CINTAL - Centro de Investigação Tecnológica do Algarve
Universidade do Algarve

Hydrodynamical Normal Mode Estimation on the INTIFANTE'00 data set

TOMPACO Phase II - Part I

O.C. Rodríguez

Rep 04/01 - SiPLAB
23/November/2001

University of Algarve
Campus da Penha
8000, Faro,
Portugal

tel: +351-289800131
fax: +351-289864258
cintal@ualg.pt
www.ualg.pt/cintal

Work requested by	CINTAL Universidade do Algarve, Campus da Penha, 8000 Faro, Portugal tel: +351-289800131, cintal@ualg.pt, www.ualg.pt/cintal
Laboratory performing the work	SiPLAB - Signal Processing Laboratory Universidade do Algarve, FCT, Campus de Gambelas, 8000 Faro, Portugal tel: +351-289800949, info@siplab.uceh.ualg.pt, www.ualg.pt/uceh/adeec/siplab
Projects Title	TOMPACO Hydrodynamical Normal Modes estimation on the INTIFANTE'00 data set
Authors	O.C. Rodríguez
Date	November 23, 2001
Reference	04/01 - SiPLAB
Number of pages	28 (twenty eight)
Abstract	This report introduces a theoretical description of Dynamic Normal Modes, their relationship to temperature, salinity, and sound speed perturbations, and also of the numerical methods that allow the calculation of Dynamic Normal Modes.
Clearance level	UNCLASSIFIED
Distribution list	DUNE (1), SiPLAB (1), CINTAL (1)
Total number of copies	3 (three)

intentionally blank

Contents

List of Figures

intentionally blank

Chapter 1

Introduction

Dynamic modes are important to describe the motion of a fluid and can be easily derived in the hydrostatic and non-hydrostatic linear cases from the general equations of fluid dynamics, neglecting the effects of friction and viscosity. In general the dynamic modes can be used to describe in detail the fields of pressure, density and currents. However, as will be shown in the following chapter, dynamic modes can be used to introduce a semi-orthogonal representation of the temperature, salinity and sound speed fields, which are the fundamental objects of interest in acoustic tomography. This internal report describes the theoretical background behind the description of dynamic modes, their relationship to temperature and sound speed, and also describes the numerical methods that allow to calculate a particular set of dynamic modes from a single temperature profile.

Chapter 2

Dynamic modes

2.1 Theoretical background

In general, the motion of a fluid obeys to the set of Navier-Stokes equations [1, 2], which include the terms of viscosity and friction. These terms play a minor role in the motion of the oceanic masses of water, allowing one to introduce the following equation for the motion of a fluid [3]:

$$\rho \frac{\partial \mathbf{U}}{\partial t} + \rho \left(\mathbf{U} \cdot \nabla \right) \mathbf{U} + 2\rho \mathbf{\Omega} \times \mathbf{U} = -\nabla p + \rho \mathbf{g} \ , \tag{2.1}$$

where $\mathbf{U} = (u, v, w)$ represents the velocity of the fluid particles, p represents the fluid pressure, ρ corresponds to its density, t represents the time coordinate, $\mathbf{\Omega}$ corresponds to the vector of angular rotation of the Earth and $\mathbf{g} = -\mathbf{k}g$, with g representing the acceleration of free fall in the gravity field of the Earth, ($g = 9.8$ m/s^2). In Eq.(2.1) ∇ represents the "nabla" operator: $\nabla = \mathbf{i}\partial/\partial x + \mathbf{j}\partial/\partial y + \mathbf{k}\partial/\partial z$. In general the terms to the left of the equality sign in Eq.(2.1) describe the motion of the fluid particles within a non-inertial frame of reference and in the absence of viscosity. The terms to the right of the sign describe the combined action of the external fields of the pressure and gravity forces. Eq.(2.1) can be simplified further by introducing the *Boussinesq approximation* [2], which states that the perturbations in density, $\rho' = \rho - \rho_0$, play a second order role in the calculations of the terms to the left of Eq.(2.1). In this way one can substitute in those terms, without loss of generality, the total density, ρ, by the equilibrium density, ρ_0. However, the same reasoning is not valid for the terms to the right of the equation. Therefore, according to the Boussinesq approximation, one can rewrite Eq.(2.1) as

$$\rho_0 \frac{\partial \mathbf{U}}{\partial t} + 2\rho_0 \mathbf{\Omega} \times \mathbf{U} + \rho_0 \left(\mathbf{U} \cdot \nabla \right) \mathbf{U} = -\nabla p + \rho \mathbf{g} \ . \tag{2.2}$$

Eq.(2.2) is insufficient to develop a complete analysis of the motion of the oceanic masses. In addition to that equation one can show that the velocity and density of the fluid particles are related to each other through the *Continuity equation* [4]:

$$\frac{1}{\rho} \frac{D\rho}{Dt} + \nabla \cdot \mathbf{U} = 0 \ , \tag{2.3}$$

where the operator of the total derivative is defined as

$$\frac{D}{Dt} \equiv \frac{\partial}{\partial t} + \mathbf{U} \cdot \nabla \ . \tag{2.4}$$

Eq.(2.3) can be splitted in two independent equations applying the *incompressibility condition*:

$$\frac{D\rho}{Dt} = 0 \ , \tag{2.5}$$

where the first of the equations corresponds to

$$\nabla \cdot \mathbf{U} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \ , \tag{2.6}$$

while the second, in its full form, is given by:

$$\frac{D\rho}{Dt} = \frac{\partial \rho}{\partial t} + u\frac{\partial \rho}{\partial x} + v\frac{\partial \rho}{\partial y} + w\frac{\partial \rho}{\partial z} = 0 \ . \tag{2.7}$$

Neglecting the non-linear terms $u\partial\rho/\partial x$ and $v\partial\rho/\partial y$ in Eq.(2.7), taking into account that $\rho = \rho_0(z) + \rho'$, and considering that $w\partial\rho/\partial z \approx w d\rho_0/dz$, one can obtain the following expression:

$$\frac{\partial \rho'}{\partial t} + w\frac{d\rho_0}{dz} \ = \ \frac{\partial \rho'}{\partial t} - N^2\frac{\rho_0}{g}w = \ 0 \ . \tag{2.8}$$

where N^2 is known as the *buoyancy frequency* (or the *Brunt-Väiasällä frequency*) [3]:

$$N^2 = -\frac{g}{\rho_0}\frac{d\rho_0}{dz} \ . \tag{2.9}$$

The buoyancy frequency corresponds to the frequency of natural oscillations of a fluid element, when that element is in the state of small amplitude harmonic motion along the vertical axis. Taking into account that $N(z)$ depends on the gradient of the equilibrium density, ρ_0, the dependency of the buoyancy frequency on depth constitutes a fundamental indicator of the environment stratification and of its stable equilibrium. Furthermore, the buoyancy profile imposes an upper limit (known as the *cuttoff frequency*) to the interval of natural frequencies of the water column.

The system of equations Eq.(2.2), Eq.(2.6) and Eq.(2.8), constitutes the starting point for the discussion, in sections 2.2 and 2.3 of the two more relevant cases of the propagation of internal waves.

2.2 Hydrostatic Normal Modes

The simplest case of propagation of internal waves corresponds to the hydrostatic linear rotationless case. First, let one admit the validity of the *hydrostatic approximation* [3] for the density and pressure of the water column:

$$\frac{\partial p}{\partial z} + \rho g = 0 \ . \tag{2.10}$$

This approximation implies automatically that $\partial w/\partial t = 0$, which corresponds to making the vertical component of Eq.(2.2) equal to zero. Furthermore, let one neglect in Eq.(2.2) the non-linear and rotational terms:

$$(\mathbf{U} \cdot \nabla)\,\mathbf{U} \approx \mathbf{0} \ \text{ and } \ \mathbf{\Omega} \times \mathbf{U} \approx \mathbf{0} \ . \tag{2.11}$$

In this way, based on the approximations (2.10) and (2.11), and after rearranging some of the terms, one can obtain the following components of Eq.(2.2):

$$\frac{\partial u}{\partial t} = -\frac{1}{\rho_0}\frac{\partial p}{\partial x} \ , \quad \text{and} \quad \frac{\partial v}{\partial t} = -\frac{1}{\rho_0}\frac{\partial p}{\partial y} \ . \tag{2.12}$$

It can be shown that the fields of currents, density perturbations and pressure, that satisfy the system of equations Eq.(2.6), Eq.(2.8), and the pair of equations (2.12), can be represented in terms of expansions on a basis of Hydrostatic Normal Modes (HNMs) Ψ_m and ϕ_m [5]:

$$\begin{aligned} w &= \sum_m w_m \Psi_m(z) \ , \quad (u,v) = D\sum_m (u_m, v_m)\,\phi_m(z) \ , \\ \rho' = \rho_0 \frac{N^2}{g} &\sum_m \rho_m \Psi_m(z) \ , \quad p = \rho_0 \sum_m p_m \phi_m(z) \ , \end{aligned} \tag{2.13}$$

where D represents the water column depth, and the modal amplitudes u_m, v_m, w_m, ρ_m and p_m depend on the horizontal coordinates (x, y), and on time t. The HNMs are related through the equation $\phi_m = d\Psi_m/dz$, where the functions Ψ_m correspond to the solutions of a Sturm-Liouville Problem (hereafter, SLP) [6]:

$$\frac{d^2\Psi_m}{dz^2} + \frac{N^2}{C_m^2}\Psi_m = 0 \tag{2.14}$$

$$+ \text{ Boundary Conditions (BCs).}$$

In Eq.(2.14) the coefficients C_m represent the propagation velocity of linear hydrostatic internal waves in a rotationless environment. ¿From the mathematical point of view the SLP guarantees the existence of a complete system of eigenfunctions Ψ_m, with orthogonal properties:

$$\langle \Psi_m \left| N^2 \right| \Psi_n \rangle = 0 \quad \text{when } m \neq n \ ; \tag{2.15}$$

in Eq.(2.15) the "inner product" $\langle f_1 \left| f_2 \right| f_3 \rangle$ is defined as

$$\langle f_1 \left| f_2 \right| f_3 \rangle = \int_0^D f_1 f_2 f_3 \ dz \ . \tag{2.16}$$

Moreover, the coefficients C_m^{-2} correspond to the eigenvalues of the functions Ψ_m. For an arbitrary choice of BCs the orthogonality of the eigenfunctions Ψ_m does not imply the orthogonality of their derivatives, ϕ_m. However, for the particular case of homogeneous BCs, on bottom and surface:

$$\Psi_m(0) = \Psi_m(D) = 0 \ , \tag{2.17}$$

one obtains that

$$\langle \phi_m \mid \phi_n \rangle = 0 \ , \tag{2.18}$$

where $\langle f_1 \mid f_2 \rangle = \langle f_1 \left| 1 \right| f_2 \rangle$. Furthermore, on the basis of the inner products (2.15) and (2.18), one can show the validity of the following relationships:

$$\begin{aligned} \langle \Psi_m \left| N^2 \right| \Psi_m \rangle &= C_m^2 \langle \phi_m \mid \phi_m \rangle \ , \\ \langle \Psi_m \left| N^2 \right| \phi_m \rangle &= \frac{1}{2} C_m^2 \langle \phi_m \left| \phi_m \right| \phi_m \rangle \ , \\ \langle \Psi_m \left| \frac{d\phi_m}{dz} \right| \phi_m \rangle &= -\frac{1}{2} \langle \phi_m \left| \phi_m \right| \phi_m \rangle \ , \\ \langle \Psi_m \left| N^2 \right| \phi_m^2 \rangle &= -\frac{1}{3} C_m^2 \ \phi_m^3 \Big|_0^D \ . \end{aligned} \tag{2.19}$$

From the physical point of view one can expect that the modal amplitudes p_m, ρ_m, ... w_m, will exhibit an oscillating behaviour. It should be remarked that the set of expansions (2.13) do not constrain in any particular manner the analytic choice of those amplitudes. However, the consistency of the system of equations (2.6), (2.8), (2.10) and (2.12)[1] implies the following linear interdependency of those amplitudes:

$$\begin{aligned} & D\left(\frac{\partial u_m}{\partial x}+\frac{\partial v_m}{\partial y}\right)+w_m=0\ , \quad \frac{\partial \rho_m}{\partial t}-w_m=0\ , \\ & D\frac{\partial u_m}{\partial t}=-\frac{\partial p_m}{\partial x}\ , \qquad D\frac{\partial v_m}{\partial t}=-\frac{\partial p_m}{\partial y}\ , \\ & \frac{p_m}{C_m^2}-\rho_m=0\ . \end{aligned} \tag{2.20}$$

In this way, by imposing a particular set of periodic conditions on a particular amplitude, one will define automatically the particular analytic structure of the other modal amplitudes.

It should be remarked that the hydrostatic linear rotationless case can be analytically extended in order to consider the presence of a mean gradient of the velocity components u and v (see, for instance, [7, 8]). The description of that case, which is of significant importance from a theoretical point of view, would exceed the objectives of this discussion and will not be considered.

2.3 Non-hydrostatic Normal Modes

Neglecting non-linear terms in Eq.(2.1)

$$(\mathbf{U}\cdot\nabla)\,\mathbf{U}\approx\mathbf{0}\ , \tag{2.21}$$

considering that $p' = p - p_0$, and constraining the hydrostatic approximation to the equilibrium terms (i.e., considering that $dp_0/dz + \rho_0 g = 0$), one can obtain the following system of equations:

$$\begin{aligned} \left(\frac{\partial u}{\partial t}-f_c v\right) &= -\frac{1}{\rho_0}\frac{\partial p'}{\partial x}\ , \\ \left(\frac{\partial v}{\partial t}+f_c u\right) &= -\frac{1}{\rho_0}\frac{\partial p'}{\partial y}\ , \\ \left(\frac{\partial w}{\partial t}\right) &= -\frac{1}{\rho_0}\frac{\partial p'}{\partial z}-\frac{\rho'}{\rho_0}g\ . \end{aligned} \tag{2.22}$$

In system (2.22) the parameter $f_c = 2\Omega\sin\vartheta$ is known as the *Coriolis frequency* [1] and ϑ corresponds to the geographic latitude; the Coriolis frequency plays an important role in the study of the motion of a fluid within a rotating system of reference.

The solutions of the system of equations (2.6), (2.8) and (2.22) for the fields of current components, and perturbations of pressure and density, can be represented again under the form of orthogonal expansions [1]:

$$\begin{aligned} w &= \sum_m w_m\tilde{\Psi}_m(z) \quad , \quad (u,v) = \sum_m (u_m,v_m)\,\tilde{\phi}_m(z) \quad , \\ \rho' &= \rho_0 N^2\sum_m \rho_m\tilde{\phi}_m(z) \quad , \quad p' = \rho_0\sum_m p_m\tilde{\phi}_m(z) \quad , \end{aligned} \tag{2.23}$$

[1] Including the approximation $\partial p/\partial z \approx \rho_0 \sum_m p_m d\phi_m/dz$.

where $\tilde{\phi}_m = d\tilde{\Psi}_m/dz$; the *non-hydrostatic* normal modes $\tilde{\Psi}_m$ are, again, eigenfunctions of a SLP of the following form:

$$\frac{d^2\tilde{\Psi}_m}{dz^2} + \left(k_h^2\right)_m \frac{N^2 - \tilde{\omega}^2}{\tilde{\omega}^2 - f_c^2}\tilde{\Psi}_m = 0 \quad + \quad \text{BCs} \quad , \tag{2.24}$$

which guarantees the orthogonal properties of the modes $\tilde{\Psi}_m$:

$$\langle\tilde{\Psi}_m \left| \frac{N^2 - \tilde{\omega}^2}{\tilde{\omega}^2 - f_c^2} \right| \tilde{\Psi}_n\rangle = 0 \quad \text{with } m \neq n \ . \tag{2.25}$$

In Eq.(2.24) $\tilde{\omega}$ corresponds to the frequency of the internal waves, and k_h represents the horizontal component of the wavenumber vector. Denoting as θ the direction of propagation of internal waves one obtains that

$$\mathbf{k}_h = \mathbf{i}k_x + \mathbf{j}k_y \quad \text{and} \quad k_x = k_h \cos\theta \ , \ k_y = k_h \sin\theta \ . \tag{2.26}$$

In contrast to the hydrostatic linear case the consistency of the system of equations (2.6), (2.8) and (2.22) depends on the constraint

$$(\ p_m \ , \ \rho_m \ , \ u_m \ , \ v_m \ , \ w_m \ ) \sim \exp\left[i\left(k_x x + k_y y - \tilde{\omega}t\right)\right] \ , \tag{2.27}$$

which imposes the particular application of expansions (2.23) to the case of plane-wave propagation.

2.4 Buoyancy and temperature

As commented previously the vertical stratification of the environment is represented in the differential equation for Ψ_m through the buoyancy frequency $N^2(z)$, which is normally related to mean density. For inversion it is better to use the alternative relationship [1]

$$N^2 = g\left[a_T \frac{dT_0}{dz} + a_T^2 \frac{gT_0}{C_{ps}} - a_s \frac{ds}{dz}\right] \ , \tag{2.28}$$

where $a_T = 2.41{\times}10^{-4}\,(^\circ\mathrm{C})^{-1}$ and $C_{ps} = 3994\ \mathrm{J(kg^\circ C)^{-1}}$. Usually the salinity depends weakly on depth so we can neglect the salinity term and develop a buoyancy profile that depends only on temperature.

2.5 Temperature perturbations

¿From the tomography point of view, the system formed by equations (2.2), (2.6) and (2.8), does not provide a clear physical basis for the analysis of temperature perturbations of the water column. In order to include the temperature within the context of the propagation problem of internal waves it becomes necessary to add a system of thermodynamic equations, relating the field of currents, $\mathbf{U}$, to the temperature field, T. By analogy with the general scheme illustrated in [9], one can consider the following thermodynamic equation [2]:

$$\frac{D}{Dt}\left(\rho c_v T\right) = \nabla \cdot \left(k_T \nabla T\right) + Q_T \ , \tag{2.29}$$

where c_v represents the specific heat of the water column, k_T corresponds to the thermal fluid conductivity, and Q_T represents the external sources of heat. In the general case it is not clear which terms can be neglected and which ones can not. However, considering both density and specific heat as constants and taking $(k_T, Q_T) = 0$ one can easily rewrite Eq.(2.29) as

$$\frac{DT}{Dt} = \frac{\partial T}{\partial t} + u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} + w\frac{\partial T}{\partial z} = 0 \ ; \tag{2.30}$$

by neglecting coupling mechanisms and taking into account that an important feature of internal waves corresponds to the significant dynamics along the depth axis, one can neglect the second and third nonlinear terms, and rewrite Eq.(2.30) as:

$$\frac{\partial T}{\partial t} + w\frac{\partial T}{\partial z} = 0 \ . \tag{2.31}$$

Now, let us consider that

$$T(x, y, z, t) \approx T_0(z) + \delta T \ , \tag{2.32}$$

where

$$\delta T = \sum_m \alpha_m(x, y, t) T_m(z) \Psi_m(z) \ ; \tag{2.33}$$

where T_m is an unknown function, and α_m is a dimensional coefficient of modal amplitude for temperature. α_m and T_m should be chosen in order to ensure the consistency of Eq.2.31. Neglecting once more coupling mechanisms between modes one can consider that

$$w\frac{\partial T}{\partial z} \approx \frac{dT_0}{dz}\sum_m w_m \Psi_m \ ; \tag{2.34}$$

further, for the first term in Eq.(2.31) one gets that

$$\frac{\partial T}{\partial t} = \sum_m \frac{\partial \alpha_m}{\partial t}\gamma_m(z)\Psi_m \ . \tag{2.35}$$

Substituting Eq.(2.34) and Eq.(2.35) into Eq.2.31 it follows automatically that

$$\frac{\partial \alpha_m}{\partial t} = -w_m \quad \text{and} \quad \gamma_m = \frac{dT_0}{dz} \ . \tag{2.36}$$

The minus sign indicates that the time oscillations of w and T have a phase difference of π radians. The last pair of equations lead to the following expansion of temperature perturbations:

$$T - T_0(z) = \frac{dT_0}{dz}\sum_m \alpha_m(x, y, t)\Psi_m(z) \ . \tag{2.37}$$

2.6 Salinity perturbations

In contrast with the temperature field, the salinity distribution, S, is not a common object of discussion within the context of the tomography problem. However, since the salinity field obeys to the differential equation [2]:

$$\frac{DS}{Dt} = \nabla \cdot (K_S \nabla S) + Q_S \ , \tag{2.38}$$

which has a structure similar to the one of Eq.(2.29), one can admit the following orthogonal expansion for salinity:

$$S - S_0(z) = \frac{dS_0}{dz}\sum_m \alpha_m(x, y, t)\Psi_m(z) \ . \tag{2.39}$$

2.7 Sound speed perturbations

The almost linear relationship between temperature and sound speed is well known. For instance, in Mackenzie's expansion [10]:

$$\begin{aligned} c &= 1448.96 + 4.591 \times T - 5.304 \times 10^{-2} T^2 + 2.374 \times 10^{-4} T^3 + \\ &\quad +1.304 \times (S - 35) + 1.630 \times 10^{-2} z + 1.675 \times 10^{-7} z^2 + \\ &\quad +1.025 \times T\,(35 - S) - 7.139 \times 10^{-13} T z^3 \ . \end{aligned} \tag{2.40}$$

it is evident the significant weigth of the linear coefficient on the linear term of temperatura. Therefore, one can admit the following orthogonal expansion for sound speed:

$$c - c_0(z) = \frac{dc_0}{dz} \sum_m \alpha_m(x, y, t) \Psi_m(z) \ . \tag{2.41}$$

In this way, calculating the modal amplitudes of sound speed allows to invert for temperature and salinity.

Chapter 3

Application to real data

The previous chapter was dedicated to the theoretical aspects involving the relationships between dynamic modes and temperature, salinity and sound speed. This chapter illustrates a few applications to real data, namely the calculation of the dynamic modes in the hydrostatic case (hereafter, hydrostatic normal modes or HNMs) from mean temperature. The hydrographic data described in this section was acquired during the INTIFANTE'00 tomography experiment [11].

3.1 Mean temperature

The mean temperature profile of the INTIFANTE'00 sea trial, T_0 (see Fig.3.1), exhibits a well defined two-layer stratification of the water column, with a thermocline which can be easily identified. This fact makes it evident the simple stratification of the monitorized water column. This particularity of the mean temperature profile constitutes an important indicator of the discrete density variations in depth, which enhance the significance of the dynamic modes.

3.2 Mean buoyancy

The calculation of hydrostatic and non-hydrostatic dynamic modes can be accomplished by calculating the mean buoyancy profile from temperature thermistor data, using the relationship Eq.(2.28), which reduces the information involved in acoustic tomography. By eliminating the vertical gradient of salinity, S, (i.e., admitting that the vertical variation of salinity can be neglected) one obtains an expression that depends only on the temperature, T_0, and that can be used to calculate the dynamic modes referred above. The result is shown in Fig.3.2.

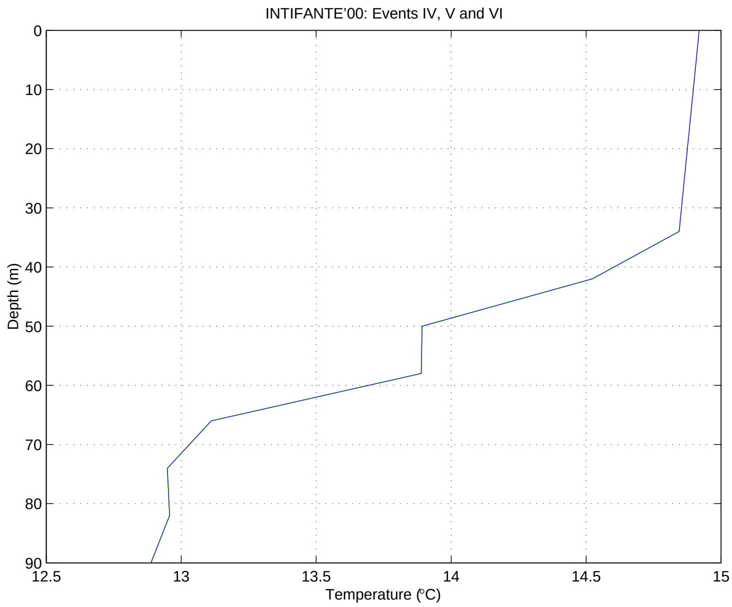

Figure 3.1: Mean temperature profile $T_0(z)$ from thermistor data acquired during Events IV, V and VI (INTIFANTE'00 experiment).

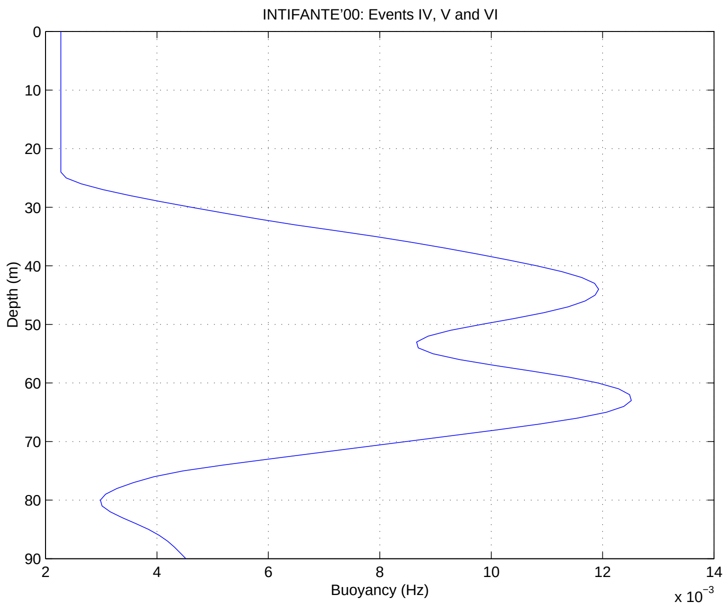

Figure 3.2: Mean buoyancy profiles $N(z)$ estimated using mean temperature (INTIFANTE'00 experiment).

3.3 Hydrostatic Normal Modes

The set of hydrostatic normal modes Ψ_m is shown in Fig.3.3. The calculation was based on the buoyancy profile calculated from mean temperature, and was developed through the usage of M-files in order to calculate the corresponding derivatives and to solve the associate Sturm-Liouville problem. The numerical solution of the Sturm-Liouville problem is explained in Appendix A and the M-files used in the calculations are presented in Appendix B.

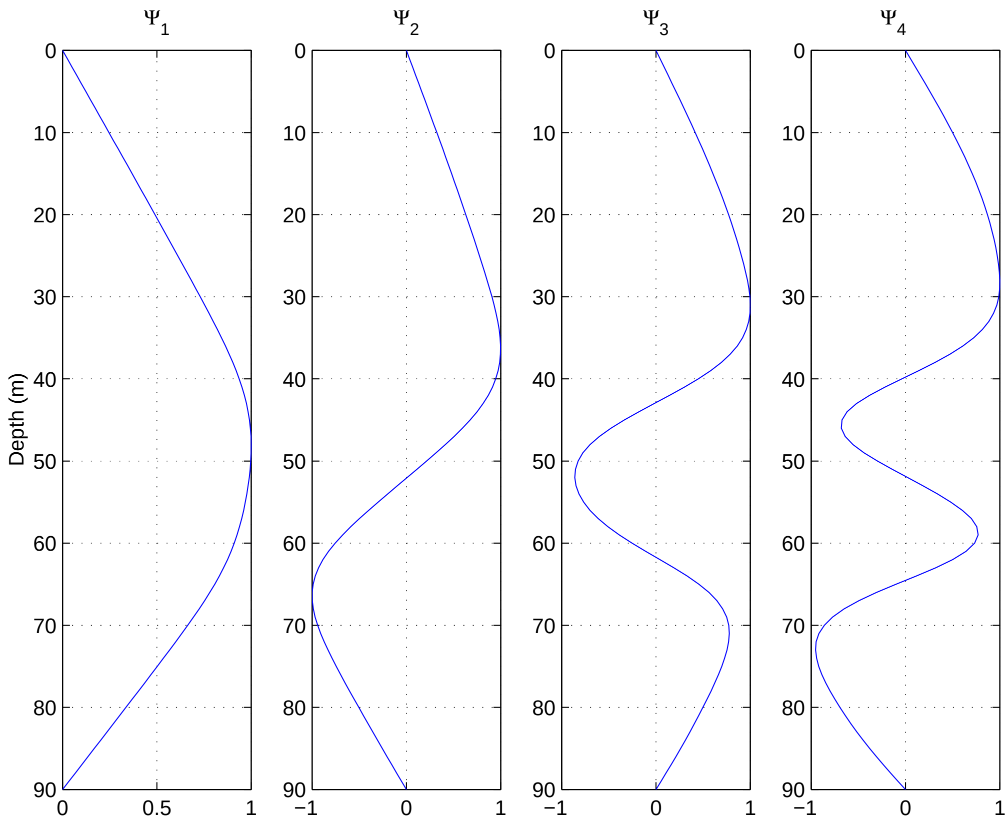

Figure 3.3: Hidrostatic normal modes calculated from mean temperature (INTIFANTE'00 experiment).

Bibliography

[1] Apel J.R., "Principles of Ocean Physics", Academic Press, International Geophysics Series, Vol.38, 1987, London.

[2] LeBlond P.H. and Mysak L.A., "Waves in the Ocean", Elsevier Scientific Publishing Company, New York, 1989.

[3] Gill Adrian E., "Atmosphere-Ocean Dynamics", Academic Press, International Geophysics Series, Vol. 30, 1982, London.

[4] Landau L.D. and Lifshitz E.M., "Plasticity Theory, VII Volume of Theoretical Physics", Nauka, 1987, Moscow (in russian).

[5] Ostrovsky L.A., "Nonlinear Internal Waves in a Rotating Ocean", Oceanology, Number 2, Vol.18, pp. 119-125, 1978.

[6] Butkov E., "Física Matemática", Editora Guanabara, 1988, Rio de Janeiro.

[7] Apel J., Orr. M., Finette S. and Lynch J., "A new model for internal solitons and tides on the continental shelf", Shallow-Water Acoustics: Proceedings of the SWAC'97 Conference, Zhang R. and Zhou J., pp. 219–225, China Ocean Pres, Beijing, April, 1997.

[8] Jackson J.F., "A brief review of Internal Waves in the ocean", Unit for Coastal and Estuarine Studies, UCES report, Number U96-7, University of Wales, Bangor, August, 1996.

[9] Chen P., Ingram R.G. and Gan J., "A Numerical Study of Hydraulic Jump and Mixing in a Stratified Channel With a Sill", Proceedings of the 3rd International Conference on Estuarine and Coastal Modelling, Oak Brook, pp. 119-133, Illinois, 1993.

[10] Urick R.J., "Principles of Underwater Sound", McGraw-Hill, New York, 1983.

[11] Jesus S., Silva A. e Soares C., "INTIFANTE'00 Sea Trial Data Report - Events I,II and III", SiPLAB Rep 02/01, May 2001.

[12] Scheid F., "Análise Numérica", McGraw-Hill, 2a. edição, Lisboa, Portugal, 1991.

Appendix A

Numerical calculation of dynamic modes

The particular cases of the Sturm-Liouville Problem discussed in sections 2.2 and 2.3 can be solved by replacing the respective differential equation, with a linear system of algebraic equations, leading to a classical (and simpler) problem of finding a set of eigenvectors and eigenvalues [12]. That substitution is described in this appendix.

First, by introducing the following notation:

$$\begin{array}{l} x = z \ , \quad y = \Psi_m \ , \\ a = 0 \ , \quad b = D \ , \end{array} \quad \lambda = \left\{ \begin{array}{c} C_m^{-2} \\ \\ (k_h^2)_m \end{array} \right. \ , \ f(x) = \left\{ \begin{array}{c} N^2 \\ \\ \dfrac{N^2 - \tilde{\omega}^2}{\tilde{\omega}^2 - f_c^2} \end{array} \right. \ , \tag{A.0-0.1}$$

one can rewrite any of the equations Eq.(2.14) or Eq.(2.24) in the form:

$$\frac{d^2 y}{dx^2} + \lambda f(x) y = 0 \ . \tag{A.0-0.2}$$

Furthermore, the boundary conditions can be written, in a general form, as

$$\alpha_1 y(a) + \beta_1 \left. \frac{d^2 y}{dx^2} \right|_{x=a} = 0 \ , \quad \text{and} \quad \alpha_2 y(b) + \beta_2 \left. \frac{d^2 y}{dx^2} \right|_{x=b} = 0 \ . \tag{A.0-0.3}$$

By discretizing the values of the independent variable:

$$x_j = jh + a \ , \ h = \frac{b-a}{N+1} \ , \quad \text{with} \quad j = 0, 1, 2, \ldots, N+1 \ , \tag{A.0-0.4}$$

approximating the second order derivative, within the grid $\{x_j\}$, using finite differences:

$$\left. \frac{d^2 y}{dx^2} \right|_{x=x_j} \approx \frac{y_{j+1} - 2y_j + y_{j-1}}{h^2} \ , \tag{A.0-0.5}$$

discretizing the values of the linear term:

$$\lambda f(x) y|_{x=x_j} = \lambda f(x_j) y_j = \lambda f_j y_j \ , \tag{A.0-0.6}$$

and approximating the boundary conditions, Eq.(A.0-0.3), as

$$\alpha_1 y_0 + \beta_1 \frac{y_1 - y_0}{h} = 0 \ , \quad \text{and} \quad \alpha_2 y_{N+1} + \beta_2 \frac{y_{N+1} - y_N}{h} = 0 \ , \tag{A.0-0.7}$$

it is possible to obtain the following system of linear equations:

$$y_{j-1}\left(\frac{1}{h^2}\right) + y_j\left(-\frac{2}{h^2}\right) + y_{j+1}\left(\frac{1}{h^2}\right) = -y_j \lambda f_j \ . \tag{A.0-0.8}$$

For the particular cases $j = 1$ e $j = N + 1$, and taking into account the pair of equations Eq.(A.0-0.7) one can obtain that

$$y_1\left(\frac{1}{h^2}\right)\left(\frac{-\beta_1}{\alpha_1 h - \beta_1} - 2\right) + y_2\left(\frac{1}{h^2}\right) = -y_1 \lambda f_1 \ , \tag{A.0-0.9}$$

and

$$y_{N-1}\left(\frac{1}{h^2}\right) + y_N\left(\frac{1}{h^2}\right)\left(\frac{\beta_2}{\alpha_2 h + \beta_2} - 2\right) = -y_N \lambda f_N \ . \tag{A.0-0.10}$$

As one can see from the last pair of equations the particular choice of boundary conditions $\alpha_1 h - \beta_1 = 0$, or $\alpha_2 h + \beta_2 = 0$, will *apriori* make it impossible to find a numerical solution of Eq.(A.0-0.2). By introducing the vector $\mathbf{y} = [\ y_1\ y_2\ y_3\ \ldots\ y_N\ ]^{\mathbf{t}}$ one can write the system of linear equations (A.0-0.8) in the following compact form:

$$\mathbf{A}\mathbf{y} = \lambda \mathbf{B}\mathbf{y} \ , \tag{A.0-0.11}$$

where the only non-zero elements of matrices $\mathbf{A}$ and $\mathbf{B}$ are

$$\begin{aligned} a_{11} &= \frac{1}{h^2}\left(\frac{-\beta_1}{\alpha_1 h - \beta_1} - 2\right) \ , \ a_{12} = \frac{1}{h^2} \ , \\ a_{j-1,j} &= \frac{1}{h^2} \ , \ a_{jj} = -\frac{2}{h^2} \ , \ a_{j,j+1} = \frac{1}{h^2} \ , \\ a_{N-1,N} &= \frac{1}{h^2} \ , \ a_{NN} = \frac{1}{h^2}\left(\frac{\beta_2}{\alpha_2 h + \beta_2} - 2\right) \ , \end{aligned} \tag{A.0-0.12}$$

and

$$b_{jj} = -\lambda f_j \ . \tag{A.0-0.13}$$

The solution of Eq.(A.0-0.11) corresponds to a general case of finding the eigenvectors $\mathbf{y}$, and eigenvalues, λ, of the matrices $\mathbf{A}$ and $\mathbf{B}$. This problem can be efficiently solved using MATLAB built-in functions.

Appendix B

M-files

This appendix contains the main M-files used in the calculation of function derivatives, and calculation of hydrostatic normal modes. Each M-file contains a brief description of the input and output parameters.

```
%Universidade do Algarve
%INTIMATE Project
%13/10/98 ( 13:00 )
%Written by TORDAR
%
%Locates minimum values and positions from function.
%
%SYNOPSIS:  [ xmin , ymin , indexes ] = getmin( x , y )
%
%See also getmax, getpeaks, getzeros and maxfxy

function [ positions , minimae , indexes ] = getmin( x , y )

j = 0 ; k = 0 ;

positions = [ ] ; minimae = [ ] ; indexes = [ ] ;

length_samples = length( y ) ;

for i = 2:length_samples-1

    if ( y( i-1 ) > y( i ) )&( y( i ) < y( i + 1 ) ),

       j = j + 1 ;
       minimae(   j ) = y( i ) ;
       positions( j ) = x( i ) ;
       indexes(   j ) =    i   ;

    end  % if

end % for i = 2:length_samples-1
```

```
%Universidade do Algarve
%INTIMATE Project
%19/10/98 ( 18:30 )
%Written by TORDAR
%
%Locates maximum values and positions from function.
%
%SYNOPSIS:  [ xmax , ymax , indexes ] = getmax( x , y )
%
%See also getmin, getpeaks, getzeros and maxfxy

function [ positions , maximae , indexes ] = getmax( x , y )

j = 0 ; k = 0 ;

positions = [ ] ; maximae = [ ] ; indexes = [ ] ;

length_samples = length( y ) ;

for i = 2:length_samples-1

    if ( y( i-1 ) < y( i ) )&( y( i ) > y( i + 1 ) ),

       j = j + 1 ;
       maximae(   j ) = y( i ) ;
       positions( j ) = x( i ) ;
       indexes(   j ) =    i   ;

    end  % if

end % for i = 2:length_samples-1
```

```
%Adds point y using point x and using linear interpolation.
%
%SYNOPSIS:  [newx,newy,y0]  = addp(x,y,x0)
%

function [newx,newy,y0] = addp(x,y,x0)

newx = [ ] ; newy = [ ] ; y0   = [ ] ;

[m,n]   = size(   y ) ;
lengthy = length( y ) ;

minx = min( x ) ;
maxx = max( x ) ;

if ( x0 < minx(1) )|( x0 > maxx(1) ),
   disp( 'Point x0 out of the interval [xmin xmax]! aborting...' ), break
end

index = find( x == x0 ) ;

if isempty( index ) == 1 ,

    index = max( find( x < x0 ) )       ;
   deltax =   x( index+1 ) - x( index ) ;
   deltay =   y( index+1 ) - y( index ) ;
   tangens_alpha = deltay/deltax         ;
   yatx0        =  y(   index ) + (  x0 - x( index ) )*tangens_alpha ;

   if m == 1

      newx      = [x( 1:index ) ,    x0 , x( index+1:lengthy )] ;
      newy      = [y( 1:index ) , yatx0 , y( index+1:lengthy )] ;

   else

      newx      = [x( 1:index ) ;    x0 ; x( index+1:lengthy )] ;
      newy      = [y( 1:index ) ; yatx0 ; y( index+1:lengthy )] ;

   end

   y0           = yatx0 ;

else

   %disp( 'Requested point already included in vector x!' ),
    newx  = x ;
    newy  = y ;
    index = find( x == x0 ) ;
    y0    = y( index(1) ) ;

end
```

```
%DYDX: Calculates approximated derivative
%
%SYNOPSIS: yprime = dydx( x , y )
%
%

function dydx = dydx( x , y ) ;

lengthx = length( x ) ;

diffx = diff( x ) ;
diffy = diff( y ) ;

diffy_over_diffx = diffy./diffx ;

for i = 2:lengthx-1
    [dummy1,dummy2,dydx(i)] = addp(x(1:lengthx-1)+diffx/2,diffy_over_diffx,x(i)) ;
end

x1 = x(1) + diffx(1)/2   ; x2 = x(2) + diffx(2)/2   ;
y1 = diffy_over_diffx(1) ; y2 = diffy_over_diffx(2) ;

tangens = ( y2 - y1 )/( x2 - x1 ) ;

dydx(1) = dydx(2) - ( x1 - x(1) )*tangens ;

x1 = x(lengthx-2) + diffx(lengthx-2)/2 ; x2 = x(lengthx-1) + diffx(lengthx-1)/2 ;
y1 = diffy_over_diffx(    lengthx-2)   ; y2 = diffy_over_diffx(    lengthx-1)   ;

tangens = ( y2 - y1 )/( x2 - x1 ) ;

dydx(lengthx) = dydx(lengthx-1) + ( x(lengthx) - x2 )*tangens ;

[m,n] = size( y ) ;

if n == 1 , dydx = dydx' ; end
```

```
%Universidade do Algarve
%INTIMATE Project
%20/01/2000 ( 23:00 )
%Written by TORDAR
%
%Calculates Hydrostatic Normal Modes.
%
%SYNOPSIS:  [ Modes , Cm ] = gethnms( N , z , nm , topbc , bottombc )
%
%           Where z and N are the column vectors of depths and buoyancy,
%           nm is the requested number of hnms ( nm < length( z ) ) and
%           topbc/bottombc is a two-element vector containing the
%           coefficients of the boundary conditions (BC) at top/bottom,
%           i.e.,
%
%               topbc = [ alpha1 beta1 ], bottombc = [ alpha2 beta2 ]
%
%           where
%
%                    alpha1*hnm(0) + beta1*dhnm/dz(0) = 0 and
%                    alpha2*hnm(D) + beta2*dhnm/dz(D) = 0 .
%
%           For instance, topbc = bottombc = [ 1 0 ] corresponds to
%
%                          hnm(0) = hnm(D) = 0 .
%
%           Be careful with the choice of BC: the existence of the
%           numerical solution depends on them!
%           Each of the nm columns of the matrix Modes corresponds to
%           a requested hnm.
%           The modes are sorted by the number of zero crossings.
%           Cm is the column vector of phase velocities.
%           Use equally spaced depths.

function [ Modes , Cm ] = gethnms( N , z , nm , tbc , bbc )

Modes = [ ] ;
Cm    = [ ] ;

%Check for zero depth:

if z(1) ~= 0 , disp( 'First depth should be zero!' ) , break , end

%Calculate depth step:

h = z(2) ;

%Get top and bottom BC:

alpha1 = tbc(1) ;
 beta1 = tbc(2) ;
alpha2 = bbc(1) ;
 beta2 = bbc(2) ;
```

```
%Don't borrow me with null BC:

if ( alpha1 == 0 )&( beta1 == 0 ),
disp( 'Ill top BC, hasta la vista baby...' ), break,
end

if ( alpha2 == 0 )&( beta2 == 0 ),
disp( 'Ill bottom BC, hasta la vista baby...' ), break,
end

%Avoid paranoic cases:

if ( beta1 ==   alpha1*h ),
disp( 'Ooops! alpha1*h = beta1, try again...' ), break,
end

if ( beta2 == - alpha2*h ),
disp( 'Ooops! alpha2*h = -beta2, try again...' ), break,
end

%Everything is O.K.? let's go andando:

%We need squared buoyancy:

N2 = N.*N ;

lengthz = length( z ) ;
lengthA = lengthz - 2 ;

%Vectors and Matrices allocation:

y = zeros( lengthz , 1 ) ;
A = zeros( lengthA ) ;
B = zeros( lengthA ) ;

%You will need this also:

Au = A ;
Al = A ;

ncdiag =  1/h^2 * ones( 1 , lengthA-1 ) ;
 cdiag = -2/h^2 * ones( 1 , lengthA   ) ;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%This is standard allocation:

%for i = 1:lengthA
%
%   B(i,i) = -N2(i+1) ;
%
%end
```

```
%for i = 2:lengthA-1
%
%   A( i , i   ) = -2/h^2 ;
%   A( i , i-1 ) =  1/h^2 ;
%   A( i , i+1 ) =  1/h^2 ;
%
%end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%This is valid for tbc = bbc = [ 1 0 ]:
%A(1,1) = -2/h^2 ;
%A(1,2) =  1/h^2 ;
%A(lengthA,lengthA-1) =  1/h^2 ;
%A(lengthA,lengthA  ) = -2/h^2 ;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%This is allocation a la Matlab:

B = diag( -N2( 2:lengthz-1 ) ) ;

A  = diag(  cdiag  )     ; % central diagonal
Au = diag( ncdiag ,  1 ) ; % Upper diagonal with 1/h^2
Al = diag( ncdiag , -1 ) ; % Lower diagonal with 1/h^2

A = A + Au + Al ;

%Update first and last diagonal elements according to top and bottom BC:

A(   1    ,   1   ) =  1/h^2 * ( ( -beta1/h )/( alpha1 - beta1/h ) - 2 )  ;
A(lengthA,lengthA) =  1/h^2 * ( (  beta2/h )/( alpha2 + beta2/h ) - 2 )  ;

rankA = rank( A ) ;

if rankA < lengthA ,
disp( 'There is no numerical solution for these BC...' ), break,
end

%O.K., go and get the modes:

[ Modes , D ] = eig(A,B) ;

d  = diag( D ) ;
Cm = 1 ./sqrt( d ) ;

Modes_at_top    = Modes( 1       , : )*( -beta1/h )/( alpha1 - beta1/h ) ;
Modes_at_bottom = Modes( lengthA , : )*(  beta2/h )/( alpha2 + beta2/h ) ;

Modes = [ Modes_at_top ; Modes ;  Modes_at_bottom ] ;

%Arrange Cms from max to min:

[ Cms , indexes ] = sort( Cm ) ;
```

```
Cms = Cm( indexes ) ;
Cm  = flipud( Cms ) ;
Cm  = Cm( 1:nm )    ;
Cm  = real( Cm )    ;

Modes = Modes( : , indexes ) ;
Modes = fliplr( Modes ) ;
Modes = real( Modes( : , 1:nm ) ) ;

%Check for polarity:

%Verify if 1st mode has more than one extremum:

first_mode = Modes( : , 1 ) ;

[posmax,maxvalues] = getmax(z,first_mode) ;
[posmin,minvalues] = getmin(z,first_mode) ;

% (1) 1st. mode has one positive extremum
% (2) 1st. mode has one negative extremum

% (a) 1st. mode has two or more (?) extrema
% (b) Positive extremum goes first.
% (c) Negative extremum goes first.

if     ( isempty( posmax ) ~= 1 )&( isempty( posmin ) == 1 ) % (1)
       Modes( : , 1 ) =  first_mode ;
elseif ( isempty( posmin ) ~= 1 )&( isempty( posmax ) == 1 ) % (2)
       Modes( : , 1 ) = -first_mode ;
else   z1 = posmax(1) ; z2 = posmin(1) ; % (a)
       if     z1 < z2 , Modes( : , 1 ) =   first_mode ; % (b)
       elseif z1 > z2 , Modes( : , 1 ) =  -first_mode ; % (c)
       else   disp('Abnormal situation with zmin(1) = zmax(1)!!!!'), break
       end
end

if nm > 1
   for i = 2:nm
       selected_mode = Modes( : , i ) ;
       [posmax,maxvalues] = getmax(z,selected_mode) ;
       [posmin,minvalues] = getmin(z,selected_mode) ;
       z1 = posmax(1) ; z2 = posmin(1) ;
       if     z1 < z2 , Modes( : , i ) =   selected_mode ;
       elseif z1 > z2 , Modes( : , i ) =  -selected_mode ;
       end
   end
end

%Do not forget to normalize your modes and
%be careful with the units of N!
```