fnval(d,xn); % evaluate depth at xn.
dy=fnval(ddy,xn); % evaluate y-slope at xn. -sign for colat
dydy=fnval(dddy,xn); % evaluate y-curvature at xn.
dddx=fnval(d,xn); % evaluate x-slope at xn.
dddx=fnval(d,xn); % evaluate x-slope at xn.
dddy=fnval(dddy,xn); % evaluate x-curvature at xn.
dxy=fnval(dddy,xn); % evaluate cross-derivative at xn.
gradh=[depth,dx,dy,dxx,dydy,dxy];

```

```

function [kfinal,lfinal,lam,theta]-initval(x,y,omega,LAM,parms,gradh)
% function [kfinal,lfinal,lam,theta]-initval(x,y,omega,LAM,parms,gradh)
%
% This function finds a starting k and l for a given wavelength and
% frequency of a TRW to be raytraced. The outputs are
% kfinal, lfinal, lam, & theta where lam is the bottom-trapping coefficient
% and theta is the angle as measured from the positive k axis. The
% inputs are x and y, which are the lon and lat starting point; omega,
% which is the frequency; LAM, which is the wavelength entered in km;
% parms, which is a matrix containing beta, dt, f, H, and n; and gradh,
% which is a matrix containing h, hx, hy, hxx, hyy, and hxy.
%
beta=parms(1);
f=parms(3);
H=parms(4);
n=parms(5);
h=gradh(1);
hx=gradh(2)/8.6263e+4;
hy=gradh(3)/11.1e+4;
LAM=LAM*1000;
%
con1=((2*pi*n)/(LAM*f))^2;
con2=(n^2 * beta)/(omega * f^2);
con3=(f*omega)/(n^2 * hx);
con4=hy/hx;
con5=((2*pi)/LAM);
%
kstart=l*con5; kfinish=con5;
while(1)
k=(kstart+((kfinish-kstart)/50000):kfinish)';
% This equation is the first of the two to be compared. It is simply the
% equation "(2pi/wavelength)^2 = k^2 + l^2" solved for l.
l1=-((con5^2 - k.^2).^(1/2));
%
dummy=(con1 + (con2*k)).^(1/2);
% This eliminates the chance of overflow from the tan.
arg=h*dummy;
dum=find(arg<-88);
if (~isempty(dum));
arg(dum)=-88*ones(dum);
end
%
% This is the second equation that will be compared. It is the second part
% of the dispersion relation solved once again for l.
l2=con4.*k - con3.*dummy.*tanh(arg);
%
% Now we take the difference of the two vectors and find the smallest
% difference, which will be the intersection point. The k and l that
% correspond to this point are those that will be output.
diff=l2-l1;
[abediff,abs(diff);
[imin,indx]=min(abs(diff);
kfinal=k(indx);
lfinal=l(indx);
lam=(n/f)*((kfinal^2 + lfinal^2 + (beta*kfinal)/omega)).^(1/2);
theta=atan(lfinal/kfinal)*180/pi;
return
end
end

```

```
function [CG, WCHECK, LOC, K, RSLOPE, LAMBDAL] = trwtrace(loc, k, prd, stepmax);
function [CG, WCHECK, LOC, K, RSLOPE, LAMBDAL] = trwtrace(loc, k, prd, stepmax);
% Modified from Erik Fields original leapfrog program to include
% a correction suggested by Bob Pickett to eliminate the time-splitting
% error inherent to the leapfrog method. Also the need for global
% variables is removed. The maximum and minimum lat. and lon. should be
% about one degree inside the boundaries of the spline area. These are
% used to prevent the trace from going outside of the area where the spline
% fit is accurate.
```

```
load parms
load [chris_data.raytrace.splines] spline_110km_eo
nccorr=5;
```

```
% assign maximum and minimum longitude cutoffs
maxlon=66;
minlon=76;
```

```
% assign maximum and minimum latitude cutoffs
maxlat=40;
minlat=35;
```

```
% assign parameters
beta = parms(1);
dt = parms(2);
if - parms(3);
H0 = parms(4);
W0 = parms(5);
```

```
% nondimensionalize items and define useful conversion factors
RD=N0*H0/i;
fscale=RD*beta;
% 2*pi/(prd*24*3600*fscale);
dt=dt*3600*fscale;
fbet=f/fscale;
dpxy=[RD/86.263e+3 RD/111.0e+3]; xydg=dpxy;
dpxy=[1 dpxy dpxy.*dpxy dpxy(1)*dpxy(2)]./H0;
k=RD*k;
```

```
% initialize counters and accumulation arrays
clear K LOC CG WCHECK
step=0; k=k; LOC=loc;
nccount=0;
```

```
% for now use uniform stratification
```

```
p=1; dndh=0;
```

```
while step<stepmax
```

```
gradh=dpxy.*test_splines(c,cprime,c2prime,xknots,loc(1),loc(2));
```

```
% save bottom slope
```

```
bslope=sqrt(gradh(2)^2+gradh(3)^2);
```

```
bg=fbet*gradh;
```

```
lam=n*sqrt(k(1)^2+k(2)^2+k(1)/w);
h=gradh(1); t=tanh(lam*h); g=lam*t; gl=t+lam*h*(1-t^2);
wcheck=(n^2*(bg(3)+k(1)-bg(2)*k(2))/(g*w))-1;
th=(k(1)*n*n)/(2*w*w*lam); ph=g/(g1*w); cmm=1/(ph+th);
dwdh=((2*w*ph-lam)*dndh/n) - (lam^2*(1-t^2)/g1);
dwdhx=(-n^2*k(2)*ph)/g; dwdhy=dwdhx*k(1)/k(2);
```

```
cg=cmm*[(ph*n^2*bg(3)/g)+(th*(w/k(1)+2*w*w))];
(-ph*n^2*bg(2)/g)+(th*(2*k(2)*w*w/k(1)))];
```

```
dkdt=cmm*[dwdh*gradh(2)+dwdhx*bg(4)+dwdhy*bg(6)];
dwdh*gradh(3)+dwdhx*bg(5)+dwdhy*bg(5)];
```

```
step=step+1
```

```
if step==1
```

```
nccount=nccount+1;
```

```
loc=LOC(step-1,:)+(xydg.*(2*dt*cg));
```

```
k=k(step-1,:)+2*dt*dkdt;
```

```
% trapezoidal correction
```

```
if nccount==nccorr
```

```
gradh=dpxy.*test_splines(c,cprime,c2prime,xknots,loc(1),loc(2));
```

```
bg=fbet*gradh;
```

```
lam=n*sqrt(k(1)^2+k(2)^2+k(1)/w);
```

```
h=gradh(1); t=tanh(lam*h); g=lam*t; gl=t+lam*h*(1-t^2);
```

```
th=(k(1)*n*n)/(2*w*w*lam); ph=g/(g1*w); cmm=1/(ph+th);
```

```
dwdh=((2*w*ph-lam)*dndh/n) - (lam^2*(1-t^2)/g1);
```

```
dwdhx=(-n^2*k(2)*ph)/g; dwdhy=dwdhx*k(1)/k(2);
```

```
cg_g=cmm*[(ph*n^2*bg(3)/g)+(th*(w/k(1)+2*w*w))];
```

```
(-ph*n^2*bg(2)/g)+(th*(2*k(2)*w*w/k(1)))];
```

```
dkdt_g=cmm*[dwdh*gradh(2)+dwdhx*bg(4)+dwdhy*bg(6)];
```

```
dwdh*gradh(3)+dwdhx*bg(5)+dwdhy*bg(5)];
```

```
loc=LOC(step,:)+(xydg.*(dt*(cg+cg_g)/2));
```

```
k=k(step,:)+(dt*(dkdt+dkdt_g)/2);
```

```
gradh=dpxy.*test_splines(c,cprime,c2prime,xknots,loc(1),loc(2));
```

```
bg=fbet*gradh;
```

```
lam=n*sqrt(k(1)^2+k(2)^2+k(1)/w);
```

```
h=gradh(1); t=tanh(lam*h); g=lam*t;
```

```
wcheck=(n^2*(bg(3)+k(1)-bg(2)*k(2))/(g*w))-1;
```

```
nccount=0;
```

```
end
```

```
else
```

```
% euler time step
```

```
loc=LOC(step,:)+(xydg.*(dt*cg));
```

```
k=k(step,:)+(dt*dkdt);
```

```
% trapezoidal correction
```

```
gradh=dpxy.*test_splines(c,cprime,c2prime,xknots,loc(1),loc(2));
```

```
bg=fbet*gradh;
```

```
lam=n*sqrt(k(1)^2+k(2)^2+k(1)/w);
```

```
h=gradh(1); t=tanh(lam*h); g=lam*t; gl=t+lam*h*(1-t^2);
```

```
th=(k(1)*n*n)/(2*w*w*lam); ph=g/(g1*w); cmm=1/(ph+th);
```

```
dwdh=((2*w*ph-lam)*dndh/n) - (lam^2*(1-t^2)/g1);
```

```
dwdhx=(-n^2*k(2)*ph)/g; dwdhy=dwdhx*k(1)/k(2);
```

```
cg_g=cmm*[(ph*n^2*bg(3)/g)+(th*(w/k(1)+2*w*w))];
```

```
(-ph*n^2*bg(2)/g)+(th*(2*k(2)*w*w/k(1)))];
```

```
dkdt_g=cmm*[dwdh*gradh(2)+dwdhx*bg(4)+dwdhy*bg(6)];
```

```
dwdh*gradh(3)+dwdhx*bg(5)+dwdhy*bg(5)];
```

```
loc=LOC(step,:)+(xydg.*(dt*(cg+cg_g)/2));
```

```
k=k(step,:)+(dt*(dkdt+dkdt_g)/2);
```

```
gradh=dpxy.*test_splines(c,cprime,c2prime,xknots,loc(1),loc(2));
```

```
bg=fbet*gradh;
```

```
lam=n*sqrt(k(1)^2+k(2)^2+k(1)/w);
```

```
% euler time step
```

```
loc=LOC(step,:)+(xydg.*(dt*cg));
```

```
k=k(step,:)+(dt*dkdt);
```

```
% trapezoidal correction
```

```
gradh=dpxy.*test_splines(c,cprime,c2prime,xknots,loc(1),loc(2));
```

```
bg=fbet*gradh;
```

```
lam=n*sqrt(k(1)^2+k(2)^2+k(1)/w);
```

```

p=gradh(1); t=tanh(lam*h); g=lam*t;
wcheck=(n^2*(bg(3)*k(1)-bg(2)*k(2))/(g*w))-1;
end

```

```

if step==1
CG=CG+sqrt(cg(1)^2+cg(2)^2);
wcheck=[wcheck,wcheck];
ESLOPE=[ESLOPE,belops];
LAMBDA=[LAMBDA,lam];

```

```

else
CG=sqrt(cg(1)^2+cg(2)^2);
wcheck=wcheck;
ESLOPE=ESLOPE;
LAMBDA=LAM;
end

```

```

LOC=[LOC,loc];
K=[K,k];

```

```

% Put in cutoffs so that trace cannot go out of spline area.

```

```

if loc(1) > maxlon
step=stepmax;
end

```

```

if loc(1) < minlon
step=stepmax;
end

```

```

if loc(2) > maxlat
step=stepmax;
end

```

```

if loc(2) < minlat
step=stepmax;
end

```

```

end % while
CG=(RD*scale+3.6*24)*CG;

```

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified			1b. RESTRICTIVE MARKINGS		
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION/AVAILABILITY OF REPORT Distribution for Public Release; Distribution is unlimited		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE			5. MONITORING ORGANIZATION REPORT NUMBER(S)		
4. PERFORMING ORGANIZATION REPORT NUMBER(S) GSO University of Rhode Island Tech Report 93-1 Graduate School of Oceanography			7a. NAME OF MONITORING ORGANIZATION		
6a. NAME OF PERFORMING ORGANIZATION Univ. of Rhode Island Grad. School of Oceanography		6b. OFFICE SYMBOL (If applicable)		7b. ADDRESS (City, State, and ZIP Code)	
6c. ADDRESS (City, State, and ZIP Code) South Ferry Road Narragansett, RI 02882		8a. NAME OF FUNDING/SPONSORING ORGANIZATION Office of Naval Research National Science Foundation		8b. OFFICE SYMBOL (If applicable)	
8c. ADDRESS (City, State, and ZIP Code) 800 N. Quincy St., Arlington, VA 22217 1800 G. St., N.W., Washington DC 20550		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER			
11. TITLE (Include Security Classification) Ray Tracing on Topographic Rossby Waves.		10. SOURCE OF FUNDING NUMBERS			
		PROGRA ELEMEN		PROJECT NO.	TASK NO.
				WORK UNIT ACCESSION NO.	
12. PERSONAL AUTHOR(S) Christopher Meinen, Erik Fields, Robert Pickart, D. Randolph Watts					
13a. TYPE OF REPORT Summary		13b. TIME COVERED FROM TO		14. DATE OF REPORT (Year, Month, Day) May, 1993	
15. PAGE COUNT 47					
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB-GROUP	Topographic Rossby Waves, SYNOP, and Mid Atlantic Bight		
19. ABSTRACT (Continue on reverse if necessary and identify by block number) Topographic Rossby Waves (TRWs) have been identified with the largest variability in deep current meter records along the continental slope in the Mid-Atlantic Bight (MAB). Ray tracing theory is applied to TRWs using the real bottom topography of the MAB and the observed stratification. The depression relation for TRWs is derived and various wavenumber limits are discussed. A computational method for tracing the waves is presented including the necessity of smoothing the bathymetry. In the examples shown, TRWs with periods of 24-48 days generally propagate southwestward, changing their wavelengths from 400 to 100 kilometers in response to the change in bottom slope. TRW paths are shown that connect from the SYNOP Central Array near 68°W to the SYNOP Inlet Array near Cape Hatteras.					
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION		
22a. NAME OF RESPONSIBLE INDIVIDUAL			22b. TELEPHONE (Include Area Code)		22c. OFFICE SYMBOL