NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A THREE-DIMENSIONAL COUPLED NORMAL MODE MODEL FOR SOUND PROPAGATION IN SHALLOW WATER WITH IRREGULAR BOTTOM BATHYMETRY

by

George A. Sagos

December, 1992

Thesis Advisor: Ching-Sang Chiu
Thesis Co-Advisor: James H. Miller

Unclassified

Security Classification of this page

REPORT DOCUMENTATION PAGE

1a Report Security Classification Unclassified	1b Restrictive Markings
2a Security Classification Authority	3 Distribution Availability of Report: Approved for public release; distribution is unlimited.
2b Declassification/Downgrading Schedule	
4 Performing Organization Report Number(s)	5 Monitoring Organization Report Number(s)
6a Name of Performing Organization Naval Postgraduate School; 6b Office Symbol *(If Applicable)* 33	7a Name of Monitoring Organization Naval Postgraduate School
6c Address *(city, state, and ZIP code)* Monterey, CA 93943-5000	7b Address *(city, state, and ZIP code)* Monterey, CA 93943-5000
8a Name of Funding/Sponsoring Organization NPS; 8b Office Symbol *(If Applicable)*	9 Procurement Instrument Identification Number
8c Address *(city, state, and ZIP code)*	10 Source of Funding Numbers: Program Element Number / Project No / Task No / Work Unit Accession No

11 Title *(Include Security Classification)* A THREE-DIMENSIONAL COUPLED NORMAL MODE MODEL FOR SOUND PROPAGATION IN SHALLOW WATER WITH IRREGULAR BOTTOM BATHYMETRY

12 Personal Author(s) George A. Sagos

13a Type of Report	13b Time Covered	14 Date of Report *(year, month,day)*	15 Page Count
Master's Thesis	From To	December 1992	71

16 Supplementary Notation The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 Cosati Codes			18 Subject Terms *(continue on reverse if necessary and identify by block number)*
Field	Group	Subgroup	Ocean acoustics;Coupled Normal Modes

19 Abstract *(continue on reverse if necessary and identify by block number*

A three-dimensional (3D) coupled normal mode model for studying sound propagation in a complex coastal environment is developed. This development corresponds to a significant upgrade of an earlier version of the model in which a flat, rigid bottom was used. By imposing the general boundary conditions for an irregular, non-rigid bottom, the coupling coefficient integrals in the system of differential equations governing the mode amplitude are reformulated. The model upgrade entails a numerical implementation of the revised formulae. With the improved physics, this latest version is capable of modeling the 3D acoustic wave-field in shallow water where sound speed, water depth and sediment properties can vary with horizontal location. To demonstate this enhanced capability, the model is used here to simulate the interactions of the normal modes as they propagate up a sloping bottom.

20 Distribution/Availability of Abstract [X] unclassified/unlimited [ ] same as report [ ] DTIC users	21 Abstract Security Classification Unclassified	
22a Name of Responsible Individual Ching-Sang Chiu	22b Telephone *(Include Area code)* (408) 646-3239	22c Office Symbol OC/Ci

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted — security classification of this page

All other editions are obsolete — Unclassified

Approved for public release; distribution is unlimited.

A three-dimensional coupled normal mode model for sound propagation in shallow water with irregular bottom bathymetry

by

George A. Sagos
Lieutenant J.G., Hellenic Navy

Submitted in partial fulfillment
of the requirements for the degrees of

MASTER OF SCIENCE IN ENGINEERING ACOUSTICS
and
MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1992

Author: ______________________
George A. Sagos

Approved by: ______________________
Ching-Sang Chiu, Thesis Advisor

____
James H. Miller, Co-Advisor

Laura L Ehret
____
Laura L. Ehret, Second Reader

____
Anthony A. Atchley, Chairman
Engineering Acoustics
Academic Comittee

M.A. Morgan
____
Michael A. Morgan, Chairman
Department of Electrical
and Computer Engineering

ABSTRACT

A three-dimensional (3D) coupled normal mode model for studying sound propagation in a complex coastal environment is developed. This development corresponds to a significant upgrade of an earlier version of the model in which a flat, rigid bottom was used. By imposing the general boundary conditions for an irregular, non-rigid bottom, the coupling coefficient integrals in the system of differential equations governing the mode amplitude are re-formulated. The model upgrade entails a numerical implementation of the revised formulae. With the improved physics, this latest version is capable of modeling the 3D acoustic wave-field in shallow water where sound speed, water depth and sediment properties can vary with horizontal location. To demonstrate this enhanced capability, the model is used here to simulate the interactions of the normal modes as they propagate up a sloping bottom.

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation and gratitude to my advisor Ching-Sang Chiu and also to Laura L. Ehret for their assistance and guidance in order to complete this thesis.

I dedicate this thesis to my wife, Denise, and my two sons, Anthony and Dennis, for their great help, support, and encouragement during all my time at NPS.

Finally a special thanks is due to the Hellenic Navy for giving me this special opportunity to study at the Naval Postgraduate School.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

A. BACKGROUND

There are three approaches to model three-dimensional(3D) sound propagation in the ocean: ray theory, parabolic equation approximation and normal mode theory.

Ray theory gives an approximate, planewave-like solution to the wave equation, which is valid at high enough frequencies and in media with gradual variations. The ray solution is constructed by raytracing. The acoustic rays provide for a visual, physical description of sound transmission in the ocean. The ray solution, however, neglects sound diffraction and dispersion and thus needs corrections near caustics and turning points. These corrections may sometimes be mathematically complicated. The Hamiltonian Acoustic Ray Tracing Program for the Ocean (HARPO) is the only 3D ray theory model available today. This computer code was originally developed by Jones et al. [Ref. 1] for the computation of 3D rays.

The parabolic equation approximation method (PE) was introduced by Tappert [Ref. 2]. PE is a "full-wave" method that accounts for both sound diffraction and dispersion. It provides for numerical solutions to the wave equation which are accurate for energy propagating at low grazing angles. The accuracy generally degrades as the angle increases. The backscattered energy is generally neglected in this approximation. A versatile 3D PE model has been developed by Lee et al. [Ref. 3] using an implicit finite difference scheme. Another 3D PE model was developed earlier by Baer [Ref. 4] which uses a split-step Fourier algorithm. The PE model of Lee has a wider angle capability, i.e., it models sound energy travelling at steeper angles more accurately.

Finally, normal mode theory describes sound propagation as a collection of eigenfunctions, called normal modes, which are a natural set of vertical vibration modes in the sound channel. Just like PE, normal mode theory is a "full-wave" approach. The original normal mode theory assumes a horizontally stratified propagation medium. This assumption is valid for many short-range, deep-water cases, where range and azimuthal variations are negligibly small. Pierce [Ref. 5] extended the theory to account for horizontal sound speed, bathymetry and bottom-property variations. These variations produce mode coupling phenomena (in which energy exchange between modes takes place). A 3D coupled normal mode model has been developed by Chiu and Ehret [Ref. 6]. This model is capable of simulating mode-mode interactions due to a 3D varying sound speed field. The effects of bottom bathymetry variations and sediment property, however, are not modeled.

B. THESIS OBJECTIVES AND OUTLINE

The main objective of this thesis is to improve the Chiu-Ehret [Ref. 6] 3D coupled normal mode model by including the effects of bathymetry variations and sediment properties on sound propagation. The improved model is useful for studying sound propagation in shallow water environments where significant bottom interaction is expected.

In Chapter II, 3D coupled mode theory is first reviewed. The formulae for the mode coupling coefficients in the system of differential equations governing the mode amplitude functions are derived. In the derivation, the general boundary conditions for an irregular, non-rigid bottom are used.

In Chapter III, alternative expressions for the mode coupling coefficients are derived. These expressions allow for an easier numerical implementation. The improved model is used to examine the effects of a sloping bottom on upslope sound propagation. The validity of the adiabatic approximation is also examined. Conclusions are given in Chapter IV.

The products coming out of this thesis are computer subroutines to include bathymetry variations and sediment properties in the 3D coupled mode model of Chiu and Ehret [Ref. 6]. The new routines are listed in the Appendix.

II. 3D COUPLED NORMAL MODE THEORY

In the mathematical formulation that follows, a cylindrical coordinate system will be used (see Fig. 1). The z-axis is perpendicular to the ocean surface and is positive downward, r is range from the source location (i.e., the origin) and θ is the azimuthal angle (positive clockwise). Sound speed in the water column, c_1, is a function of r, z and θ, where sound speed in the sediment, c_2, is assumed to be r and θ dependent only. The density of the water column, ρ_1, is considered to be constant. The density of the sediment, ρ_2, is also considered to be constant. The water-sediment interface is located at $z=H(r,\theta)$.

A. THE MATHEMATICAL PROBLEM

In the case of isodensity layers, the 3D, homogeneous, monofrequency Helmholtz Equation governing the acoustic pressure, p, is:

$$\nabla^2 p(z,r,\theta) + k^2(z,r,\theta)p(z,r,\theta) = 0 \qquad (1)$$

where $k(z,r,\theta)=\omega/c(z,r,\theta)$ is the acoustic wavenumber, ω is the source angular frequency and c is sound speed (c_1 in the water layer and c_2 in the sediment layer).

A quasi-separable solution to Eq. (1) is postulated, which is locally a linear combination of normal modes or depth

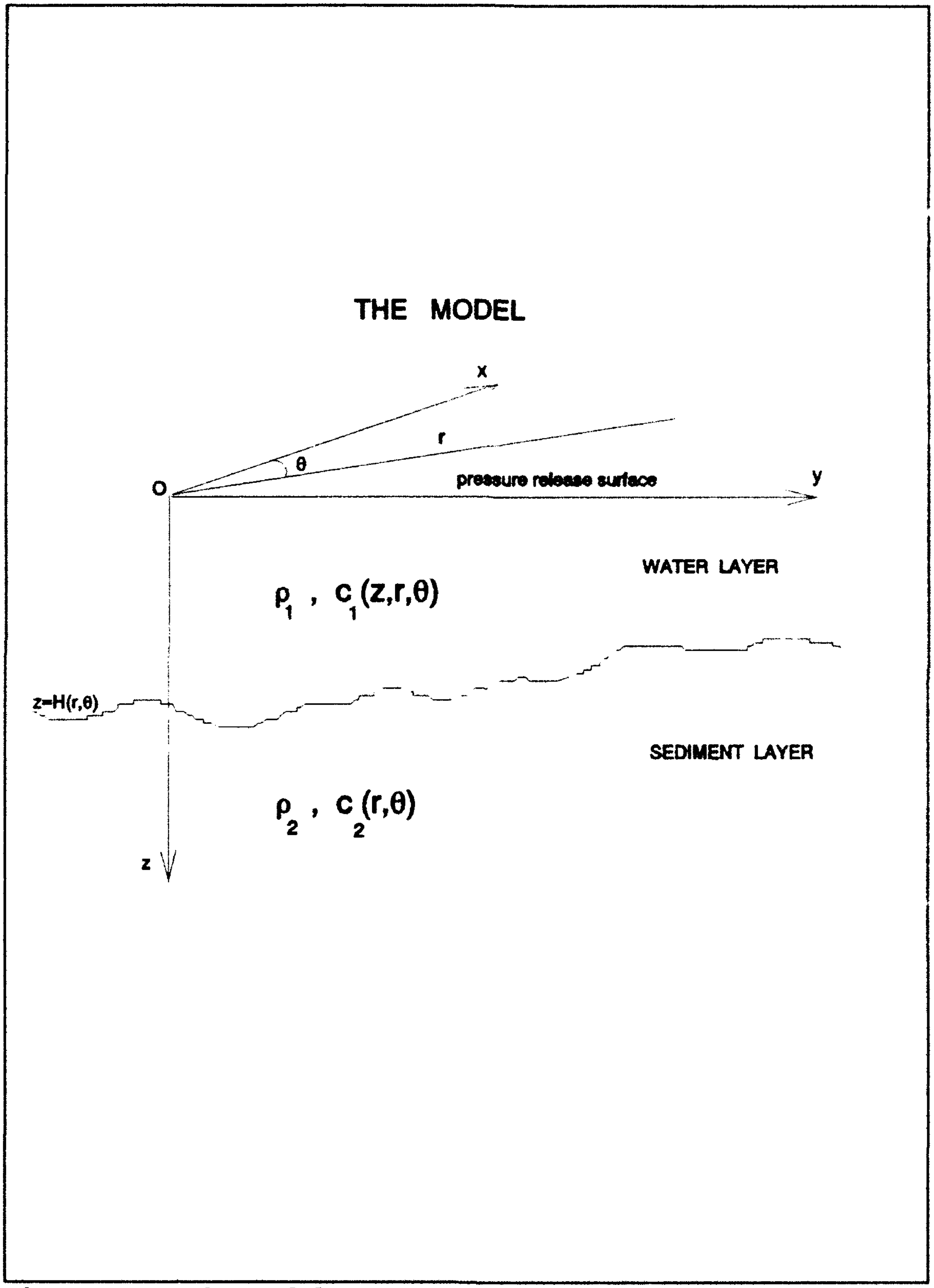

Figure 1. The model geometry

functions, Z_n:

$$p(z,r,\theta) = \sum_{n} R_n(r,\theta)\, Z_n(z;r,\theta) \tag{2}$$

where R_n are the mode amplitude functions and n is the mode number.

The normal modes Z_n are required to satisfy the depth equation:

$$\left[\frac{\partial}{\partial z^2} + k^2(z,r,\theta) - k^2{}_n(r,\theta)\right] Z_n(z;r,\theta) = 0 \tag{3}$$

where k_n is the horizontal component of the wavenumber (eigenvalue) associated with the n^{th} mode.

It can be easily shown, using the boundary conditions for Z_n (to be derived next) and the depth equation (Eq. (3)), that the normal modes form a complete set of orthogonal functions, with the inverse of the medium density ρ as a weighting function in the normalization:

$$\int_0^\infty \frac{1}{\rho(z)} Z_n(z;r,\theta)\, Z_m(z;r,\theta)\, dz = \delta_{nm} \tag{4}$$

where δ_{nm} is the Kronecker delta. Note that the integration is carried over the entire depth from 0 to ∞. Also, note that the density ρ in the model is considered to be constant in each layer (ρ_1 in the water and ρ_2 in the sediment).

B. THE BOUNDARY CONDITIONS

The appropriate boundary condition for the acoustic pressure at the sea surface is

$$p_1(z=0;r,\theta) = 0 \tag{5}$$

where the subscript 1 denotes the water column. This pressure release condition implies that the normal modes Z_n must also be zero at the sea surface, i.e.,

$$Z_n(z=0;r,\theta) = 0 \tag{6}$$

It also implies that the horizontal derivatives of Z_n at z=0 are zero, i.e.,

$$\frac{\partial Z_n(z=0r,\theta)}{\partial r} = 0 \tag{7}$$

$$\frac{1}{r}\frac{\partial Z_n(z=0;r,\theta)}{\partial \theta} = 0 \tag{8}$$

At the interface between the sediment and the water column, i.e., at $z=H(r,\theta)$, the boundary conditions are continuity of pressure and continuity of the particle velocity component normal to the interface:

$$p_1(z=H;r,\theta) = p_2(z=H;r,\theta) \tag{9}$$

$$\frac{1}{\rho_1}\nabla p_1 . \hat{n} = \frac{1}{\rho_2}\nabla p_2 . \hat{n} \tag{10}$$

The subscripts 1 and 2 denote the water column and the sediment, respectively.

The unit directional vector $\hat{n}$ normal to the bottom interface is

$$\hat{n} = \frac{\nabla(z-H(r,\theta))}{|\nabla(z-H(r,\theta)|} = \frac{\hat{z}-\frac{\partial H(r,\theta)}{\partial r}\hat{r}-\frac{1}{r}\frac{\partial H(r,\theta)}{\partial \theta}\hat{\theta}}{\left[1+(\frac{\partial H(r,\theta)}{\partial r})^2+(\frac{1}{r}\frac{\partial H(r,\theta)}{\partial \theta})^2\right]^{1/2}} \tag{11}$$

where $\hat{z}$, $\hat{r}$ and $\hat{\theta}$ are the unit directional vectors associated with the z, r and θ directions, respectively. In the case of a small bottom slope, the boundary condition of Eq. (10) can be approximated by

$$\frac{1}{\rho_1}\frac{\partial p_1(z=H;r,\theta)}{\partial z} = \frac{1}{\rho_2}\frac{\partial p_2(z=H;r,\theta)}{\partial z} \tag{12}$$

The small slope approximation is accurate when

$$\left|\frac{\partial H(r,\theta)}{\partial r}\right| \ll 1 \quad , \quad \left|\frac{1}{r}\frac{\partial H(r,\theta)}{\partial \theta}\right| \ll 1 \tag{13}$$

Following Eq. (9) and Eq. (12), we obtain the following boundary conditions at the water-sediment interface for the normal modes:

$$\frac{1}{\rho_1}\frac{\partial Z_{1n}(z=H;r,\theta)}{\partial z} = \frac{1}{\rho_2}\frac{\partial Z_{2n}(z=H;r,\theta)}{\partial z} \tag{14}$$

$$Z_{1n}(z=H;r,\theta) = Z_{2n}(z=H,r,\theta) \tag{15}$$

These boundary conditions hold for each individual normal mode because they are orthogonal functions.

The boundary condition for p at $z \to \infty$ is

$$p_2(z\to\infty,r,\theta) = 0 \tag{16}$$

This implies that the normal modes and their horizontal derivatives are also zero as $z \to \infty$:

$$Z_{2n}(z\to\infty;r,\theta) = 0 \tag{17}$$

$$\frac{\partial Z_{2n}(z\to\infty;r,\theta)}{\partial r} = 0 \tag{18}$$

$$\frac{1}{r}\frac{\partial Z_{2n}(z\to\infty;r,\theta)}{\partial\theta} = 0 \tag{19}$$

C. MODE COUPLING COEFFICIENTS DERIVATION

Substituting Eq. (2) into Eq. (1), multiplying by $Z_m(z;r,\theta)/\rho$, integrating over the entire depth and finally rearranging terms, we obtain the coupled mode equations governing the mode amplitude functions:

$$[\nabla_h^2+k^2{}_m(r,\theta)]\ R_m(r,\theta) = -\sum_n [\vec{A}_{mn}(r,\theta)\nabla_h R_n(r,\theta) + B_{mn}(r,\theta)R_n(r,\theta)] \tag{20}$$

where the two mode coupling coefficients are defined as

$$\vec{A}_{mn}(r,\theta) = 2\int_0^\infty \frac{1}{\rho} Z_m(z;r,\theta)\nabla_h Z_n(z;r,\theta)\,dz \tag{21}$$

and

$$B_{mn}(r,\theta) = \int_0^\infty \frac{1}{\rho} Z_m(z;r,\theta)\nabla^2{}_h Z_n(z;r,\theta)\,dz \tag{22}$$

∇_h is the horizontal gradient operator, i.e.,

$$\nabla_h = \hat{r}\frac{\partial}{\partial r} + \hat{\theta}\frac{1}{r}\frac{\partial}{\partial \theta} \tag{23}$$

The mode coupling coefficients are measures of the significance of exchange of acoustic energy between modes resulting from horizontal variations in the medium. As the variations become stronger, the coupling coefficients become larger and so is the energy exchange. In the case of a completely range independent medium, the coupling coefficients are identically zero and the RHS of Eq. (20) vanishes. In such case, the modes propagate independently of each other.For range-dependent cases, the neglect of mode coupling leads to the adiabatic approximation [Ref. 5].

Cylindrical spreading can be removed from the coupled mode equation (Eq. (20)) by replacing the mode amplitude function $R_n(r,\theta)$ with $P_n(r,\theta)/r^{1/2}$. The result is

$$\left[\frac{\partial^2}{\partial r^2} + k_m^2(r,\theta) + \frac{1}{r^2}\frac{\partial^2}{\partial \theta^2} + \frac{1}{4r^2}\right] P_n(r,\theta) =$$

$$= \sum_n \left(\vec{A}_{mn}(r,\theta) \cdot \left[\nabla_h P_n(r,\theta) - \hat{r}\frac{P_n(r,\theta)}{2r}\right] + B_{mn}(r,\theta)\, P_n(r,\theta)\right) \tag{24}$$

III. THE NUMERICAL MODEL AND EXAMPLE RUNS

In this chapter, the procedure to upgrade the Chiu-Ehret model [Ref. 6] is discussed. The upgrade has entailed the derivation of alternative expressions for the mode coupling coefficients and the generation of new code to compute these coefficients based on the alternative expressions.

The numerical results from two simple example model runs associated with two different bottom slopes are also presented in this chapter. Both cases deal with upslope propagation in isospeed layers. These runs have allowed for an examination of the coupling between modes caused by bathymetry change. In addition, they have allowed for an examination of the validity of the adiabatic approximation.

A. THE CHIU-EHRET APPROACH

In the far field, i.e., kr>>1 , the coupled mode equation (Eq. (24)) for the mode amplitude functions, can be recast as

$$\left[\frac{\partial^2}{\partial r^2}+k_m^2(r,\theta)+\frac{1}{r^2}\frac{\partial^2}{\partial\theta^2}\right]P_n(r,\theta) \doteq \sum_n \left(\vec{A}_{mn}(r,\theta)\,.\nabla_h P_n(r,\theta)+B_{mn}(r,\theta)\,P_n(r,\theta)\right) \tag{25}$$

In the Chiu-Ehret model [Ref. 6], P_n is decomposed as

$$P_n(r,\theta) = U_n(r,\theta)\ e^{j\phi_n(r,\theta)}$$

$$\phi_n(r,\theta) = \int_0^r k_n(r,\theta)\ dr \tag{26}$$

where U_n is the slowly varying complex envelope of P_n modulating the rapidly varying two-dimensional (2D) adiabatic solution, i.e., $\exp(j\phi_n)$, and ϕ_n is the adiabatic phase. The Chiu-Ehret model numerically computes the envelopes U_n instead of P_n using Runge-Kutta schemes.

B. ALTERNATIVE EXPRESSIONS FOR THE MODE COUPLING COEFFICIENTS

For simpler numerical implementation, the expressions of the mode coupling coefficients in Eq. (21) and Eq. (22) are rewritten in alternative forms. These alternative forms do not require integrations of expressions involving the horizontal derivatives of normal modes. In the following, the derivation of these alternative forms is presented.

1. VECTOR MODE COUPLING COEFFICIENT, $\vec{A}_{mn}$

a. Case of $m \neq n$

Applying the horizontal gradient operator ∇_h to both sides of the depth equation (Eq. (3)), we get

$$\frac{\partial^2 \nabla_h Z_n}{\partial z^2} + 2[k(z,r,\theta)\nabla_h k(z,r,\theta) - k_n(r,\theta)\nabla_h k_n(r,\theta)]Z_n +$$

$$[k^2(z,r,\theta) - k^2{}_n(r,\theta)]\nabla_h Z_n = 0 \tag{27}$$

Multiplying Eq. (27) by $Z_m(z;r,\theta)/\rho$ and then integrating over the entire depth, we get

$$\int_0^\infty \frac{1}{\rho} Z_m \frac{\partial^2 \nabla_h Z_n}{\partial z^2}\, dz + \int_0^\infty (k^2 - k_n^2)\frac{1}{\rho} Z_n\ \nabla_h Z_n\, dz =$$

$$2k_n\ \nabla_h k_n\ \delta_{nm} - \int_0^\infty \frac{2}{\rho} k \nabla_h k\ Z_n\ Z_m\, dz \tag{28}$$

In order to recast the first term of Eq. (28) in a form useful for this derivation, we first use integration by parts twice with respect to z. The resulting expression, after some lengthy manipulations, is

$$\int_0^\infty \frac{1}{\rho} Z_m \frac{\partial^2 \nabla_h Z_n}{\partial z^2}\, dz = \int_0^\infty \frac{1}{\rho}\ \nabla_h Z_n\ \frac{\partial^2 Z_m}{\partial z^2}\, dz + \frac{1}{\rho_1} Z_{1m} \frac{\partial \nabla_h Z_{1n}}{\partial z}\bigg|_0^{H(r,\theta)} +$$

$$\frac{1}{\rho_2} Z_{2m} \frac{\partial \nabla_h Z_{2n}}{\partial z}\bigg|_{H(r,\theta)}^{\infty} - \frac{1}{\rho_1}\frac{\partial Z_{1m}}{\partial z}\nabla_h Z_{1n}\bigg|_0^{H(r,\theta)} - \frac{1}{\rho_2}\frac{\partial Z_{2m}}{\partial z}\nabla_h Z_{2n}\bigg|_{H(r,\theta)}^{\infty}$$

(29)

Again, the subscripts 1 and 2 denote the water column and the sediment, respectively.

Application of the boundary conditions Eqs. (6), (7), (8), (14), (15), (17), (18) and (19) to Eq. (29), yields subsequently

$$\int_0^\infty \frac{1}{\rho} Z_m \frac{\partial \nabla_h^2 Z_n}{\partial z^2}\, dz =$$
$$\int_0^\infty \frac{1}{\rho} \nabla_h Z_n \frac{\partial^2 Z_m}{\partial z^2}\, dz + \tag{30}$$
$$\frac{1}{\rho_1} Z_{1m} \left[\frac{\partial \nabla_h Z_{1n}}{\partial z} - \frac{\partial \nabla_h Z_{2n}}{\partial z} \right] \Bigg|_{z=H(r,\theta)} -$$
$$\frac{\partial Z_{1m}}{\partial z} \left[\frac{1}{\rho_1} \nabla_h Z_{1n} - \frac{1}{\rho_2} \nabla_h Z_{2n} \right] \Bigg|_{z=H(r,\theta)}$$

Replacing the first term of Eq. (28) by Eq. (30) and then making use of the depth equation (Eq. (3)), we obtain the following alternative expression for the vector mode coupling coefficient, for $m \neq n$:

$$\vec{A}_{mn}(r,\theta) = \beta_{mn}\hat{r} + \gamma_{mn}\hat{\theta} = \frac{2}{k_n^2(r,\theta) - k_m^2(r,\theta)} \; [2\int_0^\infty \frac{1}{\rho} k \, \nabla_h k \; Z_m \; Z_n \; dz \; +$$

$$\frac{1}{\rho_1} Z_{1m} \left(\frac{\partial \nabla_h Z_{1n}}{\partial z} - \frac{\partial \nabla_h Z_{2n}}{\partial z} \right) \bigg|_{z=H(r,\theta)} - \; \frac{\partial Z_{1m}}{\partial z} \left(\frac{1}{\rho_1} \nabla_h Z_{1n} - \frac{1}{\rho_2} \nabla_h Z_{2n} \right) \bigg|_{z=H(r,\theta)} \;] \tag{31}$$

or equivalently, in light of the boundary conditions Eq. (14) and Eq. (15),

$$\vec{A}_{mn}(r,\theta) = \beta_{mn}\hat{r} + \gamma_{mn}\hat{\theta} = \frac{2}{k_n^2(r,\theta) - k_m^2(r,\theta)} \; [2\int_0^\infty \frac{1}{\rho} k \, \nabla_h k \; Z_m \; Z_n \; dz \; +$$

$$\frac{1}{\rho_1} \left(1 - \frac{\rho_1}{\rho_2} \right) Z_{1m} \frac{\partial \nabla Z_{1m}}{\partial z} \bigg|_{z=H(r,\theta)} - \; \left(\frac{1}{\rho_1} - \frac{1}{\rho_2} \right) \frac{\partial Z_{1m}}{\partial z} \; \nabla_h Z_{1n} \bigg|_{z=H(r,\theta)} \;] \tag{32}$$

The above expression only involves Z_n and not their horizontal derivatives in the integrands. Therefore, the corresponding numerical evaluations are more efficient.

The last two terms of Eq. (32) express the direct contribution of bathymetry change and sediment properties in $\vec{A}_{mn}$. They were excluded in the previous model but are included in the latest version.

b. Case of m=n

In order to derive an expression for the vector mode coupling coefficient for $m=n$, we differentiate the orthonormal condition Eq. (4) using *Leibniz rule*. The result is

$$\vec{A}_{nn}(r,\theta) = \beta_{nn}\hat{r} + \gamma_{nn}\hat{\theta} = - \nabla_h H(r,\theta) \left(\frac{1}{\rho_1}-\frac{1}{\rho_2}\right) Z_{1n}^{2}\Big|_{z=H(r,\theta)} \tag{33}$$

Note that, this coupling coefficient is zero for a flat horizontal bottom. The latest version of the model has included this new term.

2. SCALAR MODE COUPLING COEFFICIENT, B_{mn}

Taking the horizontal gradient of both sides of Eq. (21), i.e., definition of the vector mode coupling coefficient, and applying the *Leibniz rule* for differentiation of a definite integral, we get after some manipulations, the following expression for the scalar mode coupling coefficient:

$$B_{mn}(r,\theta) = \frac{1}{2}\nabla_h \vec{A}_{mn}(r,\theta) - \int_0^\infty \frac{1}{\rho} \nabla_h Z_m . \nabla_h Z_n \, dz - \tag{34}$$
$$- \left(\frac{1}{\rho_1} - \frac{1}{\rho_2}\right) Z_{1m} \nabla_h Z_{1n} . \nabla_h H(r,\theta) \Big|_{z=H(r,\theta)}$$

There is a unique property associated with a complete set of orthonormal functions, called the *"closure relationship."* For the normal modes, this relationship can be expressed as

$$\sum_n \frac{1}{\rho} Z_n(z;r,\theta) \; Z_n(z';r,\theta) = \delta(z-z') \tag{35}$$

Taking the horizontal gradient of both sides of the closure relationship, multiplying by $Z_m(z;r,\theta)$ and then integrating over the entire depth, we get, after some rearranging of terms,

$$\nabla_h Z_m(z;r,\theta) = - \sum_n \vec{E}_{mn}(r,\theta) Z_n(z;r,\theta) \tag{36}$$

where

$$\vec{E}_{mn}(r,\theta) = \frac{1}{2}\vec{A}_{mn}(r,\theta) + \nabla_h H(r,\theta)\left(\frac{1}{\rho_1}-\frac{1}{\rho_2}\right) Z_{1n}(H;r,\theta)\, Z_{1m}(H;r,\theta) \tag{37}$$

Substituting now Eq. (36) in Eq. (34), we finally obtain the following alternative expression for B_{mn}:

$$B_{mn} = \frac{1}{2}\nabla_h \vec{A}_{mn}(r,\theta) - \sum_{l} \vec{E}_{nl}(r,\theta) . \vec{E}_{ml}(r,\theta) - \left(\frac{1}{\rho_1}-\frac{1}{\rho_2}\right) Z_{1m}(z;r,\theta)\nabla_h Z_{1n}(z;r,\theta) . \nabla_h H(r,\theta) \Big|_{z=H(r,\theta)} \tag{38}$$

Eq. (38) is valid for both the $m \neq n$ and $m=n$ cases. The last term of Eq. (38), is new in the model. The magnitude of this mode coupling coefficient is generally much less than the magnitude of the vector coefficient.

C. NUMERICAL IMPLEMENTATION

The major part of the model upgrade was the replacement of the old routines with new ones for the computations of the rederived mode coupling coefficients according to Eqs. (32), (33) and (38).

These new routines are contained in a program called *"sedbot"* and are listed in Appendix C. Normal modes and the horizontal gradient vector of the wavenumber as function of position are required as input to *"sedbot."*

The normal mode field is created by a program called *"modes"*, whereas the horizontal gradients of the wavenumber are calculated in the program *"kder."* These two programs are listed in Appendices A and B, respectively.

D. EXAMPLE RUNS

For both example runs, the medium is taken to have two isospeed, isodensity layers separated by a constant-slope interface. Sound speed is taken to be 1500 m/sec in the water column and 3000 m/sec in the sediment. Density is taken to be 1000 Kg/m^3 in the water column and 2500 Kg/m^3 in the sediment. The source is taken to be harmonic in time, with a frequency of 100 Hz, and is located at a depth of 50 m. The coupled mode solution along two radii, 90° and 45°, are displayed and discussed.

1. BOTTOM SLOPE = .001 RADIANS

The bottom slope in this first case is taken to be .001 radians. The water depth is 100 m at the source location and 70 m after 30 km in the y direction (see Fig. 2).

At the source location there are twelve trapped modes in the water layer. Only eight trapped modes exist at the location 30 km upslope.

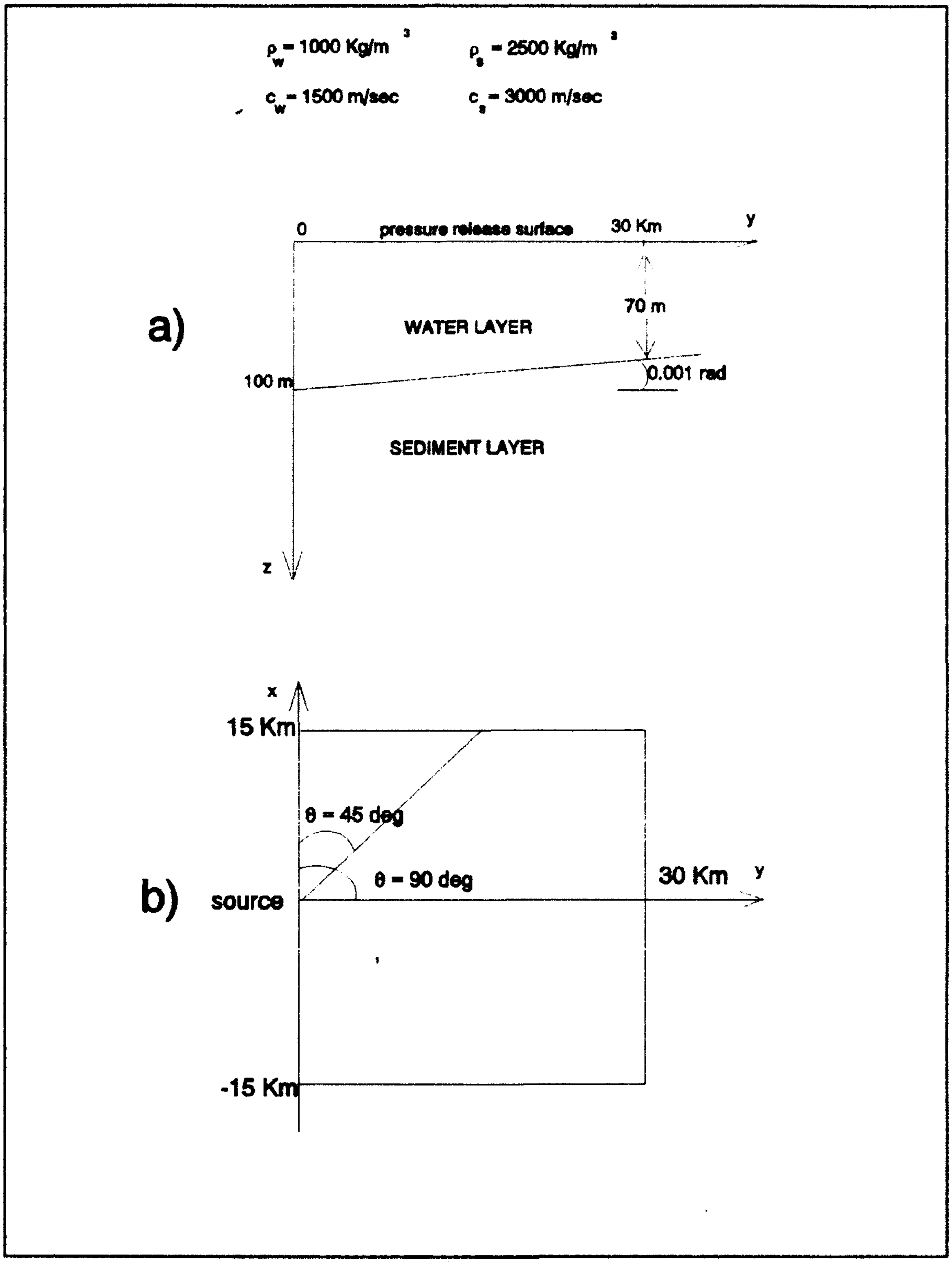

Figure 2. Geometry of the first example case with a constant slope of .001 radians along y-axis (a), and a plane view showing the $\theta = 90°$ and $\theta = 45°$ propagation paths (b)

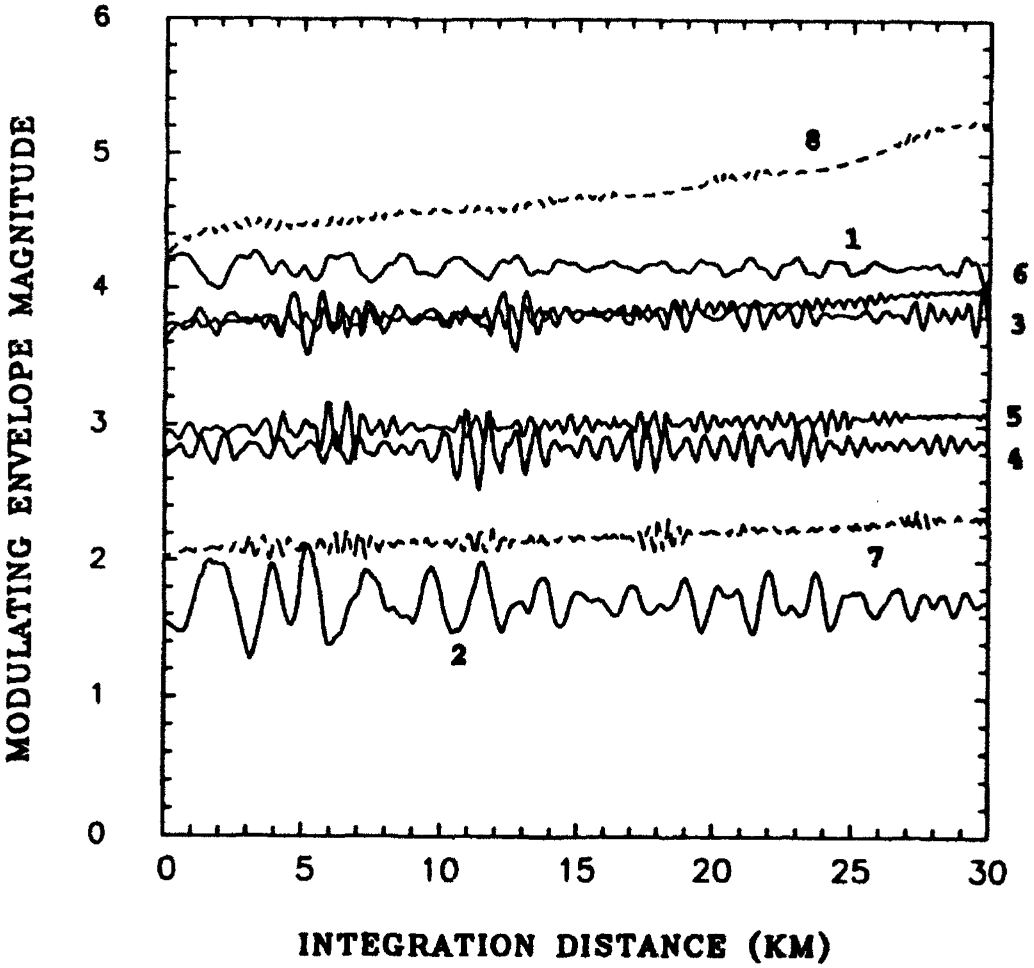

Figure 3. Envelope amplitudes of the first eight trapped modes in the 3D coupled mode solution along the path $\theta = 90°$ for a bottom slope of .001 radians

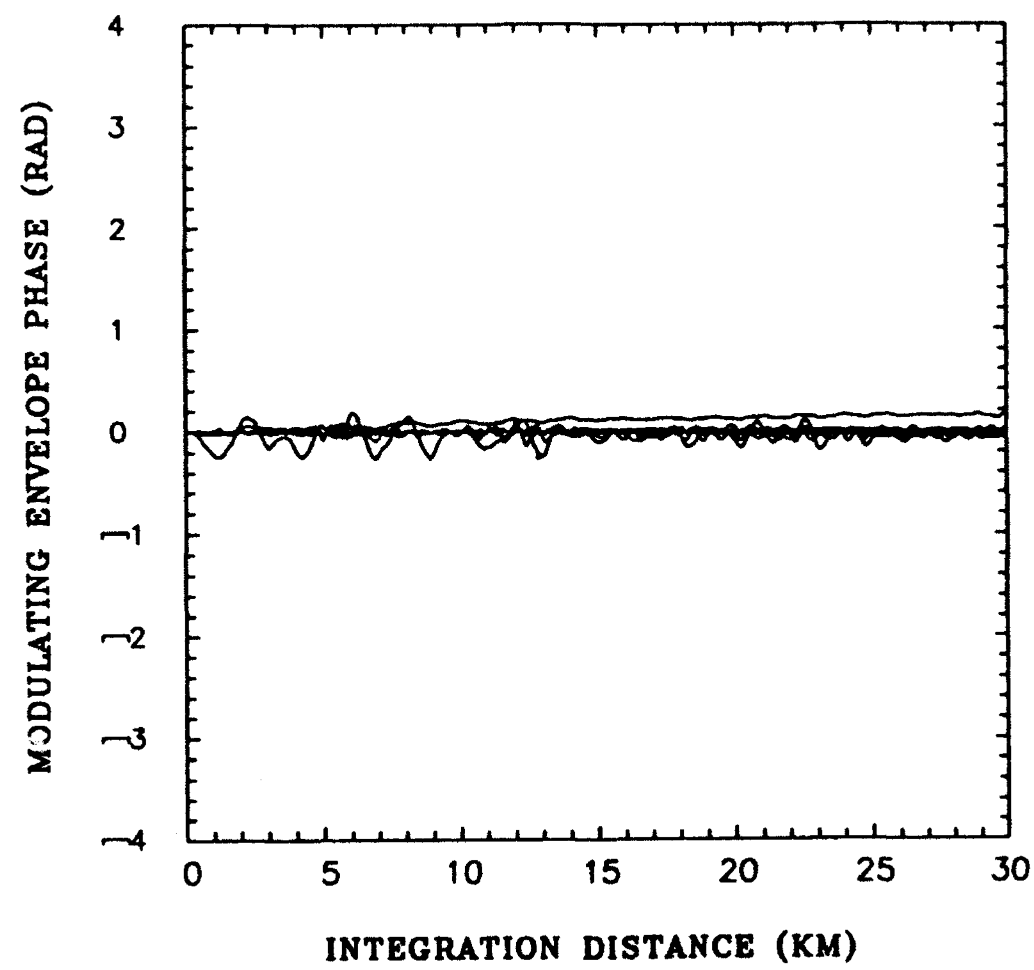

Figure 4. Envelope phases of the first eight trapped modes in the 3D coupled mode solution along the path $\theta = 90°$ for a bottom slope of .001 radians

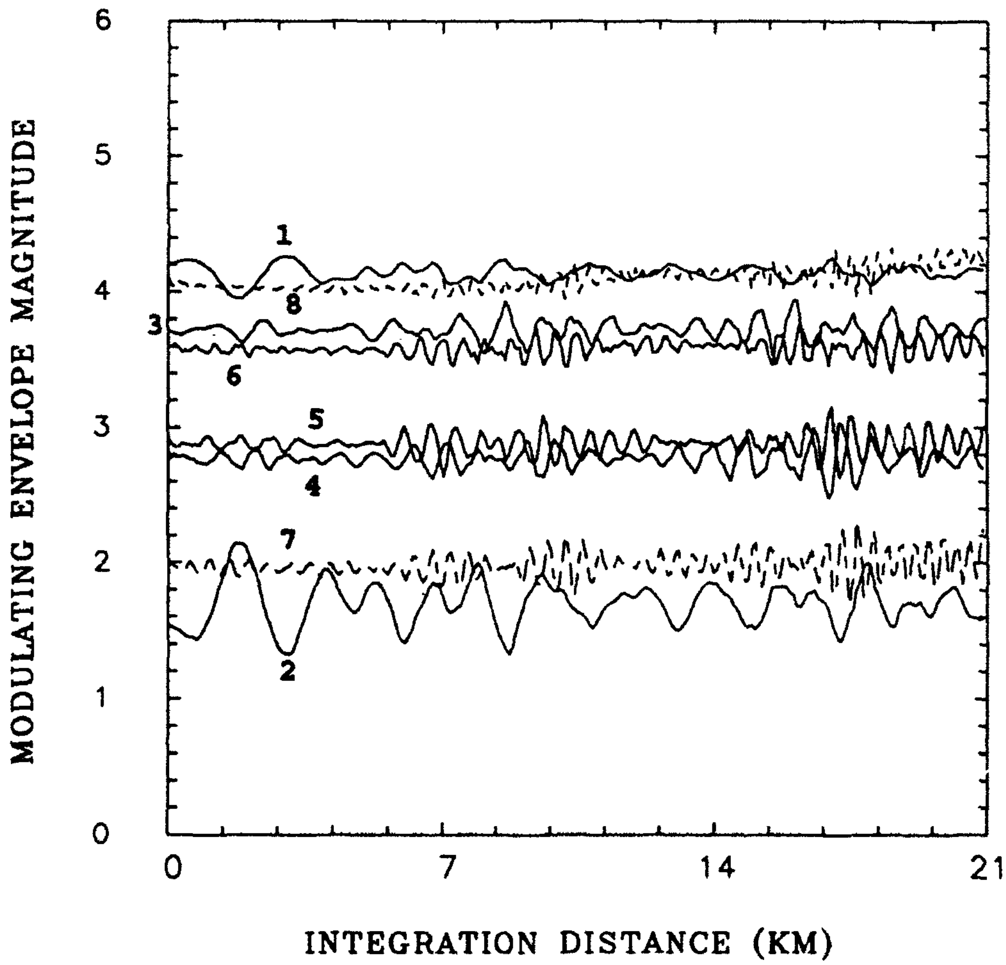

Figure 5. Envelope amplitudes of the first eight trapped modes in the 3D coupled mode solution along the path $\theta = 45°$ for a bottom slope of .001 radians

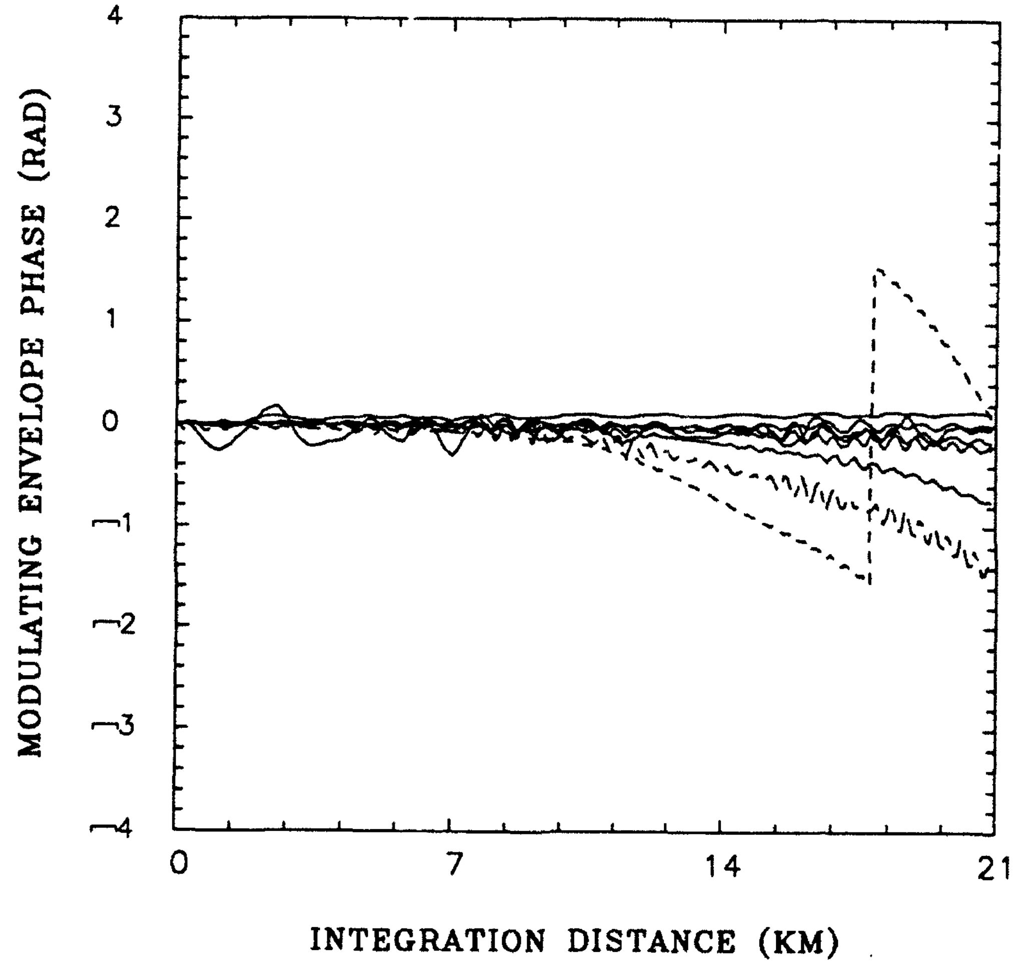

Figure 6. Envelope phases of the first eight trapped modes in the 3D coupled mode solution along the path $\theta = 45°$ for a bottom slope of .001 radians

Fig. 3 and Fig. 4 show the amplitude and phase of the modulation envelope, U_n, for the first eight modes travelling in the upslope y direction, i.e., along the path $\theta = 90°$ (see Fig. 2). An upslope enhancement is noticed, especially for the higher order modes, as they propagate into shallower water. The phase of the envelope, which is the phase deviation from the 2D adiabatic approximation, is very small (about 11° maximum). The amplitude fluctuations are between 15% and 30% for all the modes. In light of the small amplitude and phase fluctuations, the adiabatic approximation can be considered reasonable along this propagation path.

Fig. 5 and Fig. 6 show the amplitude and phase of the modulation envelope, U_n, for the first eight modes, along the propagation path $\theta = 45°$ (see Fig. 2). Here, the upslope enhancement is significantly less and the fluctuations of the amplitude of the higher order modes at greater range are slightly larger than along the previous path. We speculate that this slight increase in the fluctuations is due to that more interacting modes remain trapped in the water column at longer ranges along this path. The higher order modes have large phase deviations from the 2D adiabatic phases. These large phase changes correspond to significant horizontal refraction of the wave fronts due to the existence of a transverse gradient in the bottom bathymetry.

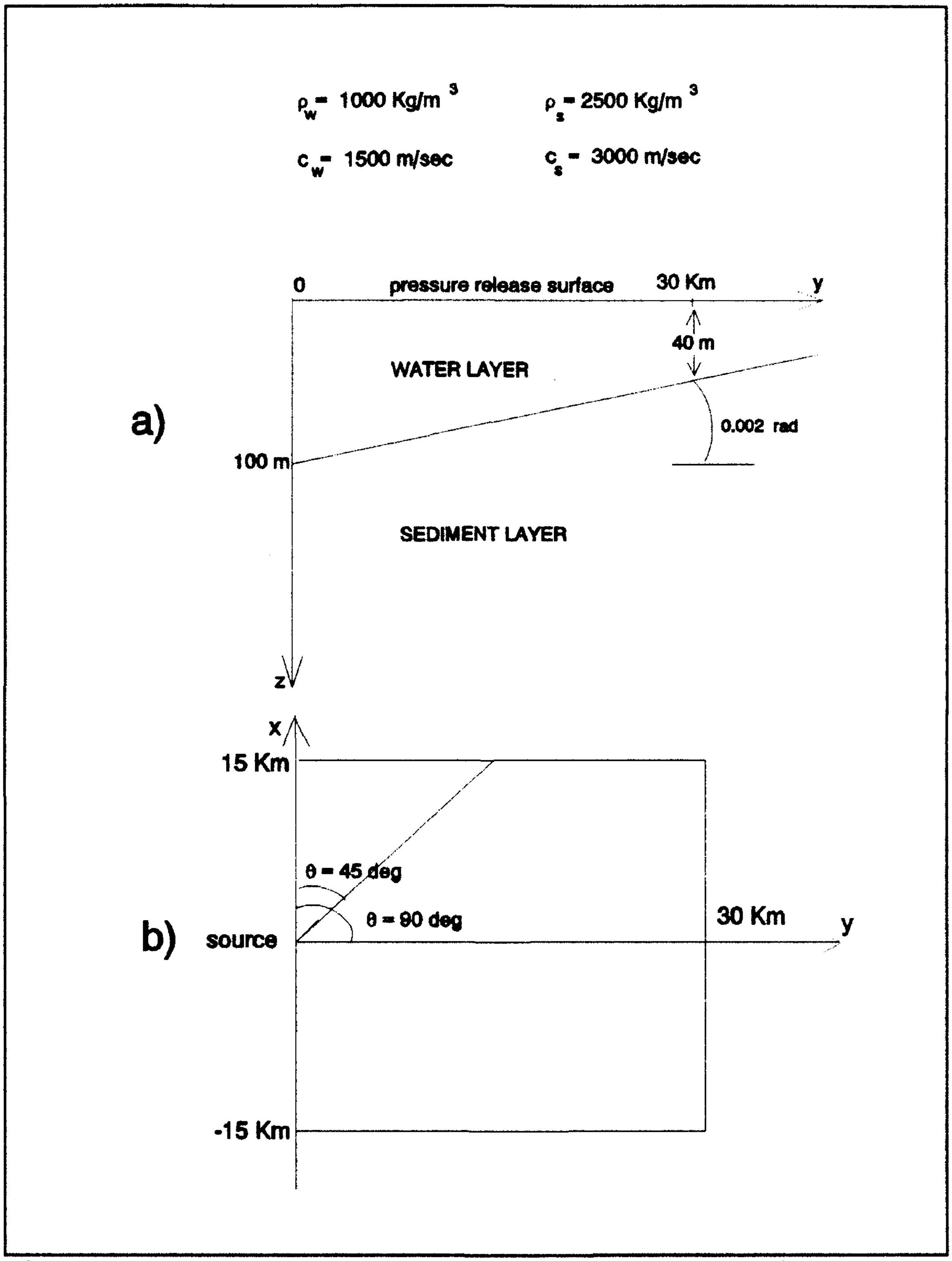

Figure 7. Geometry of the second example case with a constant slope of .002 radians along y-axis (a), and a plane view showing the $\theta = 90°$ and $\theta = 45°$ propagation paths (b)

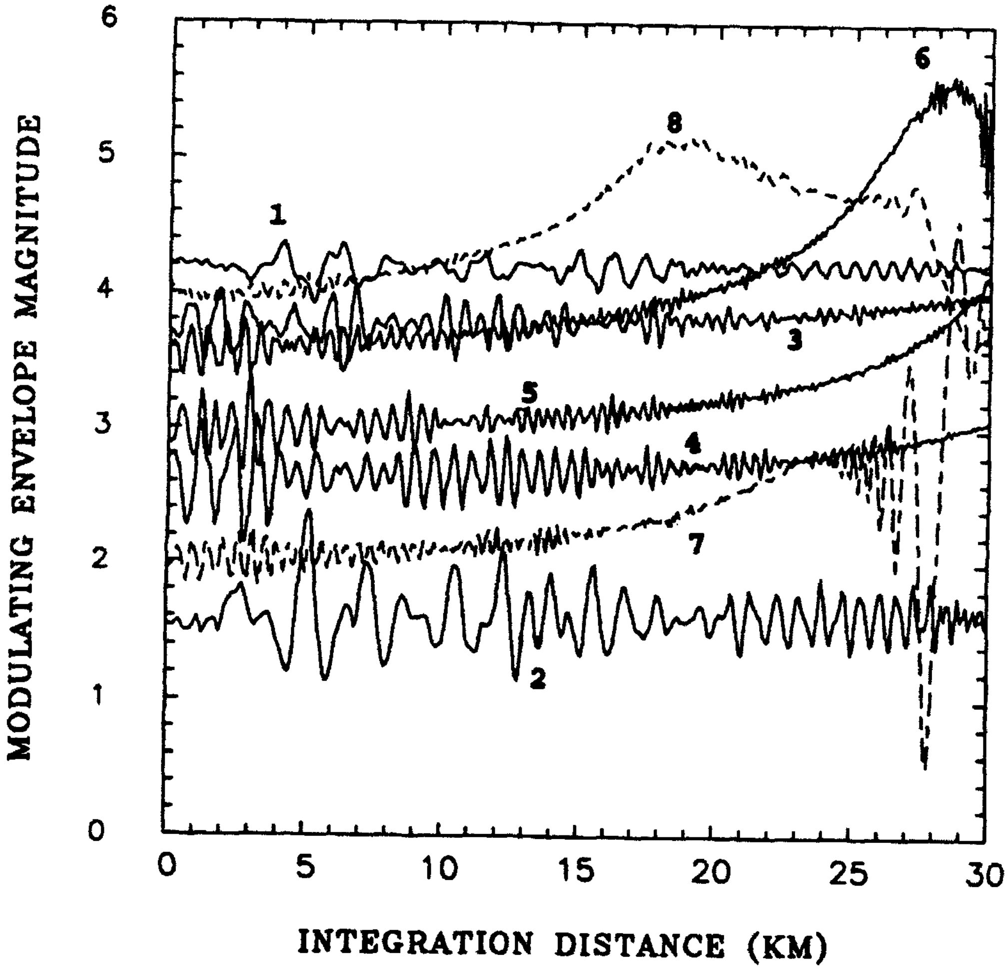

Figure 8. Envelope amplitudes of the first eight trapped modes in the 3D coupled mode solution along the path $\theta = 90°$ for a bottom slope of .002 radians

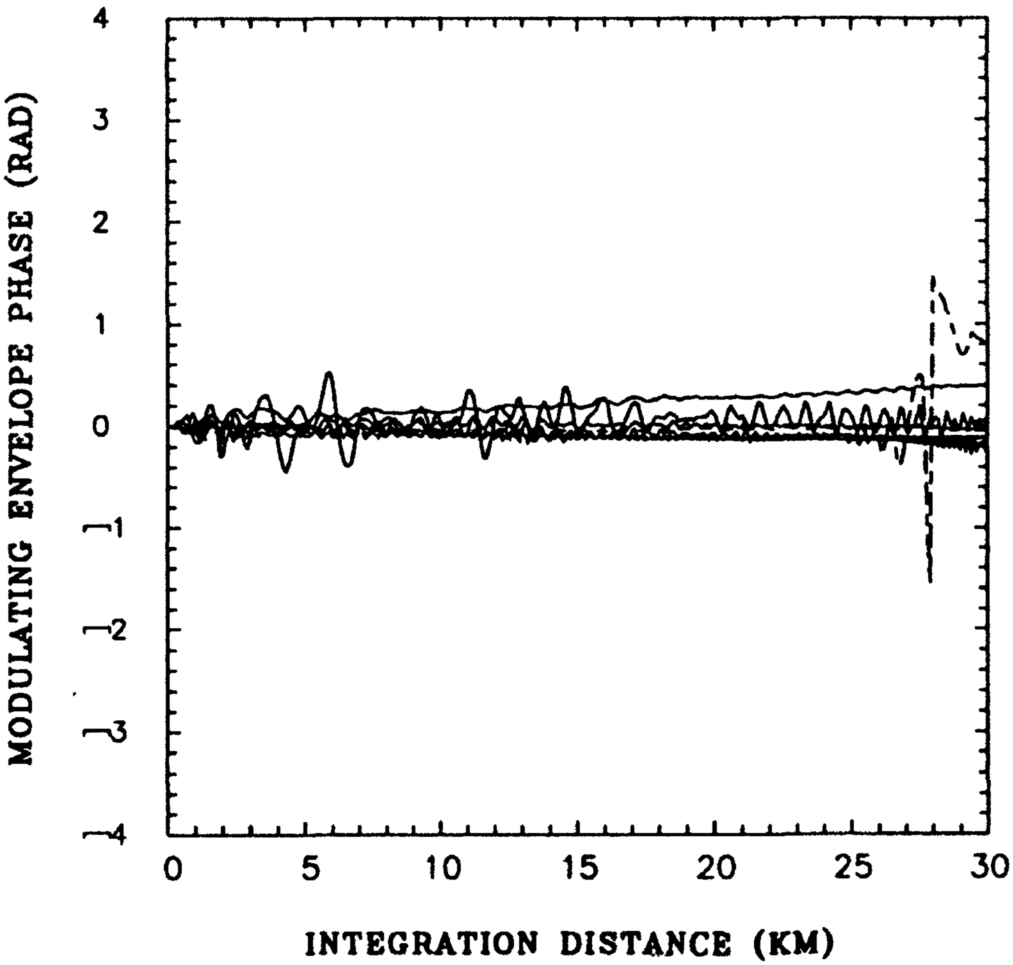

Figure 9. Envelope phases of the first eight trapped modes in the 3D coupled mode solution along the path $\theta = 90°$ for a bottom slope of .002 radians

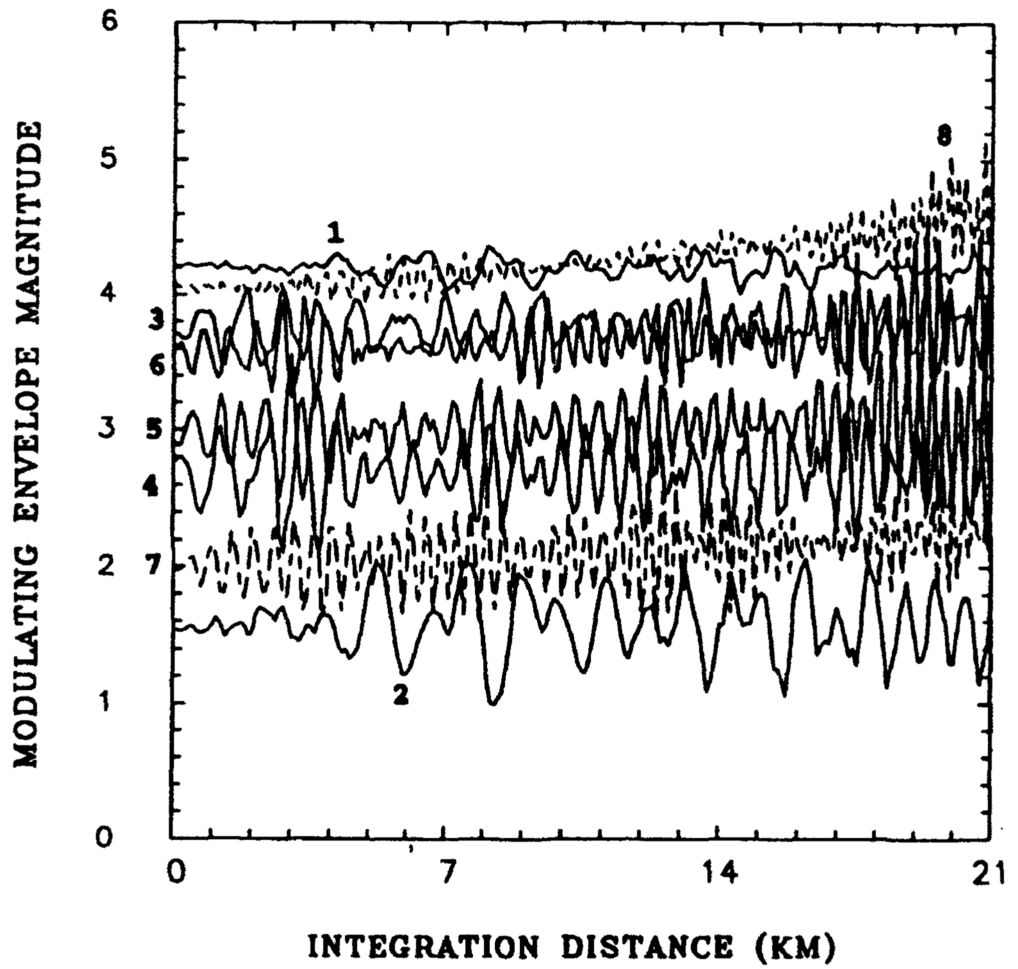

Figure 10. Envelope amplitudes of the first eight trapped modes in the 3D coupled mode solution along the path $\theta = 45°$ for a bottom slope of .002 radians

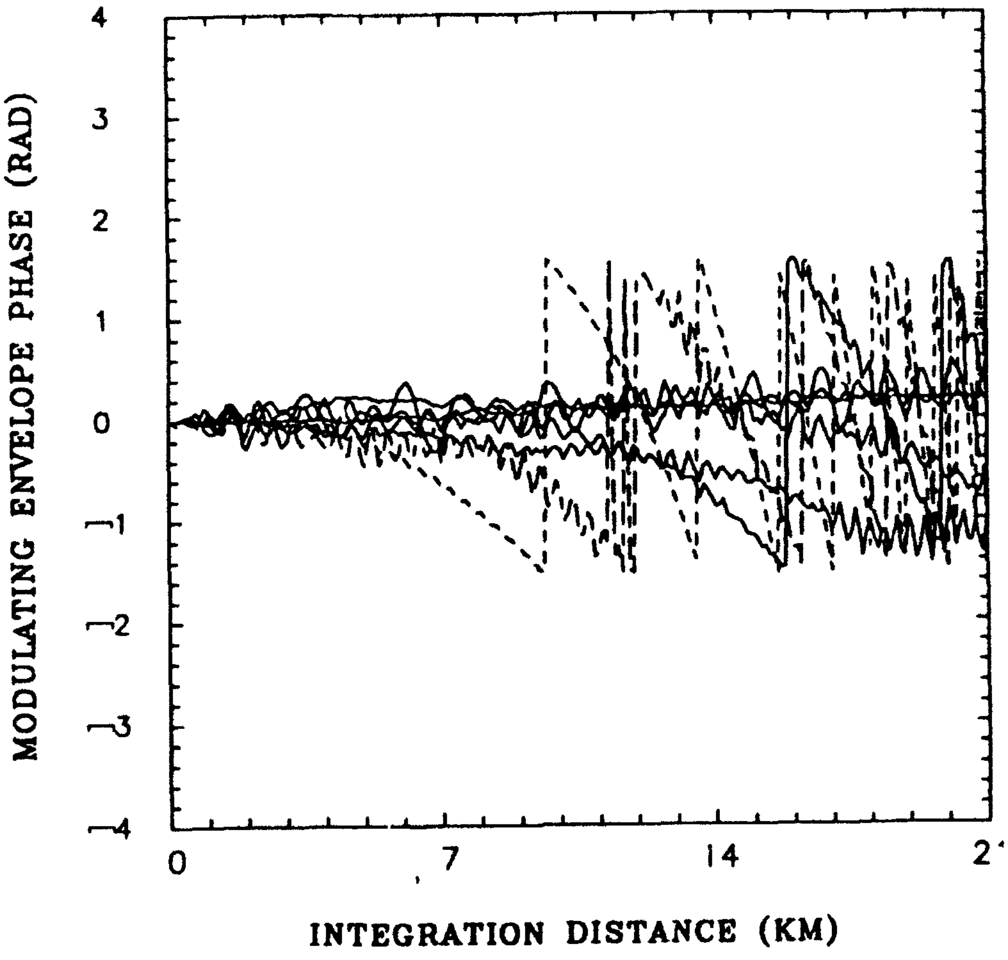

Figure 11. Envelope phases of the first eight trapped modes in the 3D coupled mode solution along the path $\theta = 45°$ for a bottom slope of .002 radians

2. BOTTOM SLOPE = .002 RADIANS

For this case, the same isospeed, isodensity, wedge shape waveguide is used, except the bottom slope is now doubled (.002 radians). The bottom depth at the source location is now 100 m and shoals to 40 m after 30 km away from the source in the y direction (see Fig. 7). At the source position there are twelve trapped modes, but in 30 km upslope, there are only five trapped modes in the water column.

Fig. 8 and Fig. 9 show the amplitudes and phases of the modulation envelope, U_n, for the first eight trapped modes travelling upslope in the y-axis direction, i.e. along the path $\theta = 90°$ (see Fig. 9). Upslope enhancement is much stronger than the previous case, especially for the higher modes. The fluctuations in amplitude is about 50% in some modes and in phase more than 20°. Thus, the adiabatic approximation would induce considerably larger errors than the case of a .001 bottom slope.

Fig. 10 and Fig. 11 show the amplitudes and phases of the modulation envelope, U_n, for the same eight modes along the propagation path, i.e., $\theta = 45°$ (see Fig. 7). The horizontal refraction phenomenon is much stronger here than for the case of a .001 slope. Along this path, the adiabatic approximation would also induce large errors. Typical percentages of amplitude fluctuations are about 50% for the second mode and

30% for the third and fourth modes. The phase deviation, especially for the higher order modes, is also large.

IV. CONCLUSIONS AND RECOMMENDATIONS

A 3D coupled normal mode model for sound propagation in shallow water with irregular bottom bathymetry, is developed in this thesis. This model can be used to examine underwater sound propagation involving significant bottom interaction. In this model, sound speed is allow to vary in three dimensions and water depth and sediment properties in horizontal location.

It is shown here that, for a frequency of 100 Hz, the adiabatic approximation is valid only for very mild bottom slopes. Typical errors for a slope of .001 radians are 15% in mode amplitude and 10° in its phase. For a slope of .002 radians, the errors are significantly larger.

The model presented in this thesis is capable of simulating the interactions of the normal modes as they propagate in complex environments. Propagation phenomena like mode-mode interaction, horizontal refraction and slope enhancement can be examined using this improved model.

In the development of the present model an approximation (Eq. 12) in the bottom boundary conditions is used. The validity of this approximation requires that the slope must be much smaller than unity (Eq. 13). In order to be able to handle very steep bottom slopes, i.e., order one slopes, one needs to use the exact form of the bottom boundary conditions

(Eq. 10). This could make the formulation more complicated but it should be tractable.

Another future improvement to the model will be to include sound energy absorption (attenuation). One way to do this is by introducing imaginary parts in the eigenvalues (wavenumbers). Lastly, a test for the accuracy of the improved model is needed. This can be achieved by comparing the results generated by this model with some exact analytic solutions.

APPENDIX A. FORTRAN ROUTINES FOR COMPUTING NORMAL MODES FIELD

The following program creates the normal mode field, "amode.dat", for a given geographical area of the ocean. Sound speed, density, and bottom depth, are defined for every grid point. Given sound speed field and density, the normal modes are calculated using a standard mode solver routine.

```
***********************************************************
*                                                         *
*   This program computes the  normal modes in a  given   *
*   area.                                                 *
*                                                         *
***********************************************************
c
c    INPUT/ARGUMENTS
c    f                  frequency, Hz
c    xmax               maximum position in x direction, meters
c    ymax               maximum position in y direction, meters
c    h(nx,ny)           depth at x,y position, meters
c    nx                 number of stations in x direction
c    ny                 number of stations in y direction
c    nz                 number of stations in z direction
c    xo                 initial x position, meters
c    yo                 initial y position, meters
c    dz                 step size in z direction, meters
c    cf(nx,ny,nz)       sound speed field in every x,y position,
c                                                            m/s
c    df(nx,ny,nz)       density field in every x,y position,
c                                                       kgr/m^3
c    isw                switch index : 1 write / 0 do not write
c    mm                 maximum number of allowable trapped
c                                                          modes
c
c    OUTPUT/ARGUMENTS
c    for each horizontal station x,y:
c    ksq_r(nz)          squared eigenvalues for each trapping
c                                                    mode (real)
c    efun_r(nz,nz)      eigenfunctions (real)
c    h(nx,ny)           water depth (meters)
c    c(nz)              sound speed profile in a specific grid
```

```
c                                                           position
c   d(nz)              density profile in a specific grid
c                                                           position
c   w                  source angular frequency (rad/sec)
c   dx,dy,dz          step size in x,y and z directions (meters)
c   nx,ny,nz          number of stations in x,y and z directions
c                                                          (metrers)
c
      program modes
c
      parameter(xmax=30000.d0,ymax=30000.d0,nx=11,ny=11,
     *                                           nz=100,mm=20)
      implicit real*8 (a-h,o-z)
      real*8 cf(nx,ny,nz),c(nz),df(nx,ny,nz),d(nz)
      real*8 h(nx,ny),ksq_r(nz),ksq_i(nz)
      real*8 efun_r(nz,nz),efun_i(nz,nz),efun(nz,mm)
      real*8 ks(mm),x(nx),y(ny),z(nz),ksed,kwat
      logical ex
c
      data isw /1/
c
      inquire(file='amode.dat',exist=ex)
      if (ex) then
         open(unit=13,file='amode.dat',status='old')
         close(13,status='delete')
      endif
      open(unit=4,file='/home/noise/sagos/modes/amode.dat',
     *     form='unformatted',status='new')
      inquire(file='mode.sys',exist=ex)
      if (ex) then
         open(unit=13,file='mode.sys',status='old')
         close(13,status='delete')
      endif
      open(unit=6,file='/home/noise/sagos/modes/mode.sys',
     *     form='formatted',status='new')
c
      write(6,*)'output field'
      dx=xmax/dfloat(nx-1)
      dy=ymax/dfloat(ny-1)
      dz=2.d0
      pi=4.d0*datan(1.d0)
      f=224.d0
      w=2.d0*pi*f
c
      call data(cf,df,h,nx,ny,nz,dx,dy,dz,xo,yo,x,y,z)
c
      nm=nz-2
      write(4) w,dx,dy,dz
      write(4) nx,ny,nz
c
      if (isw.eq.1) then
```

```
         write(6,*)'dz= ',dz, ' meters'
         write(6,*)'nx= ',nx,' ny= ',ny,' nz= ',nz
         write(6,*)'interface depth'
         write(6,*)'h= ',h(1,1) ,' meters'
         write(6,*)'sound speed profile, m/s'
         write(6,*)(cf(1,1,iz),iz=1,nz,2)
         write(6,*)'density profile, kgr/m^3'
         write(6,*)(df(1,1,iz),iz=1,nz,2)
      endif
c
      mcntr=0
c
      do ix=1,nx
         do iy=1,ny
      icounter=0
      ichk=0
      if (ix.eq.1.and.iy.eq.1) ichk=4
             write(4) h(ix,iy)
             do iz=1,nz
                c(iz)=cf(ix,iy,iz)
                d(iz)=df(ix,iy,iz)
             enddo
             write(4) c
             write(4) d
             write(6,*)'ix=',ix,' iy=',iy,' ichk=',ichk
             call mode(f,nz,dz,c,d,nm,ksq_r,ksq_i,
     *                  efun_r,efun_i,ichk)
c
c    choose only the trapped modes
c    cs : sound speed in the sediment (constant)
c    cw : sound spedd in the water column,next to the
c         interface
c
             cs=c(int(h(ix,iy)/dz)+2)
             cw=c(int(h(ix,iy)/dz)-1)
c
c    set zeros in the eigenvalues-eigenfunctions arrays
c
             do i=1,mm
                ks(i)=0.d0
                do j=1,nz
                   efun(j,i)=0.d0
                enddo
             enddo
c
             do i=1,nm
                ksed=(w/cs)**2
                kwat=(w/cw)**2
                if (ksed.lt.ksq_r(i).and.ksq_r(i).lt.kwat)
     *                                                         then
                   icounter=icounter+1
```

```
                 ks(icounter)=ksq_r(i)
                 do iz=1,nz
                    efun(iz,icounter)=efun_r(iz,i)
                 enddo
              endif
           enddo
c
           if (isw.eq.1) then
               write(6,*)'h= ',h(ix,iy)
               write(6,*)'limits for trapped modes : '
               write(6,*)kwat,ksed
               write(6,*)'icounter= ',icounter
               write(6,*)'kn^2'
               write(6,*)(ks(i),i=1,mm)
           endif
c
           if (ix.eq.3.and.iy.eq.3) then
               write(6,*)'efun(iz,18)'
               write(6,*)(efun(iz,18),iz=1,nz)
           endif
c
           write(4) icounter
           if (icounter.gt.mm.and.icounter.gt.mcntr)
     *                                  mcntr=icounter
           write(4) ks
           write(4) efun
        enddo
      enddo
c
      if (mcntr.ne.0) write(6,1001)mcntr
 1001 format(i3,' trapped modes, exceeds limit, increase mm
c                                              and rerun')
      close(4)
      close(6)
      end


************************************************************
*                                                          *
*  The following program provides an example data input.   *
*                                                          *
************************************************************
c
c    INPUT/ARGUMENTS
c    nx               number of stations in x direction
c    ny               number of stations in y direction
c    nz               number of stations in z direction
c    dx               step size in x direction, meters
c    dy               step size in y direction, meters
c    dz               step size in z direction, meters
```

```
c    xo                initial x position, meters
c    yo                initial y position, meters
c
c    OUTPUT/ARGUMENTS
c    cf(nx,ny,nz)      sound speed field in every x,y position,
c                                                          m/s
c    df(nx,ny,nz)      density field in every x,y position,
c                                                      kgr/m^3
c    h(nx,ny)          interface depth, meters
c
      subroutine data(cf,df,h,nx,ny,nz,dx,dy,dz,xo,yo,x,y,z)
c
      implicit real*8 (a-h,o-z)
      real*8 cf(nx,ny,nz),df(nx,ny,nz),h(nx,ny)
      real*8 x(nx),y(ny),z(nz)
c
      do ix=1,nx
         do iy=1,ny
            x(ix)=xo+dx*dfloat(ix-1)
            y(iy)=yo+dy*dfloat(iy-1)

c
c    bathymetry field
c
            h(ix,iy)=-.0005d0*x(ix)+100.d0
c
c    sound speed and density fields
c
            do iz=1,nz
               z(iz)=dz*dfloat(iz-1)
               if  (z(iz).le.h(ix,iy)) then
                  cf(ix,iy,iz)=.0005d0*x(ix)-
     *                             .1d0*z(iz)+1490.d0
                   df(ix,iy,iz)=1000.d0
               else
                   cf(ix,iy,iz)=1800.d0
                   df(ix,iy,iz)=2000.d0
               endif
            enddo
         enddo
      enddo
c
      return
      end
```

APPENDIX B. FORTRAN ROUTINE FOR COMPUTING WAVENUMBER DERIVATIVES

This program inputs from "amode.dat" as created by the previous program *"modes"*. Computes the horizontal derivatives of the total wavenumber k, at every position of the acoustic field. The derivative calculation requires definition of a computational domain. Output is to the file "kder.dat".

```
*************************************************************
*                                                           *
* This program assigns the source position relative to     *
* the input field via ixorig and iyorig.                    *
* This program also specifies the radial increment for the *
* spline definition, the number of intervals, and the      *
* angular increment  between integration paths (dr,nr,da)  *
*                                                           *
* Procedure: xy-spline at each depth                        *
*            evaluate dk/dx,dk/dy                           *
*            transform into dk/dr,dk/da                     *
*                                                           *
*************************************************************
c
      program kder
c
      parameter(nx=11,ny=11,nz=100,nm=20,ndum=nz*nm,
     *         ixorig=1,iyorig=3,nwk=2*ny*nx+2*max0(nx,ny))
      implicit real*8 (a-h,o-z)
      real*8 kd(nx,ny,nz,2),k(nx,ny),kdxy(6),kc(2,nx,2,ny)
      real*8 x(nx),y(ny),ang(nx,ny),c(nz),ct(nx,ny,nz)
      real*8 wk(nwk),efun(ndum),hork(nz)
      character*20 filename
      logical ex
c
c---------------------------------------------
c open statements
c---------------------------------------------
c
        inquire(file='kder.dat',exist=ex)
        if (ex) then
           open(unit=13,file='kder.dat',status='old')
           close(13,status='delete')
        endif
        inquire(file='kder.sys',exist=ex)
```

```
          if (ex) then
             open(unit=13,FILE='kder.sys',status='old')
             close(13,status='delete')
          endif
          inquire(file='efun_orig.dat',exist=ex)
          if (ex) then
             open(unit=13,file='efun_orig.dat',status='old')
             close(13,status='delete')
          endif
             open(unit=4,file='amode.dat',form='unformatted',
     *       status='old',err=2001)
             open(unit=14,file='kder.dat',form='unformatted',
     *       status='new',err=2002)
             open(unit=24,file='efun_orig.dat',form=
     *       'unformatted',status='new',err=2003)
             open(unit=6,file='kder.sys',form='formatted',
     *       status='new',err=2004)
c
c input w (rad/sec), dx,dy (meters), dz (meters)
      read(4) w,dx,dy,dz
c
c input: number of x indices, no. y indices
c        number of modes, TOTAL vertical increments
      read(4) nxt,nyt,nzt
c
      write(6,1009) w
      write(6,*)' dx(m),dy(m),dz(m)'
      write(6,*) dx,dy,dz
      if (m.ne.nm) write(6,*)'m=',m,'  nm=',nm
      if (nx.ne.nxt) write(6,*)'nx=',nx,'  nxt=',nxt
      if (ny.ne.nyt) write(6,*)'ny=',ny,'  nyt=',nyt
      if (nzp1.ne.nzt) write(6,*)'nzp1=',nzp1,'  nzt=',nzt
c
      pi = dacos(-1.d0)
c
c -----------distribution parameters for spline-----------
      dda = .8d0
      da = dda*pi/180.
      dr = 3000.d0
c number of points in spline
      nr = (nx-ixorig)*dx/dr +1
      write(6,*) nr,' spline locations with interval=',dr
c number of radial paths
      tang = datan( .5*dy*(ny-1) / (dx*(nx-1)) )
c      na = 2*idint(tang/da) + 1
      na=3
      write(6,*) na,' integration paths for da=',dda,' deg'
c-----------------------------------------------------------
c
c horizontal field grid in meters
      do 11 ix=1,nx
```

```
   11 x(ix) = dfloat(ix-ixorig)*dx
      do 12 iy=1,ny
   12 y(iy) = dfloat(iy-iyorig)*dy
      write(6,*)'x range: (',x(1),',',x(nx),')'
      write(6,*)'y range: (',y(1),',',y(ny),')'
c
c read in c-field and
c calculate angle (ccw from x-axis)
      do 14 ix=1,nx
      do 14 iy=1,ny
c full sound speed profile(nzp1) (0-5000m)
      read(4) h
      read(4) c
      read(4) dens
      read(4) icounter
      read(4) hork
      read(4) efun
c create file to obtain initial conditions
      if (iy.eq.iyorig.and.ix.eq.ixorig) write(24)efun
      do iz=1,nz
      ct(ix,iy,iz) = c(iz)
      enddo
      if (ix.eq.ixorig) then
      if (iy.ge.iyorig) ang(ix,iy) = pi/2.
      if (iy.lt.iyorig) ang(ix,iy) = -pi/2.
      else
      ang(ix,iy) = datan((y(iy))/(x(ix)))
      endif
   14 continue
c
c calculate derivatives from first station below surface to
c bottom
c
      do 100 iz=1,nz
         do 110 ix=1,nx
         do 110 iy=1,ny
c
c wavenumber is in rad/m
c
  110    k(ix,iy) = w/ct(ix,iy,iz)
c
c fit bi-cubic spline to iz-th level waveno.
c
      ic = nx
      call ibcccu(k,x,nx,y,ny,kc,ic,wk,ier)
      if (ier.ne.0) write(6,1001) ier
c
c use spline to evaluate cartesian derivatives at each grid
c point
c transform derivatives into cylindrical coordinates
c
```

```
         do 120 ix=1,nx
            do 121 iy=1,ny
            call dbcevl(x,nx,y,ny,kc,ic,x(ix),y(iy),kdxy,ier)
            if (ier.ne.0)
     *          write(6,1007) ix,iy,X(ix),Y(iy),ier
            call cyl(ang(ix,iy),kdxy(2),kdxy(3),
     *                    kd(ix,iy,iz,1),kd(ix,iy,iz,2))
  121       continue
  120    continue
       if (iz.eq.1) then
          write(6,*)' For first subsurface layer'
          write(6,1005)
          write(6,1006) ((ct(ix,iy,1),ix=1,nx),iy=ny,1,-1)
          write(6,1011)
          write(6,1016) ((k(ix,iy),ix=1,nx),iy=ny,1,-1)
          write(6,1003)
          write(6,1002) ((kd(ix,iy,1,1),ix=1,nx),iy=ny,1,-1)
          write(6,1004)
          write(6,1002) ((kd(ix,iy,1,2),ix=1,nx),iy=ny,1,-1)
          write(6,*)
       endif
c
  100 continue
c
      write(14) ixorig,iyorig
      write(14) da,na,dr,nr
      write(6,*) 'da,na,dr,nr',da,na,dr,nr
      write(14) kd
c
      write(6,*)' For bottom level'
      write(6,1005)
      write(6,1006) ((ct(ix,iy,nz),ix=1,nx),iy=ny,1,-1)
      write(6,1011)
      write(6,1016) ((k(ix,iy),ix=1,nx),iy=ny,1,-1)
      write(6,1003)
      write(6,1002) ((kd(ix,iy,nz,1),ix=1,nx),iy=ny,1,-1)
      write(6,1004)
      write(6,1002) ((kd(ix,iy,nz,2),ix=1,nx),iy=ny,1,-1)
      write(6,*)
      write(6,1008) (kd(4,3,iz,1),iz=1,nz,2),kd(4,3,nz,1)
     goto 2020
c
 1001 format(' ier:',i3,' for ibcccu, xy-spline')
 1002 format(5(1x,e12.5))
 1003 format(' dk/dr   ( (rad/m)   /  m )')
 1004 format(' dk/rda   ( (rad/m)   /  m )')
 1005 format(' c  (m/s)')
 1006 format(5(2x,f8.3))
 1016 format(5(2x,f8.5))
 1007 format(' ix,iy,X,Y,ier for dbcevl: ',2I3,2F6.1,i3)
 1008 format(' at ix,iy=4,3   dk/dr(z)'/11(5(1x,d11.5)/) )
```

```
 1009 format(' frequency,rad/sec :',d14.7)
 1010 format(' k'/7(4(3x,d11.4)/3x,3(3x,d11.4)/) )
 1011 format(' k   (rad/m)')
c
c--------------------------------------
c close statements
c--------------------------------------
c
2001   filename='amode.dat'
       goto 2010
2002   filename='kder.dat'
       goto 2010
2003   filename='efun_orig.dat'
       goto 2010
2004   filename='kder.sys'
2010   write(*,2011) filename
2011   format(' ERROR OPENING FILE : ',A)
2020   close(4)
       close(14)
       close(6)
       end


       subroutine cyl(ang,x,y,r,a)
c   polar transformation subroutine
       implicit real*8 (a-h,o-z)
       r = x*dcos(ang) + y*dsin(ang)
       a = x*dsin(ang) - y*dcos(ang)
       return
       end
```

APPENDIX C. FORTRAN ROUTINES FOR COMPUTING THE COUPLING COEFFICIENTS

```
***********************************************************
*                                                         *
*  This program manages the subroutines "sub1.f", "sub2.f",*
*  "sub3.f", "partial.f" which compute the two mode       *
*  coupling coefficients. The input is from "amode.dat"   *
*  and "kder.dat", specifically the modes, horizontal,    *
*  wavenumber, horizontal derivatives of total wavenumber, *
*  bathymetry and density. The output file is "mcoupl.dat".*
*                                                         *
***********************************************************
c
c   rho1 : water column density-constant in depth (kg/m^3)
c   rho2 : sediment density-constant in depth (kg/m^3)
c   dz : vertical step size (m)
c   nm : maximum number of trapped modes,in the water column
c   nx,ny : number of stations in x and y directions
c   h  : bottom bathymetry
c   zb : acoustic pressure eigenfunctions,at the interface
c                                                    depth
c   zbm1 : acoustic pressure eigenfuctions,one step size
c                                   above the interface depth
c   zbm : zb for the mth mode
c   zbmm1 : zbm1 for the mth mode
c   zbn : zb for the nth mode
c   zbnm1 : zbm1 for the nth mode
c   ar : range component of the first coupling coeff.
c   aa : angle component of the first coupling coeff.
c   cr : correction at the range component of the first
c                                          coupling coeff.
c   ca : correction at the angle component of the first
c                                          coupling coeff.
c   b  : second coupling coeff.
c   k  : square of horizontal wavenumbers(eigenvalues) of
c                                               the modes
c
      program sedbot
c
        parameter(nx=5,ny=5,nz=100,nw=2*nx*ny+2*max(nx,ny),
     *                                                 nm=20)
      implicit real*8 (a-h,o-z)
      real*8 h(nx,ny),zbm(nx,ny),b(nm,nm,nx,ny),km(nx,ny)
      real*8 zbmm1(nx,ny),cr(nm,nm,nx,ny),ca(nm,nm,nx,ny)
      real*8 ar(nm,nm,nx,ny),aa(nm,nm,nx,ny),kn(nx,ny)
      real*8 zbn(nx,ny),zbnm1(nx,ny),c(2,nx,2,ny),wk(nw)
      real*8 cr1(nx,ny),ca1(nx,ny),x(nx),y(ny)
```

```
      real*8 zbnpr(nx,ny),zbnpa(nx,ny),zb(nm,nx,ny)
      real*8 zbnpx(nx,ny),zbnpy(nx,ny),hpr(nx,ny),hpa(nx,ny)
      real*8 hpx(nx,ny),hpy(nx,ny),b1(nx,ny),kk(nm)
      real*8 rho1(nx,ny),rho2(nx,ny),k(nx,ny,nz)
      real*8 cs(nz),d(nz),efun(nz,nm),kd(nx,ny,nz,2)
      integer icounter(nx,ny)
      logical ex
c
      inquire(file='mcoupl.dat',exist=ex)
      if (ex) then
         open(unit=13,file='mcoupl.dat',status='old')
         close(13,status='delete')
      endif
      open(unit=4,file='/home/noise/sagos/modes/amode.dat',
     *     form='unformatted',status='old')
      inquire(file='coupl.sys',exist=ex)
      if (ex) then
         open(unit=13,file='coupl.sys',status='old')
         close(13,status='delete')
      endif
      open(unit=6,file='/home/noise/sagos/modes/coupl.sys',
     *     form='formatted',status='new')
      open(unit=8,file='/home/noise/sagos/modes/mcoupl.dat',
     *     form='unformatted',status='new')
      open(unit=14,file='/home/noise/sagos/modes/kder.dat',
     *     form='unformatted',status='old')
c
      read(4) w,dx,dy,dz
c
      rewind 4
c
      read(14) ixorig,iyorig
      read(14) da,na,dr,nr
      write(8) ixorig,iyorig
      write(8) da,na,dr,nr
      read(14) kd
c
      do ix=1,nx
         x(ix)=dx*dfloat(ix-ixorig)
      enddo
c
      do iy=1,ny
         y(iy)=dy*dfloat(iy-iyorig)
      enddo
c
      write(6,*)'x range: (',x(1),',',x(nx),')'
      write(6,*)'y range: (',y(1),',',y(ny),')'

c
c   computation of first mode coupling coefficients
c
```

```
      do n=1,nm
      do m=1,nm
         read(4) w,dx,dy,dz
         read(4) nxx,nyy,nzz
c
         do ix=1,nx
            do iy=1,ny
               read(4) h(ix,iy)
c
c   i, is the last station in the water column
c
               i=int(h(ix,iy)/dz) + 1
               read(4) cs
c
c   calculate the total wavenumber
c
               do iz=1,nz
                  k(ix,iy,iz)=w/cs(iz)
               enddo
               read(4) d
               rho1(ix,iy)=d(i)
               rho2(ix,iy)=d(i+1)
               read(4) icounter(ix,iy)
               read(4) kk
               km(ix,iy)=kk(m)
               kn(ix,iy)=kk(n)
               read(4) efun
c
               if (m.eq.n) then
                  ar(m,n,ix,iy)=0.d0
                  aa(m,n,ix,iy)=0.d0
               else
c
c   trapezoid integration, to find the integral part of
c   first coefficient
c
                  sumx=0.d0
                  sumy=0.d0
                  denom=km(ix,iy)-kn(ix,iy)
c
                  do iz=1,nz
                     a=k(ix,iy,iz)*efun(iz,n)*
     *                 efun(iz,m)/d(iz)
                     sumx=sumx+a*kd(ix,iy,iz,1)
                     sumy=sumy+a*kd(ix,iy,iz,2)
                  enddo
c
                  if (m.eq.17.and.n.eq.16)
     *            write(6,*)'ix,iy,sumx,sumy',ix,iy,
     *                                                 sumx,sumy
                  ar(m,n,ix,iy)=4.d0*sumx*dz/denom
```

```
                 aa(m,n,ix,iy)=4.d0*sumy*dz/denom
                 if (m.eq.17.and.n.eq.16)
     *           write(6,*)'ix,iy,ar(17,16,ix,iy)',
     *                                ix,iy,ar(17,16,ix,iy)
             endif
c
             zbm(ix,iy)=efun(i,m)
             zbmm1(ix,iy)=efun(i-1,m)
             zbn(ix,iy)=efun(i,n)
             zbnm1(ix,iy)=efun(i-1,n)
          enddo
       enddo
c
       call partial(h,nx,ny,x,y,hpr,hpa,nw,hpx,hpy,c,wk)
c
       call partial(zbn,nx,ny,x,y,zbnpr,zbnpa,nw,
     *                                 zbnpx,zbnpy,c,wk)
c
       if (m.eq.17.and.n.eq.16) then
          write(6,*)'zbn = '
          write(6,1001)((zbn(ix,iy),ix=1,nx),
     *                                      iy=ny,1,-1)
          write(6,*)'zbnpr ='
          write(6,1001)((zbnpr(ix,iy),ix=1,nx),
     *                                      iy=ny,1,-1)
          write(6,*)'zbnpa ='
          write(6,1001)((zbnpa(ix,iy),ix=1,nx),
     *                                      iy=ny,1,-1)
 1001     format(5(2x,e12.4))
       endif
c
       if (m.ne.n) call sub1(rho1,rho2,dz,zbm,zbn,zbmm1,
     *          zbnm1,cr1,ca1,km,kn,m,n,zbnpr,zbnpa,x,y)
c
       if (m.eq.n) call sub2(rho1,rho2,zbm,h,cr1,ca1,
     *                                nx,ny,hpr,hpa,x,y)
c
          do ix=1,nx
             do iy=1,ny
                cr(m,n,ix,iy)=cr1(ix,iy)
                ca(m,n,ix,iy)=ca1(ix,iy)
             enddo
          enddo
c
    if (m.eq.17.and.n.eq.16) then
       write(6,*)'checking quadrature :'
       write(6,*)'ar(,ix,iy)='
       write(6,100)((ar(17,16,ix,iy),ix=1,5),iy=1,5)
       write(6,*)'aa(17,16,ix,iy)='
       write(6,100)((aa(17,16,ix,iy),ix=1,5),iy=1,5)
    endif
```

```
c
         do ix=1,nx
            do iy=1,ny
               ar(m,n,ix,iy)=ar(m,n,ix,iy)+cr(m,n,ix,iy)
               aa(m,n,ix,iy)=aa(m,n,ix,iy)+ca(m,n,ix,iy)
            enddo
         enddo
c
      rewind 4
c
      enddo
c
         do ix=1,nx
            do iy=1,ny
               zb(n,ix,iy)=zbn(ix,iy)
            enddo
          enddo
c
      enddo
c
c   computation of second mode coupling coefficients
c
      do n=1,nm
      do m=1,nm
c
         call sub3(rho1,rho2,zb,h,b1,ar,aa,n,m,hpr,hpa,
     *                                     zbnpr,zbnpa,x,y)
c
         do ix=1,nx
            do iy=1,ny
               if (icounter(ix,iy).lt.n.or.
     *                            icounter(ix,iy).lt.m) then
                  b(m,n,ix,iy)=0.d0
               else
                  b(m,n,ix,iy)=b1(ix,iy)
               endif
            enddo
         enddo
c
      enddo
      enddo
c
      write(6,*)'checking the mode coupling coefficients :'
      write(6,*)'ar(16,17,ix,iy)='
      write(6,100)((ar(16,17,ix,iy),ix=1,5),iy=1,5)
      write(6,*)'aa(16,17,ix,iy)='
      write(6,100)((aa(16,17,ix,iy),ix=1,5),iy=1,5)
      write(6,*)'b(16,17,ix,iy)='
      write(6,100)((b(16,17,ix,iy),ix=1,5),iy=1,5)
 100  format(5(1x,e12.5))
c
```

```
      write(8) ar
      write(8) aa
      write(8) b
c
      close(4)
      close(8)
      close(14)
      end
```

```
*************************************************************
*                                                           *
* This subroutine computes the vector mode coupling         *
* coefficient correction, due to small bathymetry           *
* variations between two different modes (m different       *
* than n.                                                   *
*                                                           *
*************************************************************
c
      subroutine sub1(rho1,rho2,dz,zbm,zbn,zbmm1,zbnm1,
     *                cr1,ca1,km,kn,m,n,zbnpr,zbnpa,x,y)
c
      parameter (nx=11,ny=11,nw=2*nx*ny+2*max(nx,ny))
      implicit real*8 (a-h,o-z)
      real*8 c(2,nx,2,ny),wk(nw),cr1(nx,ny),ca1(nx,ny),
     *                                            zbm(nx,ny)
      real*8 zbmm1(nx,ny),zbn(nx,ny),zbnm1(nx,ny),kn(nx,ny)
      real*8 zbnpz(nx,ny),zbmpz(nx,ny),zbnpzpr(nx,ny),
     *                                             km(nx,ny)
      real*8 zbnpzpa(nx,ny),zbnpzpx(nx,ny),zbnpzpy(nx,ny),
     *                                          zbnpr(nx,ny)
      real*8 zbnpa(nx,ny),rho1(nx,ny),rho2(nx,ny)
      real*8 x(nx),y(ny)
c
      do ix=1,nx
         do iy=1,ny
            zbnpz(ix,iy)=(zbn(ix,iy)-zbnm1(ix,iy))/dz
            zbmpz(ix,iy)=(zbm(ix,iy)-zbmm1(ix,iy))/dz
         enddo
      enddo
      call partial(zbnpz,nx,ny,x,y,zbnpzpr,zbnpzpa,nw,
     *                              zbnpzpx,zbnpzpy,c,wk)
      if (m.eq.2.and.n.eq.18) then
         write(6,*)'zbnpz(3,3)=',zbnpz(3,3)
         write(6,*)'zbnpzpr(3,3)=',zbnpzpr(3,3)
         write(6,*)'zbnpzpa(3,3)=',zbnpzpa(3,3)
         write(6,*)'zbnpr(3,3)=',zbnpr(3,3)
         write(6,*)'zbnpa(3,3)=',zbnpa(3,3)
      endif
c
      do ix=1,nx
         do iy=1,ny
            r=dsqrt(x(ix)**2 + y(iy)**2)
            if (r.lt.1.d-20) goto 100
            cr1(ix,iy)=zbnpzpr(ix,iy)*zbm(ix,iy)*
     *          (1.d0-rho2(ix,iy)/rho1(ix,iy))/rho1(ix,iy) -
     *          zbmpz(ix,iy)*zbnpr(ix,iy)*
     *          (1.d0/rho1(ix,iy)-1.d0/rho2(ix,iy))
            ca1(ix,iy)=zbm(ix,iy)*zbnpzpa(ix,iy)*
     *          (1.d0-rho2(ix,iy)/rho1(ix,iy))/
```

```
     *            (rho1(ix,iy)*r)  -  zbmpz(ix,iy)*
     *            zbnpa(ix,iy)*
     *            (1.d0/rho1(ix,iy)-1.d0/rho2(ix,iy))/r
              cr1(ix,iy)=cr1(ix,iy)*2.d0/(kn(ix,iy)-km(ix,iy))
              ca1(ix,iy)=ca1(ix,iy)*2.d0/(kn(ix,iy)-km(ix,iy))
 100          continue
           enddo
        enddo
c
        return
        end
```

```
****************************************************************
*                                                              *
* This subroutine computes the vector mode coupling            *
* coefficient correction  due to small bathymetry              *
* variations, in the case of m equals n.                      *
*                                                              *
****************************************************************
c
      subroutine sub2(rho1,rho2,zbm,h,cr1,ca1,nx,ny,
     *                                        hpr,hpa,x,y)
c
      implicit real*8 (a-h,o,z)
      real*8 cr1(nx,ny),ca1(nx,ny),zbm(nx,ny),rho1(nx,ny),
     *                                         rho2(nx,ny)
      real*8 hpr(nx,ny),hpa(nx,ny),x(nx),y(ny)
c
      do ix=1,nx
         do iy=1,ny
            r=dsqrt(x(ix)**2 + y(iy)**2)
            if (r.lt.1.d-20) goto 100
            cr1(ix,iy)=-(1.d0/rho1(ix,iy)
     *        -1.d0/rho2(ix,iy))*(zbm(ix,iy)**2)*hpr(ix,iy)
            ca1(ix,iy)=-(1.d0/rho1(ix,iy)-1.d0/rho2(ix,iy))*
     *         (zbm(ix,iy)**2)*hpa(ix,iy)/r
 100        continue
         enddo
      enddo
c
      return
      end
```

```
****************************************************************
*                                                              *
*  This subroutine computes the scalar mode coupling           *
*  coefficient small bathymetry changes included.              *
*                                                              *
****************************************************************
c
      subroutine sub3(rho1,rho2,zb,h,b1,ar,aa,n,m,hpr,hpa,
     *                                      zbnpr,zbnpa,x,y)
c
      parameter (nx=11,ny=11,nw=2*nx*ny+2*max(nx,ny),nm=20)
      implicit real*8 (a-h,o-z)
      real*8 c(2,nx,2,ny),wk(nw),zb(nm,nx,ny),h(nx,ny),
     *                                                 b1(nx,ny)
      real*8 hpr(nx,ny),hpa(nx,ny),ar(nm,nm,nx,ny),
     *                                         aa(nm,nm,nx,ny)
      real*8 zbnpr(nx,ny),zbnpa(nx,ny),ern(nm,nx,ny)
      real*8 ean(nm,nx,ny),sum(nx,ny),armn(nx,ny),
     *                                               aamn(nx,ny)
      real*8 arpr(nx,ny),arpa(nx,ny),arpx(nx,ny),arpy(nx,ny)
      real*8 aapr(nx,ny),aapa(nx,ny),aapx(nx,ny),aapy(nx,ny)
      real*8 x(nx),y(ny),erm(nm,nx,ny),eam(nm,nx,ny)
      real*8 rho1(nx,ny),rho2(nx,ny)
c
      np=17
      mp=16
c
      if (n.eq.np.and.m.eq.mp) then
         write(6,*)'rho1'
         write(6,100)((rho1(ix,iy),ix=1,5),iy=5,1,-1)
         write(6,*)'rho2'
         write(6,100)((rho2(ix,iy),ix=1,5),iy=5,1,-1)
         write(6,*)'hpr'
         write(6,100)((hpr(ix,iy),ix=1,5),iy=5,1,-1)
         write(6,*)'zb(',m,',...'
         write(6,100)((zb(m,ix,iy),ix=1,5),iy=5,1,-1)
         write(6,*)'zb(',n,',...'
         write(6,100)((zb(n,ix,iy),ix=1,5),iy=5,1,-1)
      endif
c
      do ix=1,nx
         do iy=1,ny
            sum(ix,iy)=0.d0
         enddo
      enddo
c
      do l=1,nm
         do ix=1,nx
            do iy=1,ny
            r=dsqrt(x(ix)**2 + y(iy)**2)
            if (r.lt.1.d-20) goto 110
```

```
          ern(l,ix,iy)=.5d0*ar(n,l,ix,iy)  + hpr(ix,iy)*
     *              (1.d0/rho1(ix,iy)-1.d0/rho2(ix,iy))*
     *              zb(n,ix,iy)*zb(l,ix,iy)
          ean(l,ix,iy)=.5d0*aa(n,l,ix,iy)  + hpa(ix,iy)*
     *              (1.d0/rho1(ix,iy)-1.d0/rho2(ix,iy))*
     *              zb(n,ix,iy)*zb(l,ix,iy)/r
          erm(l,ix,iy)=.5d0*ar(m,l,ix,iy)  + hpr(ix,iy)*
     *              (1.d0/rho1(ix,iy)-1.d0/rho2(ix,iy))*
     *              zb(m,ix,iy)*zb(l,ix,iy)
          eam(l,ix,iy)=.5d0*aa(m,l,ix,iy)  + hpa(ix,iy)*
     *              (1.d0/rho1(ix,iy)-1.d0/rho2(ix,iy))*
     *              zb(m,ix,iy)*zb(l,ix,iy)/r
 110      continue
        enddo
      enddo
    enddo
c
    do ix=1,nx
       do iy=1,ny
          do l=1,nm
             sum(ix,iy)=sum(ix,iy) +
     *                    ern(l,ix,iy)*erm(l,ix,iy)+
     *                    ean(l,ix,iy)*eam(l,ix,iy)
          enddo
       enddo
    enddo
c
    if (n.eq.np.and.m.eq.mp) then
       write(6,*)'sum for im,in=',mp,np
       write(6,100)((sum(ix,iy),ix=1,5),iy=5,1,-1)
  100 format(5(1x,e12.3))
    endif
c
    do ix=1,nx
       do iy=1,ny
          armn(ix,iy)=ar(m,n,ix,iy)
          aamn(ix,iy)=aa(m,n,ix,iy)
       enddo
    enddo
c
    call partial(armn,nx,ny,x,y,arpr,arpa,nw,
     *                                          arpx,arpy,c,wk)
    call partial(aamn,nx,ny,x,y,aapr,aapa,nw,
     *                                          aapx,aapy,c,wk)
c
    do ix=1,nx
       do iy=1,ny
          r=dsqrt(x(ix)**2 + y(iy)**2)
          if (r.lt.1.d-20) then
          b1(ix,iy)=.5d0*arpr(ix,iy) - sum(ix,iy) -
     *    (1.d0/rho1(ix,iy)-1.d0/rho2(ix,iy))*zb(m,ix,iy)*
```

```
     *        (hpr(ix,iy)*zbnpr(ix,iy) )
              else
              b1(ix,iy)=.5d0*arpr(ix,iy)+.5d0*ar(m,n,ix,iy)/r+
     *                  .5*d0*aapa(ix,iy)/r - sum(ix,iy) -
     *                  (1.d0/rho1(ix,iy)-
     *                  1.d0/rho2(ix,iy))*zb(m,ix,iy)*
     *                  (hpr(ix,iy)*zbnpr(ix,iy)+
     *                  hpa(ix,iy)*zbnpa(ix,iy)/r**2)
              endif
           enddo
        enddo
c
        return
        end
```

```
*************************************************************
*                                                           *
* This subroutine computes the partial derivatives with     *
* respect to range and azimuthal angle of a given function *
* f(x,y). It uses a bicubic spline to calculate the         *
* cartesian derivatives and then perform a coordinate       *
* transformation to cylindrical coordinates.                *
*                                                           *
*************************************************************
c
      subroutine partial(f,nx,ny,x,y,fpr,fpa,nw,fpx,fpy,
     *                                                  c,wk)
c
      implicit real*8 (a-h,o-z)
      real*8 f(nx,ny),x(nx),y(ny),fpr(nx,ny),fpa(nx,ny)
      real*8 wk(nw),fpx(nx,ny),fpy(nx,ny),c(2,nx,2,ny)
c
      external ibcccu
c
      ic=nx
      pi=dacos(-1.d0)
c
      call ibcccu(f,x,nx,y,ny,c,ic,wk,ier)
c
     do ix=1,nx
         do iy=1,ny
            fpx(ix,iy)=c(2,ix,1,iy)
            fpy(ix,iy)=c(1,ix,2,iy)
            if (x(ix).eq.0.d0) then
               theta=dsign(y(iy),1.d0)*pi/2.d0
            else
               theta=datan(y(iy)/x(ix))
            endif
            r=dsqrt(x(ix)**2+y(iy)**2)
            fpr(ix,iy)=fpx(ix,iy)*dcos(theta)+
     *                 fpy(ix,iy)*dsin(theta)
            fpa(ix,iy)=-fpx(ix,iy)*r*dsin(theta)+
     *                 fpy(ix,iy)*r*dcos(theta)
         enddo
      enddo
c
      return
      end
```

LIST OF REFERENCES

1. R.M. Jones, J.P. Riley, T.M. Georges, "HARPO A Versatile Three-Dimensional Hamiltonean Ray-Tracing Program for Acoustic Waves in an Ocean with Irregular Bottom," *Wave Prop. Lab. NOAA, Boulder, Colorado, 1986, pp.457.*

2. F.D. Tappert, "The Parabolic Approximation Method," in *Wave Propagation and Underwater Acoustics,* edited by J.B.Keller and J.S.Papadakis, Springer-Verlang, New York, 1977.

3. D. Lee, Y. Saad, and M.H. Schultz, "An Efficient Method for Solving the Three-Dimensional Wide-angle Wave Equation," *Computational Acoustics, IMACS, 1988, pp.75-89.*

4. R.N. Baer, "Propagation Through a Three-Dimensional Eddy Including Effects on an Array," *J. Acoust. Soc. Am., v.69(1), 1981, pp.70-75.*

5. A.D. Pierce, "Extension of the Method of Normal Modes to Sound Propagation in an Almost Stratified Medium," *J.Acoust. Soc. Am., v.37, 1965, pp.19-27.*

6. C.S. Chiu, and L.Ehret, *Computation of Sound Propagation in a Three-Dimensionally Varying Ocean: A Coupled Normal Mode Approach*, Computational Acoustics, Volume 1, Elsevier Science Publishers B.V.,1990.

BIBLIOGRAPHY

1. F.S. Acton, *Numerical Methods That Work*, Harper & Row Publishers, New York, 1970.

2. C.S. Clay, and H. Medwin, *Acoustical Oceanography: Principles and Applications*, John Wiley & Sons,Inc., New York, 1977.

3. D.M. Etter, *Structured FORTRAN 77 for Engineers and Scientists*, The Benjamin/Cummings Publishing Company Inc., New York, 1990.

4. C.F. Gerald, and P.O. Wheatley, *Applied Numerical Analysis*, Addison-Wesley, New York, July 1990.

5. L.E. Kinsler, A.R. Frey, A.B. Coppens, J.V. Sanders, *Fundamentals of Acoustics*, John Wiley & Sons, New York, 1982.

6. D.D. McCracken, and W.I. Salmon, *Computing for Engineers and Scientists with FORTRAN 77*, John Wiley & Sons, New York, 1988.

7. S.R. Rutherford, *An Examination of Coupled Mode Theory as Applied to Underwater Sound Propagation*, Ph.D. Dissertation, August 1979.

8. G.D. Smith, *Numerical Solution of Partial Differential Equations: Finite Difference Methods*, Clarendon Press, Oxford, 1985.

9. I. Tolstoy, and C.S. Clay, *Ocean Acoustics: Theory and Experiment in Underwater Sound*, McGraw-Hill, New York, 1966.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Chairman, Code OC Department of Oceanography Naval Postgraduate School Monterey, CA 93943	1
4.	Chairman, Code EC Department of Electrical & Computer Engineering Naval Postgraduate School Monterey, CA 93943	1
5.	Chairman, Code PH Department of Physics Naval Postgraduate School Monterey, CA 93943	1
6.	Professor Ching-Sang Chiu, Code OC/Ci Department of Oceanography Naval Postgraduate School Monterey, CA 93943	2
7.	Professor James H. Miller, Code EC/Mr Department of Electrical & Computer Engineering Naval Postgraduate School Monterey, CA 93943	1
8.	Professor Laura L.Ehret, Code OC/Eh Department of Oceanography Naval Postgraduate School Monterey, CA 93943	1
9.	Professor Alan B. Coppens, Code PH/Co Department of Physics Naval Postgraduate School Monterey, CA 93943	1

10. Embassy of Greece 2
Naval Attache
2228 Massachusetts Ave, NW
Washington, DC 20008

11. LTJG George Sagos 1
Kavalas 2, New Philadelphia
Athens, GREECE