Bulletin of the Seismological Society of America, Vol. 71, No. 3, pp. 613–654, June 1981

SURFACE-WAVE DISPERSION COMPUTATIONS: RAYLEIGH WAVES ON A SPHERICAL, GRAVITATING EARTH

BY F. SCHWAB, J. FREZ, G. F. PANZA, A. H. LIAO, AND E. G. KAUSEL

ABSTRACT

In this algorithmic and numerical analysis of Rayleigh-wave dispersion computations on a spherical, gravitating earth, we first simplify and optimize the existing algorithms based on the direct, Alterman-Jarosch-Pekeris (AJP) formulation; compare computation speed with the fastest existing methods; investigate the relation between integration step size when using direct-integration methods, and layer thickness when using homogeneous-layer methods; and give specific details concerning numerical limitations. Even in highly optimized form, the AJP, direct-integration formulation is slow relative to the best of the techniques based on the flat-, homogeneous-layer approximation. This motivates the development of an improved computational algorithm given as the final result of this report. Loss-of-precision problems appear to be intrinsic to the AJP formulation. At a fixed period, this results in the attainable accuracy of the phase velocity decreasing as radial mode number increases; for fixed accuracy in the phase velocity, as period decreases the maximum mode number that can be treated successfully decreases. Specific numerical relationships among period, mode number, and attainable accuracy are presented.

For *gravitating* structures, a solution to this loss-of-precision problem is given which is based on Gram-Schmidt orthogonalization. Our analytical and numerical studies show that: the dispersion function—roots of which determine dispersion properties—is invariant under orthogonalization, which ensures a smooth, regular variation of this function as roots are sought; the required modifications that orthogonalization introduces into the computation of actual components of motion are quite straightforward (full details are given); and, the addition of orthogonalization increases computation cost by only 5 to 10 per cent. Down to periods of 10 sec, for all radial mode numbers up to 90 to 100, the numerical tests indicate that orthogonalization brings the loss-of-precision problem under *complete* control. All analyses in our investigation show that this problem exists, with the same properties, when integrating the equations of motion in either upward or downward directions; there is no preferred direction of integration relative to this difficulty. Likewise, there is no preferred direction relative to accuracy or overflow problems. In fact, the overflow features are shown to be *precisely* the same, in form and magnitude, for the two directions. The advantage that we *do* find for downward integration is the degree to which it simplifies computational algorithms.

The problem of Rayleigh-wave propagation on a gravitating, spherical earth is characterized by a linear system of six differential equations in six unknowns; for a nongravitating earth this system reduces to four equations in four unknowns. We have succeeded in reducing the sixth-order system to fourth order, while retaining the effect of gravity. The speed of computation for a gravitating earth is thereby increased to about that of the nongravitating case, i.e., to $\frac{4}{5}$ that for nongravitating spheres and to $\frac{2}{3}$ the speed of the fastest of existing algorithms for flat, nongravitating structures treated with the homogeneous-layer approximation. Even greater speeds are possible with the new algorithm if delta matrices are used to control precision loss.

1. INTRODUCTION

Dorman *et al.* (1960) described the use of an electronic computer to calculate surface-wave dispersion for multi-layered, perfectly elastic half-spaces. Their com-

putations were based on the technique devised by Thomson (1950) and Haskell (1953). Press *et al.* (1961) also used the Thomson-Haskell technique, and with a more advanced computer, greatly improved the speed of computation. Randall (1967) later applied Knopoff's (1964) method to this problem and reported a further improvement in speed for the Rayleigh-wave case. In a later series of papers, Schwab (1970) and Schwab and Knopoff (1970, 1971, 1972, 1973) improved the optimization, for computer application, of both the Thomson-Haskell technique and Knopoff's method for flat, multi-layered media. These last papers also provide complete details for obtaining full control over the accuracy of the computations, and for generalizing the algorithms to include computation of attenuation due to the anelasticity of the earth.

For Love waves, the use of spherical-to-flat structure transformations (Biswas and Knopoff, 1970; Schwab and Knopoff, 1971, 1972, 1973; Kausel and Schwab, 1973) makes it possible to carry out all spherical dispersion, attenuation, and focal-response problems using the optimized algorithms for flat structures. Several attempts have been made to develop the same type of transformation for Rayleigh-wave computations (Alterman *et al.*, 1961; Bolt and Dorman, 1961; Biswas, 1972; Schwab and Knopoff, 1972), but these have all yielded only empirical results which lack general applicability. Thus, at the present time at least, it appears that one cannot apply transformation theory to Rayleigh-wave dispersion computations on any arbitrary, spherical, gravitating earth. Bhattacharya's (1976) recent results—although we will not pursue this approach in the present paper—suggest the feasibility of an interesting new procedure for treating spherical, gravitating structures: gravitation alone might be handled by transformation techniques, while Bhattacharya's approach could be used to optimize the treatment of sphericity.

Since transformation techniques have not yet been fully successful in allowing us to apply the fast, flat-structure algorithms to Rayleigh-wave computations with spherical, gravitating structures, in this paper we return to the basic spherical formulation, i.e., that given by Alterman-Jarosch-Pekeris (AJP) (1959). Our primary purpose is the optimization and improvement of this formulation to the point where the efficiency and control of accuracy approach those of the best algorithms for flat, homogeneous-layered, nongravitating structures. The main application that we envision for this improved algorithm is the synthesis of relatively short-period, complete theoretical seismograms for the spheroidal-wave components of motion; for torsional waves, such seismograms are already being computed and compared directly with records from the long-period instruments of the WWSSN (Liao *et al.*, 1977, 1978). Our secondary purpose is to present—we believe for the first time—an explicit, quantitative comparison of the relative efficiencies of the two basic techniques for performing surface-wave dispersion computations: that in which an exact structural specification is employed with approximate mathematical methods, and that in which exact analytical techniques are applied to an approximate model of the structure, i.e., where the structure is replaced by a sequence of homogeneous layers.

The main numerical limitation of the direct, AJP formulation is an intrinsic loss-of-precision problem, a solution to which is required if we are to compute complete synthetic seismograms for direct comparison with records from the WWSSN. Our previous work (Liao *et al.*, 1977, 1978) has shown that this will require an effective treatment down to a period of 10 sec, for radial mode numbers up to about 90 to 100. Our detailed, numerical description of this precision loss is followed by an analysis of its cause. The basic task of these dispersion computations is the numerical integration, over depth, of three independent vectors, and the precision loss appears

to be equivalent to a persistent tendency of these "independent" vectors to become parallel. Over a certain range of depths, the vector solutions contain both exponential and oscillatory terms; the combination of a finite-precision computer and the dominance of exponential over oscillatory depth dependence leads to only this dominant tendency remaining in the solutions, which engenders the spurious parallelism. To avoid this, the first solution we have studied is the combination of Gram-Schmidt orthogonalization with the actual integration of the equations of motion. As a second approach to the problem we present the formalism of the practical solution by delta matrices.

The problem of Rayleigh-wave propagation on a gravitating, spherical earth is characterized by a linear system of six differential equations in six unknowns; for a nongravitating earth this system reduces to four equations in four unknowns. The main source of dissatisfaction with computations for gravitating structures is the slow speed of the present algorithms. Even in our most efficient optimization, the full, sixth-order formulation is about five times slower than the comparable algorithm for Love (torsional) waves. An improved algorithm has therefore been developed to increase the speed of computation for these Rayleigh-wave calculations. In this improvement, the sixth-order formulation is reduced to one of fourth order, i.e., although the effect of gravity is retained, the formulation is reduced to essentially that for nongravitating structures.

2. AJP Formulation

We begin with an analysis, simplification, and optimization of this formulation, the specific purposes being: (1) to report on our initial study of the optimization of the *direct* computations (see Wiggins, 1976, for a discussion of computations based on the *variational* technique); (2) to report the results of our study concerning accuracy considerations; and (3) to determine relative efficiencies of computations including gravity, and the analogous calculations for nongravitating structures. Also, a new computational algorithm is developed for the *direct* calculation of group velocities and partial derivatives of the phase velocity with respect to the structural variables; this direct technique, unlike variational methods, does not depend on the numerical evaluation of "energy integrals."

The basic formulation for our problem (Pekeris and Jarosch, 1958) involves the solution of three second-order, ordinary differential equations constrained by a set of boundary conditions. For purposes of numerical solution, it is advisable to reduce this system to six, linear, first-order differential equations, as was done by Alterman *et al.* (1959). Bolt and Dorman (1961) applied this formulation to the evaluation of Rayleigh-wave dispersion, and reported on those numerical details which it was economically feasible to investigate with second-generation computing equipment. Detailed algorithmic testing of accuracy, precision, and efficiency characteristics really requires the present, third-generation machinery, which we have employed in the current study; the work we report here can be considered as a logical extension, for Rayleigh waves, of the above series of papers.

To sketch the AJP formulation, if we let $\bar{y}_i = dy_i/dr$, where r is the distance from the center of the Earth, then the sixth-order system is

$$\begin{bmatrix} \bar{y}_1 \\ \bar{y}_2 \\ \bar{y}_3 \\ \bar{y}_4 \\ \bar{y}_5 \\ \bar{y}_6 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & a_{24} & 0 & a_{26} \\ -a_{33} & 0 & a_{33} & a_{34} & 0 & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & 0 \\ a_{51} & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & a_{63} & 0 & a_{65} & a_{66} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} \tag{2.1}$$

with y_1 and y_3 related to the components of displacement $u_r(r, \theta, \phi)$, $u_\theta(r, \theta, \phi)$, and $u_\phi(r, \theta, \phi)$ by

$$u_r = y_1(r)\, \chi_l^m(\theta)\, e^{\mathrm{im}\phi} e^{i\omega t}$$

$$u_\theta = y_3(r)\, \frac{d}{d\theta} \chi_l^m(\theta)\, e^{\mathrm{im}\phi} e^{i\omega t}$$

$$u_\phi = \frac{\mathrm{im}}{\sin\theta} y_3(r) \chi_l^m(\theta)\, e^{\mathrm{im}\phi} e^{i\omega t}. \tag{2.2}$$

For propagating surface waves diverging from the epicenter,

$$\chi_l^m = \frac{1}{2}\left(P_l^m + i\frac{2}{\pi} Q_l^m \right). \tag{2.3}$$

For waves converging toward the epicenter,

$$\chi_l^m = \frac{1}{2}\left(P_l^m - i\frac{2}{\pi} Q_l^m \right). \tag{2.4}$$

For a treatment of the situations that require the use of (2.3), (2.4), or their sum, see Schwab and Kausel (1976). In this same reference, the justification is given for our major departure from previously reported computations of Rayleigh-wave dispersion on a sphere: strictly speaking, these solutions only exist on a sphere at the discrete set of frequencies corresponding to integral values of the polar order number l. However, fixing l and evaluating the corresponding angular frequency ω does not yield the dispersion data at equal frequency intervals, which we desire to use in the usual numerical technique for obtaining time series by inverse Fourier transformation. Schwab and Kausel (1976) have shown that, for most practical applications of propagating surface waves, nonintegral l at equally spaced frequencies can be used without introducing significant errors; therefore, we adopt the procedure of fixing ω and computing l, or

$$c = a\omega/(l + \tfrac{1}{2}), \tag{2.5}$$

where a is the radius of the earth. The relation between c and the true spherical phase velocity is also treated by Schwab and Kausel. In equation (2.1), y_2 and y_4 are, respectively, the radial dependencies of the rr, and the $r\theta$ and $r\phi$ components of stress; y_5 and y_6 arise from the presence of the gravitational field.

To construct the simplest possible algorithm for performing the required numerical integration, we have chosen to proceed from the free surface downward. The integration is then carried down to a depth sufficient to make it immaterial—to σ significant figures in l or c—just how we terminate the integration: e.g., with an approximation of a free or rigid surface. The fact that such a termination process is valid has been checked by extensive numerical tests in the course of this work. These tests follow the lines of the layer-reduction experiments described by Schwab and Knopoff (1970, 1972), and are discussed in the following subsection on "Terminating Boundary Conditions." In this discussion, we also describe the termination of the structure at depth by either a solid or liquid, homogeneous, gravitating sphere.

Here, we should point out that the warnings given in the literature against proceeding downward from the free surface when integrating the system of differential equations, or when forming the layer-matrix product if applying the Thomson-

Haskell technique or Knopoff's method, do not appear to be justified by our experience. In our long experience with matrix methods applied to surface-wave dispersion computations, the formation of matrix products upward toward the free surface was not found to have any advantage over formation of the product in the downward direction. In fact, when treating Love, or torsional waves on a sequence of homogeneous layers, it is important for algorithmic efficiency that it be possible to form matrix products in *both* directions at the same time. This permits us to avoid $[2 \times 2][2 \times 2]$ matrix products when evaluating group velocities [equations (18) to (25), Schwab and Knopoff (1972)] or partial derivatives in general. With Rayleigh waves on a sequence of homogeneous layers, formation of these products in both directions at once allows one to avoid (the equivalent of) $[5 \times 5][5 \times 5]$ products in group velocity and partial derivative computations [equations (63) to (66) for downward formation, equations (92) to (98) for upward formation of the products, Schwab and Knopoff (1972)]. Our programs based on concurrent, two-directional formation of matrix products have received much testing in practical application for some years now without exhibiting any problems. In the work upon which we report herein, downward integration did not give rise to any difficulties not experienced when using algorithms based on upward integration. This statement is the result of comparative numerical tests carried out with each important feature of the computational process: overflow, loss of precision, and accuracy of calculated phase velocity. In each comparison the feature was investigated with a program based on downward integration, and then with another utilizing upward integration. The overflow feature is precisely the same in form and magnitude for both directions of integration (see following main section for details). In our numerical investigation of the loss-of-precision problem (subsection on "Existence of Solution"), the results have the same character in the two cases; no reason for preferring either direction of integration is apparent from these results. The final test of absolute accuracy—at the four-significant-figure level—is a direct comparison of dispersion results from both types of programs; and, since the results agree to the specified accuracy, there is no obvious preference to be inferred from this feature. The important advantage that we have found for downward integration is the degree to which it simplifies computational algorithms, such as those described in the following few paragraphs.

To begin, we consider a continental structure. In this case y_2 and y_4 vanish, and $y_6 = -y_5\,(l+1)/a$ at $r = a$. thus, in a manner similar to that of Pekeris (1966), we can write the starting vector as

$$Y_S(a) = \begin{bmatrix} y_1(a) \\ y_2(a) \\ y_3(a) \\ y_4(a) \\ y_5(a) \\ y_6(a) \end{bmatrix} = \begin{bmatrix} y_1(a) \\ 0 \\ y_3(a) \\ 0 \\ y_5(a) \\ -y_5(a)(l+1)/a \end{bmatrix} \tag{2.6}$$

$$= y_1(a) \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + y_3(a) \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + y_5(a) \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ -(l+1)/a \end{bmatrix}$$

or

$$Y_S(a) = y_1(a)X_1(a) + y_3(a)X_2(a) + y_5(a)X_3(a), \tag{2.7}$$

and for $r < a$

$$Y_S(r) = y_1(a)X_1(r) + y_3(a)X_2(r) + y_5(a)X_3(r). \tag{2.8}$$

The three quantities that are unknown—$y_1(a)$, $y_3(a)$, and $y_5(a)$—can be carried implicitly in the computations, while we integrate the vectors whose starting values are known exactly: X_1, X_2, and X_3. That is, we integrate to obtain X_1 at depth; this is repeated, in turn, with X_2 and X_3. Thus we actually use equation (2.1) in the form $\dot{X}_i = AX_i$ to integrate from the surface $r = a$, to the depth at which the boundary conditions are to be applied: $r = r_0$, where we can again express Y_S in terms of the undetermined coefficients by using (2.8).

If we define a rigid boundary at depth by

$$y_1(r_0) = y_3(r_0) = y_5(r_0) = 0, \tag{2.9}$$

we then obtain three linear, homogeneous equations in three unknowns—the undetermined coefficients—and the determinant of the coefficient matrix must vanish if we are to have a nontrivial solution. Thus the dispersion function takes the form

$$F_A(c, \omega) = \begin{vmatrix} [X_1(r_0)]_1 & [X_2(r_0)]_1 & [X_3(r_0)]_1 \\ [X_1(r_0)]_3 & [X_2(r_0)]_3 & [X_3(r_0)]_3 \\ [X_1(r_0)]_5 & [X_2(r_0)]_5 & [X_3(r_0)]_5 \end{vmatrix}, \tag{2.10}$$

zeros of which define valid (c, ω) dispersion pairs. For the two approximations to free boundaries at depth, we have used the definitions

$$y_2(r_0) = y_4(r_0) = 0 \tag{2.11}$$

and either

$$y_6(r_0) = -y_5(r_0)(l + 1)/r_0 \tag{2.12}$$

or

$$y_6(r_0) = -y_5(r_0)(l + 1)/a, \tag{2.13}$$

which yield, respectively, dispersion functions $F_B(c, \omega)$

$$\begin{vmatrix} [X_1(r_0)]_2 & [X_2(r_0)]_2 & [X_3(r_0)]_2 \\ [X_1(r_0)]_4 & [X_2(r_0)]_4 & [X_3(r_0)]_4 \\ [X_1(r_0)]_6 + \dfrac{l+1}{r_0}[X_1(r_0)]_5 & [X_2(r_0)]_6 + \dfrac{l+1}{r_0}[X_2(r_0)]_5 & [X_3(r_0)]_6 + \dfrac{l+1}{r_0}[X_3(r_0)]_5 \end{vmatrix}, \tag{2.14}$$

and $F_C(c, \omega)$

$$\begin{vmatrix} [X_1(r_0)]_2 & [X_2(r_0)]_2 & [X_3(r_0)]_2 \\ [X_1(r_0)]_4 & [X_2(r_0)]_4 & [X_3(r_0)]_4 \\ [X_1(r_0)]_6 + \dfrac{l+1}{a}[X_1(r_0)]_5 & [X_2(r_0)]_6 + \dfrac{l+1}{a}[X_2(r_0)]_5 & [X_3(r_0)]_6 + \dfrac{l+1}{a}[X_3(r_0)]_5 \end{vmatrix}. \tag{2.15}$$

Next, we consider an oceanic structure. In this case, the analog of equation (2.1) is, for the homogeneous, oceanic (liquid) layer,

$$\begin{bmatrix} \bar{y}_1 \\ \bar{y}_2 \\ \bar{y}_5 \\ \bar{y}_6 \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{15} & 0 \\ b_{21} & b_{22} & b_{25} & -\rho \\ 4\pi G\rho & 0 & 0 & 1 \\ b_{61} & b_{62} & b_{65} & b_{66} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_5 \\ y_6 \end{bmatrix}. \tag{2.16}$$

At $r = a$, y_2 vanishes and $y_6 = -y_5\ (l + 1)/a$, and we can write the starting vector as

$$Y_L(a) = \begin{bmatrix} y_1(a) \\ y_2(a) \\ y_5(a) \\ y_6(a) \end{bmatrix} = \begin{bmatrix} y_1(a) \\ 0 \\ y_5(a) \\ -y_5(a)(l+1)/a \end{bmatrix} = y_1(a) \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + y_5(a) \begin{bmatrix} 0 \\ 0 \\ 1 \\ -(l+1)/a \end{bmatrix} \tag{2.17}$$

or

$$Y_L(a) = y_1(a)Z_1(a) + y_5(a)Z_2(a), \tag{2.18}$$

and for $r < a$

$$Y_L(r) = y_1(a)Z_1(r) + y_5(a)Z_2(r). \tag{2.19}$$

Again, we carry the unknown quantities—$y_1(a)$ and $y_5(a)$—implicitly, and integrate the vectors whose starting values we know exactly: Z_1 and Z_2, using equation (2.16) in the form $\bar{Z}_i = BZ_i$. On the oceanic side of the liquid-solid boundary at the bottom of the ocean, $r = r_1$, we then have

$$Y_L(r_1) = y_1(a) \begin{bmatrix} [Z_1(r_1)]_1 \\ [Z_1(r_1)]_2 \\ [Z_1(r_1)]_5 \\ [Z_1(r_1)]_6 \end{bmatrix} + y_5(a) \begin{bmatrix} [Z_2(r_1)]_1 \\ [Z_2(r_1)]_2 \\ [Z_2(r_1)]_5 \\ [Z_2(r_1)]_6 \end{bmatrix}. \tag{2.20}$$

At this boundary, y_1, y_2, y_5, and y_6 are continuous, y_4 must vanish, and y_3 is

undetermined. Thus on the solid side of this interface, we have

$$Y_S(r_1) = y_1(a)\begin{bmatrix}[Z_1(r_1)]_1\\ [Z_1(r_1)]_2\\ 0\\ 0\\ [Z_1(r_1)]_5\\ [Z_1(r_1)]_6\end{bmatrix} + y_3(r_1)\begin{bmatrix}0\\ 0\\ 1\\ 0\\ 0\\ 0\end{bmatrix} + y_5(a)\begin{bmatrix}[Z_2(r_1)]_1\\ [Z_2(r_1)]_2\\ 0\\ 0\\ [Z_2(r_1)]_5\\ [Z_2(r_1)]_6\end{bmatrix} \tag{2.21}$$

or

$$Y_S(r_1) = y_1(a)X_1(r_1) + y_3(r_1)X_2(r_1) + y_5(a)X_3(r_1), \tag{2.22}$$

and the integration proceeds and terminates, from r_1 to r_0, exactly as in the continental case.

The length of our complete analysis of the AJP formulation, plus the descriptions of the improvements we have developed to deal with its shortcomings, presents something of a problem. In order to condense this information into a single *Bulletin* article, much of the detailed information that will be needed by those interested in applying this work has had to be included in a set of additional notes on microfiche found in the back of this issue. These notes contain all figures and tables, as well as those details not critical to the development followed in the main text. (Full-size copies of the additional notes are available from the senior author.) Perhaps the most interesting, and potentially useful of these supplemental details is the development of new algorithms for calculating Rayleigh-wave group velocities and partial derivatives with respect to structural parameters (section 1 of the additional notes). These methods are natural extensions of the simplified algorithms given above, and we believe them to be the first *direct* computational techniques for the determination of such group velocities and partial derivatives on a gravitating earth.

Numerical technique for integrating the system of differential equations. Relative to numerical integration, our main purpose is to minimize computation time for practical application of the direct, AJP formulation. The fundamental decision here is the absolute accuracy to be sought in the calculated phase velocities. To decide upon the accuracy that will be required for our application to the synthesis of relatively short-period, complete theoretical seismograms for spheroidal waves, we can refer to the analogous work for torsional waves. From those studies, a four-figure accuracy in c appears to be quite sufficient. (See section 6 of the additional notes for references and further discussion of this point.) We want complete control over accuracy in our present work, but relative to the application planned, we want to specify the accuracy to be *as low as possible.* This will allow us to use the maximum possible step sizes in our numerical integration (for the order of the Runge-Kutta and predictor-corrector methods utilized), thus minimizing the required number of integration steps and expense of computation.

To optimize these computations, it is also useful to constrain the structural specification somewhat. These constraints, and our empirically determined step sizes, are as follows.

1. The liquid, oceanic layer is limited to a single, homogeneous layer. A special Runge-Kutta technique of the fourth order (see below) is used for the first three steps of the integration—step size of about 1 km—and a fourth-order predictor-

corrector method (see below) is employed, if necessary, to continue the integration to the bottom of the oceanic layer (with the same step size).

2. The sedimentary layers are limited to a sequence of homogeneous layers, each of which does not exceed 1 km in thickness. These layers are treated with one, fourth-order, Runge-Kutta step.

3. The subsedimentary crustal layers must also be homogeneous, and each is treated as the oceanic layer; the step size fixed at about 1 km.

4. The sub-Moho mantle requires continuous velocity-depth and density-depth distributions, although discontinuities can be approximated as closely as desired by specifying large gradients. As with the oceanic layer, three Runge-Kutta steps are followed by a fourth-order, predictor-corrector method. The initial step size is 1.5625 km ($12.5/2^3$ km), with which we execute the Runge-Kutta and seven predictor-corrector steps. The step size is then doubled, and five predictor-corrector steps are performed; this procedure is repeated until the step size reaches 12.5 km, and the predictor-corrector method is then applied with this fixed step size.

Comparison of our phase velocities computed with this dependence of step size on depth, with those obtained from a completely independent program, was used as a final test to verify that these steps are sufficient to yield the desired, four-figure accuracy. More precisely, for realistic models of the earth, our comparisons show agreement to an average of 3.7 significant figures; however, at this time there is little purpose in further refining the structural specification to make the slight improvement to 4.0 figures. The currently incomplete state of our analysis of the algorithms for treating the basic numerical difficulty with the original AJP formulation—an intrinsic loss-of-precision problem—makes such a refinement premature. The general features of this difficulty are illustrated by the results of numerical experiments described in the following subsection; an extensive discussion of the problem, from several points of view, is given in the following main section (and associated sections of the additional notes), where a specific difficulty with step size is also pointed out for periods in the decreasing range from 25 to 10 sec; and in the final section, the reported numerical results indicate that loss-of-precision problems become noticeable near 10 sec when computations have only reached the third mode. All of the prior results indicate that new algorithms incorporating loss-of-precision control—orthogonalization or delta-matrix techniques—should be the ones upon which we base any final, precise statement about optimum step distribution with depth.

Concerning the accuracy of four significant figures in computed values of c, one should review the rather detailed treatment given by Schwab and Knopoff (1972, pp. 107–111, 116–118, 140–141, 145), in which piecewise-continuous velocity- and density-depth distributions are treated with the homogeneous-layer approximation. Comparison will show that when four-figure accuracy is specified, the thicknesses of the layers as a function of a depth, in that approximation, are roughly the same as the integration step sizes in the present analysis. That is, to the degree of accuracy with which two such dissimilar methods can be compared, if the step sizes above are used as thicknesses in the homogeneous-layer approximation, that technique will yield four significant figures in the computed values of c. This is a particularly appealing result in that it introduces some degree of unification between the two main approaches to dispersion computations by direct methods.

To start the predictor-corrector method, we have used a Runge-Kutta technique designed specifically for this purpose. Here one is *only* interested in being able to

minimize the bounds on the truncation error. Ralston (1962) has treated this problem, and gives the algorithm (also see section 2 of the additional notes) for obtaining the first four points needed to start the predictor-corrector routine.

The fourth-order, predictor-corrector method we have used (Hamming, 1959) is fully described along with the details concerning doubling of step size, by Ralston (1960). Highly practical details, relative to the combination of these particular Runge-Kutta and predictor-corrector routines, will be found in Anon. (1970). One should be warned, however, that the use of the subroutines described there is inadvisable for our present purposes. The use of these *general purpose* routines can increase computation expense by a factor of 10 to 100 over that of the optimized algorithm.

Optimization of the AJP formulation. The key to optimizing the integration is to apply our knowledge about this particular problem to specify all the depths, r_k, at which $a_{ij}(r)$ are to be evaluated. The evaluation of these elements can then be removed from the innermost, integration loops of the program. The details concerning these depths are contained in the preceding section. In Figure 1, the optimized scheme for the evaluation of $a_{ij}(r_k)$—for the solid sedimentary layers, subsedimentary crustal layers, and mantle—is indicated in outline form. It will be seen that most of the procedure for evaluating these elements can even be removed from within the ω and c loops. In Figure 2, the same information is given for the elements $b_{ij}(r)$ of the matrix describing integration through the liquid, oceanic layer.

Relative to optimization, a few specific programming details are appropriate here since they are critical to minimizing computation time and expense.

In the integration procedures themselves, it is very important to form matrix products, such as those in (2.1) and (2.16), in an explicit manner. This permits full use to be made of the many zero elements, and those that are independent of r, or are equal to another matrix element. For example, the basic AJP matrix multiplication for solid layers is illustrated in Figure 3.

In reports on our earlier work with Love-wave dispersion and dispersion-attenuation computations, for both flat and spherical structures it was possible to give short, key, FORTRAN program segments (Figure 2, Schwab and Knopoff, 1972; Figures 4 and 5, Schwab and Knopoff, 1973). These together with descriptions of root-bracketing and root-refining procedures completely specify the optimization when the multi-, homogeneous-layer approximation is employed. If this approximation is used with Rayleigh waves on flat structures, the optimization can be specified in the same manner (Figures 11 to 13, Schwab and Knopoff, 1972). When employing the method of direct integration of the equations of motion, it is not possible to exhibit the *complete* optimization in this compact, simple manner for our problem of Rayleigh-wave dispersion on a spherical, radially heterogeneous, gravitating earth. However, it *is* possible to present the most important part of the algorithm as a relatively compact program segment. This is given in Figure 4a, which illustrates the predictor-corrector method we have applied from below the Moho to the selected value of r_0; the automatic doubling of integration step size is included. Most of the computation time is spent in this segment, which is entered with

$$\text{COEFF1} = (4/3)\ H$$

$$\text{COEFF2} = 3H$$

$$\text{COEFF6} = (121/36)H$$

$$H = -25/16;$$

the indices for successive step-size regions of the integration are given in Table 1; for the details concerning $B(I, J)$, see the description of subroutine DHPCL (Anon., 1970). In this type of programming there are important, machine-dependent considerations: the manner in which the 6 x 6 matrix elements are stored in memory, and the way in which indices are handled in the program segment given in Figure 4a. In fact, the indices ITP1, ITP3, ITP8, ITP9, ITP10, which are used for compactness in the figure, actually slow computation speed on the IBM 360/91; these subscripts are best used in explicit form, IT +1, IT + 3, IT + 8, IT + 9, IT + 10. The key program segment is given in the form shown in Figure 4a for two reasons: (1) to illustrate the logic as clearly and simply as possible; and (2) to provide an illustrative example of the importance of handling subscripting and storage in the manner most appropriate for a given machine. The time required to execute DO-loop 170 once, specifies the necessary time to perform one integration step for each of the three vectors $\boldsymbol{X}_i$; thus, to complete one integration step in forming the dispersion function, each step of DO-loop 170 must be executed three times. The time for one integration step in forming F is termed the characteristic time τ, which we use to illustrate the importance of correct subscripting and storage. The characteristic time for the segment in Figure 4a is 489×10^{-6} sec/step/iteration. By simply reversing the order of the subscripts of $B(I, J)$, this time is improved by 92×10^{-6} sec/step/iteration; if ITP1, etc., are used explicitly as IT + 1, etc., τ is decreased still further by an amount of 67×10^{-6} sec/step/iteration; and if a_{ij} are stored more efficiently, still another 44×10^{-6} sec/step/iteration can be saved, bringing τ down to 286×10^{-6} sec/step/iteration. DO-loop 170, in a form incorporating the above improvements, is shown in Figure 4b. It should be understood that 286×10^{-6} sec/step/iteration is a *lower bound*; the corresponding *effective* characteristic time—given in (2.24)—must reflect time spent in other parts of this particular test program, which is designed to treat only a few frequencies in a given run. The effective characteristic times of our final routines (fourth section) will be governed by lower bounds, as defined above, since the applications we are mainly interested in involve the treatment of extremely large numbers of frequencies in a single computer run, which means that most of the execution time is spent in DO-loop 170.

Although the present work was carried out with both IBM 360 and 370 equipment: a 360/91 computer in Los Angeles, a 360/65 in Bari, and 370/145 computers in Santiago and Cosenza, all timing results are given here in terms of the first computer. This installation was also used in our final optimization with Rayleigh-wave computations for flat structures (Schwab and Knopoff, 1972), which allows us to make an accurate evaluation of the relative characteristic times for calculations with flat, nongravitating structures, and with spherical, gravitating models. In the former case this time is

$$\text{FLAT}\tau_{\text{RAYLEIGH}} = 110 \times 10^{-6} \text{ sec/layer/iteration} \tag{2.23}$$

(which corresponds to Knopoff's method applied to a sequence of homogeneous layers), and in the latter case,

$$\text{SPHERICAL}\tau_{\text{RAYLEIGH}} = 336 \times 10^{-6} \text{ sec/step/iteration} \tag{2.24}$$

(which corresponds to the optimization of the AJP formulation described above). Since our integration "step" can be considered nearly equivalent to a "layer" in computations based on the homogeneous-layer approximation, and since, to the accuracy possible in this type of comparison, the "iterations" required in the two

cases can be considered equivalent (see Schwab and Knopoff, 1972, for details concerning iteration procedures), the relative efficiencies of the two types of Rayleigh-wave dispersion computation can therefore be determined by simple comparison of their characteristic times; we find that the optimized, sixth-order, AJP formulation is three times slower than the fastest of the Rayleigh-wave algorithms. That is, the time required to integrate each of the three vectors over depth in the spherical, gravitating case, is the same as that required to carry out the analogous operation—the formation of the matrix product—for the flat, nongravitating case.

To obtain a valid comparison of the direct-integration method with the homogeneous-layer technique, clearly we should not include gravity in the former method. The removal of gravity reduces to two the number of vectors that must be integrated, and reduces the number of elementary operations (additions and multiplications) in (2.1) from 34 to 23. Thus the ratio of computation times for nongravitating and gravitating spheres, when treated by the direct-integration method, is approximately $(2 \times 23)/(3 \times 34)$, or, for the nongravitating case,

$$\text{SPHERICAL}\tau_{\text{RAYLEIGH}} = 151 \times 10^{-6} \text{ sec/step/iteration.} \tag{2.25}$$

Thus the direct-integration method, for a nongravitating sphere, is only 36 per cent slower than the optimized computations for a flat, nongravitating structure, where the latter is treated with the homogeneous-layer approximation.

From this result we are led to speculate on the advisability of redirecting attempts to devise Rayleigh-wave transformation techniques. To the present time, these attempts have been focused mainly upon finding a single transformation that will permit spherical, gravitating structures to be treated with computational algorithms for flat, nongravitating models. Perhaps more success could be expected if we attempted to make use of algorithms for nongravitating spheres, and only tried to include gravity by transformational methods.

Another improvement in the *lower bounds* of the AJP characteristic times is possible. The above, limiting bound for the gravitating case was obtained by integrating each of X_i separately; the time is improved to 266×10^{-6} sec/step/iteration if the three vectors are treated simultaneously within DO-loop 170. If the two vectors of the nongravitating case are handled simultaneously within the loop (see Figure 4c), the lower bound of τ becomes 143×10^{-6} sec/step/iteration. These times are the results of actual measurements.

The timing results given in this section are to be considered preliminary; our final, most complete treatment of timing details is given in the fourth section of the main text, and in sections 6 to 8 of the additional notes.

It is obviously of interest, to those involved in practical work of this type, to have some idea of the relative speeds of such computations for the various computers currently in use at the larger installations. For rough estimates of spheroidal-wave computation times, the conversion factors given by Porter *et al.* (1980) for Rayleigh waves can be applied to the characteristic times given here for an IBM 360/91 computer.

Existence of solutions as a function of numerical and algorithmic procedures. On IBM 360 and 370 equipment, large-scale numerical work is routinely carried out in double precision: the equivalent of 16 to 17 decimal digits. Except where indicated otherwise, this precision was used to investigate the existence criteria for solutions from our optimization of the basic AJP formulation.

Our testing procedure followed the lines of the layer-reduction experiments described by Schwab and Knopoff (1972): at each of a set of periods, c is computed for a complete range of terminating values, r_0, for the integration. At each fixed period, by comparing the values of c as a function of r_0, the range of r_0 over which c is stable to four significant figures is immediately evident. In terms of r_0 and period, our results for an oceanic, and a continental shield structure (Figure 5) are given in Figure 6. The fundamental and first seven higher modes are treated in each case. The results are similar to those previously obtained for Rayleigh waves when the homogeneous-layer approximation is employed (Figures 14, 15, 17, Schwab and Knopoff, 1972). At each period, a certain minimum amount of structure (maximum r_0) must be retained to ensure four significant figures in c. This maximum value at r_0 is a physical limitation. For the mode and period of interest, there is significant energy content down to a depth of $a - r_0$, and the structure above this point thus affects all four figures of c.

In the initial program testing it is useful if one can integrate to any depth, irregardless of whether this results in loss of accuracy in the computed phase velocity, or in the expenditure of more computer time than if the integration were terminated at the minimum acceptable depth for the desired accuracy in the phase velocity. The use of a very large value of $a - r_0$, or more precisely, a large number of wavelengths of structure, will result in overflow. Thus a simple, temporary solution to this problem is useful. Such a solution is the extension of the normalization technique described by Schwab and Knopoff (1970, 1972). Application of this technique to the direct-integration procedure is quite simple; it is not necessary to include normalization until predictor-corrector integration has begun in the mantle. To normalize, all one need then do is determine the maximum of the absolute values of y_i at the end of each integration step; one then divides all $y_i(r_j)$ and $\bar{y}_i(r_j)$ by this value, where r_j are the seven positions at which y_i and $\bar{y}_i$ must be specified so as to permit the next step of the fourth-order predictor-corrector method. (Seven, rather than four r_j are required to allow the automatic doubling of step size when certain depths are reached.) For ease of reference, a normalization scheme is given in Figure 7 which is appropriate for the program segment in Figure 4a. There are two warnings. (1) If only sparing use of normalization is planned, or if it is to be invoked from an IF statement, the segment in Figure 7 will be satisfactory; however, if large-scale use is envisioned, efficiency requires the inclusion of normalization directly within the coding of Figure 4a. (2) Unless absolutely necessary, this very powerful form of normalization should not be included in these computations; it can result in quite a significant increase in computation time. An inexpensive, restricted form of normalization will be introduced further on, along with a description of some rather detailed testing of overflow characteristics. Here, we wish only to establish the number of wavelengths of structure that can be retained without overflow occurring. These tests were performed with the average (oceanic) earth structure given by Wiggins (1968). The results are given in Table 2. For IBM 360 and 370 equipment, when using the equivalent of 16 to 17 decimal digits, overflow occurs when $|F| \approx 10^{70}$ to 10^{80}. Although these results show that it is possible to retain up to 8 to 9 wavelengths of structure for the lowest four (radial) mode numbers, they also illustrate the trend of a decreasing allowable number of wavelengths of structure as mode number increases. This is a general trend, and it will force us to include normalization even for routine computations when dealing with relatively large mode numbers.

From the results shown in Figures 6 and 8 we can determine the maximum periods and values of c that can be treated without taking the structure of the core into consideration [other than to determine the values of $g(r)$]. These results are shown in Figure 9, along with the corresponding minimum order number, ℓ. Relative to the computation of theoretical seismograms, the combination of the results in Figure 9 and those given by Schwab and Kausel (section 5, 1976), suggest potentially useful conclusions: (1) when $l > l_{\min}$, only the crust-mantle system need be used in the computations; c can be computed at specified, equally spaced frequencies and inverse Fourier transformation can be used to calculate the theoretical seismogram for this range of periods; and the first term of the asymptotic expansions for P_l^m and Q_l^m can be used (possibly corrected by automatic numerical interpolation from the data in Figures 2 and 3 of Kausel and Schwab, 1976). (2) When $l < l_{\min}$, the core must be included in the computations; ω should be calculated at integral values of l, and summation over l should replace Fourier synthesis; and the exact, integral-l expressions should be used to evaluate the associated Legendre functions.

A point of considerable interest to those involved in practical computation of surface-wave dispersion, is whether or not a difficulty analogous to the Thomson-Haskell "loss-of-precision" problem (Schwab and Knopoff, 1970; Schwab, 1970) is intrinsic to the AJP formulation. In calculations based on the homogeneous-layer approximation, when the original version (Haskell, 1953) of the Thomson-Haskell formulation for Rayleigh waves is used, this problem can cause serious difficulties if the computer is employed in a low-precision mode. To test for an analog of this loss-of-precision problem, in our optimization of the AJP formulation we simulated single-precision (the equivalent of about six decimal digits) computation by replacing DO-loop 160 (Figure 4a) in our double-precision program, with the segment shown in Figure 10. The results of these tests are similar to those from the original Thomson-Haskell formulation (Figure 2, Schwab and Knopoff, 1970), and are illustrated in Figure 11. In the r_0 to σ range shown, where σ is the number of significant figures in the computed phase velocity, our results indicate that there is no problem with modes 0 to 4 in double-precision computations; but when the calculations are reduced to single precision, the loss-of-precision problem is clearly evident. In the latter case, there is seen to exist a minimum value of r_0, below which we cannot go and still retain a given accuracy in the computed phase velocities. Thus the AJP formulation does indeed exhibit the analog of the Thomson-Haskell, loss-of-precision problem.

Since practical work with dispersion computations on IBM 360 and 370 equipment is routinely performed in double precision, it is important to estimate the numerical limitations imposed by loss of precision in this computational mode. The results of our tests at periods of 50 and 25 sec are shown in Figures 12 and 13. Less extensive tests were also carried out at 65 sec. At a given period, the right-most point of each of the smoothed curves was used to determine the maximum accuracy possible for each mode. This information was then collected in Figures 14 and 15. Although the data is necessarily sparse, due to the expense of this type of experiment, the results are quite clear. For a fixed period, as we go to higher and higher (radial) mode numbers, the attainable accuracy in c becomes less and less; for a fixed accuracy in c, as we go to shorter and shorter periods, the maximum mode number that can be treated successfully becomes smaller and smaller.

Our planned synthesis of seismograms for the spheroidal-wave components of motion, will require dispersion information from the gravest period down to a minimum of 10 sec for each of about 90 to 100 radial modes (Liao *et al.*, 1977, 1978).

Thus the above results clearly show that a control feature for loss of precision must be added to the original AJP formulation before these synthetic seismograms can be generated. This addition requires first, however, the development of an improved, high-speed algorithm for spherical, gravitating structures, thus making it economically feasible even to *attempt* such a synthesis. Such an algorithm is developed in the fourth section, which now leaves us with the problem of selecting the best method for controlling loss of precision (see, e.g., Gilbert and Backus, 1966; Wiggins, 1968; Neigauz and Shkadinskaya, 1972; Section II.D.4, Takeuchi and Saito, 1972; Appendix A, Nolet, 1976). In sections 4 and 5 of the additional notes these various control features are discussed in general. The following section of the main text presents a detailed investigation of orthogonalization, which is shown to control loss of precision when gravity is included in the model of the earth. We believe that this is the first such demonstration. In the fourth section of the main text it is shown that the AJP, sixth-order formulation of our problem can be reduced to one of fourth order, while still retaining the effect of gravity. This makes it possible—again, apparently for the first time—to cast the formulation in a practical, delta-matrix form for controlling loss of precision. The most immediate new investigation that is indicated by our present analysis thus appears to be the determination of which of these two methods is preferable in an algorithm for the generation of synthetic seismograms.

Terminating boundary conditions. As noted above, our present efforts are concentrated on the computation of phase velocities at arbitrarily specified (equally spaced) frequencies. Thus to keep our algorithms as simple as possible, and to avoid the difficulties involved in the evaluation of spherical Bessel functions of nonintegral order, we have chosen to terminate our integrations at depth with boundary conditions appropriate to free or rigid surfaces. We have therefore included the details of our numerical evaluation of the efficiency of this departure from the usual manner of terminating the integrations: (1) ending the integration within the mantle by applying terminating boundary conditions for a gravitating, homogeneous, solid sphere below r_0; and (2) terminating at the mantle-core boundary by applying the conditions for a homogeneous, liquid sphere below r_0. The analytic details for these two cases, the respective dispersion functions, and the generalization of the new algorithms for direct computation of group velocities and partial derivatives, are contained in section 3 of the additional notes. Also included there is the complete description of our numerical experiments for testing the efficiency of our simplified boundary conditions at depth; the results are summarized in item 3 of the following "Conclusions."

Conclusions. An analysis of direct, Rayleigh-wave dispersion computations on a spherical, gravitating earth has been performed using the AJP (1959) formulation.

1. No difficulty was encountered when we reversed the usual procedure, for practical purposes, and computed phase velocities (or polar order numbers) at specified periods.

2. Integration from the free surface downward, again reversing the "standard" procedure, resulted in no unexpected difficulties. In fact, this procedure much simplified the specification of the algorithm for integrating the system of differential equations to obtain phase-velocity dispersion. This procedure also makes the generalization from the algorithm for continental, to oceanic structures relatively trivial; it makes it possible to develop direct algorithms for obtaining group velocities for the two types of structures, as well as partial derivatives with respect to the structural parameters; and it was an important aid in the development of the

improved, high-speed algorithm described in the fourth section. The simplification of computational algorithms is the important advantage provided by downward integration. Comparative numerical analysis of integration in the upward and downward directions leads us to infer no preference relative to overflow, loss-of-precision, or accuracy features.

3. The accuracy and efficiency of replacing the usual terminating boundary condition—a homogeneous sphere at depth—with boundary conditions that simplify the algorithm, e.g., a free or rigid boundary at depth, have been tested by numerical experiments. To obtain a given accuracy in the phase velocity, the boundary of the homogeneous sphere is situated above the free or rigid boundary. Thus computations with the usual condition require fewer integration steps. This advantage is significant, however, only for the lowest (radial) mode numbers. For computations directed toward the synthesis of complete theoretical seismograms, which are to be compared directly with those recorded on the long-period instruments of the WWSSN, this advantage is lost and the simplified terminating boundary conditions can be used without penalty. The use of these conditions also simplifies our direct algorithms for calculating group velocities and partial derivatives, the high-speed algorithm described in the fourth section, and especially the delta-matrix extension of this high-speed algorithm.

4. The overflow problem in the AJP formulation can be controlled by simple normalization. Program segments are given which describe the procedure explicitly. In general, for the relatively low radial order numbers treated in this section, overflow is encountered only in program testing. In routine computations, it is easily avoided by retaining only the minimum amount of structure needed to attain the desired accuracy in the calculated phase velocities. The *general* normalization scheme that is given here should be avoided in routine computations due to the additional expense that it imposes. For use with relatively high-radial orders numbers, a more restricted form of normalization, the cost of which is trivial, is described in the following section.

5. Our optimization of the AJP formulation is based on removing all function evaluations from the innermost, integration (over r) loops of the program. In fact, most of the evaluation procedure for $a_{ij}(r_k)$ can even be removed from the phase-velocity and frequency loops. The efficiency of an algorithm for dispersion computations can be specified by its characteristic time, i.e., the time required to complete one integration step when using a direct-integration procedure to form the dispersion function, or the time needed to treat one layer when employing the homogeneous-layer approximation. The applications we are mainly interested in involve the treatment of extremely large numbers of frequencies in a single computer run. For such applications, the effective characteristic times of our optimizations of the AJP formulation are 266×10^{-6} sec/integration step/iteration for a gravitating sphere, and 143×10^{-6} sec/step/iteration when gravity is not included. These times for the AJP, direct-integration procedure provide a basis of comparison with the homogeneous-layer approximation for a nongravitating, flat structure, which has a characteristic time of 110×10^{-6} sec/layer/iteration (optimized form of Knopoff's method, Schwab and Knopoff, 1972). All of the above times apply to the IBM 360/91 computer at the University of California at Los Angeles. These timing results are preliminary; our final, most complete timing details are given in the fourth section.

6. Our results here, combined with those of Schwab and Knopoff (1972), indicate that an integration "step" (in the AJP procedure, when using Runge-Kutta and predictor-corrector methods of the fourth order) can be considered roughly equiv-

alent to a "layer" in computations based on the homogeneous-layer approximation. Also, to the accuracy possible in this type of comparison, the "iterations" required in the two techniques (for details see Schwab and Knopoff, 1972) can be considered equivalent. Thus the relative efficiencies of the two types of Rayleigh-wave dispersion computations can be determined by simple comparison of the above characteristic times. For the applications in which we are mainly interested, these preliminary timing results show that the optimized form of the AJP formulation (spherical structures) is 2.42 times slower than Knopoff's method (flat, nongravitating structures) when gravity is included in the spherical model, and 1.30 times slower when gravity is not included.

The fact that approximately the same number and sizes of "steps" must be used in the direct-integration procedure, as number and sizes of "layers" in the homogeneous-layer approximation, indicates an apparent lack of validity of the usual assumption that the former method does a better job of treating continuous parameter-depth distributions. This conclusion is based on tests performed at the four-figure accuracy level (c or l), with which we are concerned; the equivalence tests of steps and layers have not been extended to higher accuracies.

7. A loss-of-precision problem appears to be intrinsic to the AJP formulation. Results of this problem: for a fixed period, as (radial) mode number increases, the attainable accuracy in phase velocity decreases, for a fixed accuracy in the phase velocity, as period decreases, the maximum mode number that can be treated successfully decreases. Even with the improved, high-speed algorithm developed below, it will be necessary to include a loss-of-precision control feature before attempting to compute complete synthetic seismograms that contain all of the energy down to a period of 10 sec. Referring to the results of the following section, this indicates that the most immediate new question, relative to our work in the present analysis, is whether orthogonalization or the delta-matrix method is preferable to control loss of precision.

3. Loss-of-Precision Problem: Short-Period Computations for High Radial Mode Numbers

In the description of this problem, four main points are of interest: the numerical aspects of the difficulty; the interpretive aspects; the connection between loss of precision and numerical instability; and the possible methods for overcoming this precision loss. Owing to limited space, our detailed discussions of each of these points are contained in the additional notes (section 4). Of the various possible methods for overcoming precision loss, in the present section we treat orthogonalization. To provide as complete a test as possible for this solution to precision loss, separate programs for propagating Rayleigh waves and for spheroidal, free-mode oscillations were employed. Therefore, in the additional notes (section 5) we have also given brief descriptions of each of these routines, along with full discussion of the results—other than those connected with orthogonalization—that are pertinent to computations at short periods, and high radial mode numbers. These results are also stated in the "Conclusions" at the end of the present section.

Orthogonalization. As is explained in the additional notes, after each integration step some degree of independence of the solution vectors, $\boldsymbol{X}_i$, is lost owing to the combination of a finite-precision machine and the dominance of exponential over oscillatory tendencies in the solutions. To avoid this tendency toward parallelism of the vectors, orthogonalization has been combined with the actual integration.

If we denote the vectors at the end of an integration step by $\boldsymbol{X}_i^{old}(r_j)$, and the vectors then obtained with Gram-Schmidt orthogonalization by $\boldsymbol{X}_i^{new}(r_j)$, then our computational algorithm is given by

$$
\begin{aligned}
X_1^{new} &= X_1^{old} \\
X_2^{new} &= X_2^{old} - \frac{(X_2^{old}, X_1^{new})}{(X_1^{new}, X_1^{new})} X_1^{new} \\
X_3^{new} &= X_3^{old} - \frac{(X_3^{old}, X_2^{new})}{(X_2^{new}, X_2^{new})} X_2^{new} - \frac{(X_3^{old}, X_1^{new})}{(X_1^{new}, X_1^{new})} X_1^{new},
\end{aligned} \tag{3.1}
$$

where the notation (D, E) is used to represent the inner product of the vectors $\boldsymbol{D}$ and $\boldsymbol{E}$. To express this in a more compact manner, we define the 6×3 matrix

$$X = [X_1 X_2 X_3], \tag{3.2}$$

each of whose columns is composed of the elements of one of the vectors $\boldsymbol{X}_i$. It is then easy to show that (3.1) can be written as

$$X^{old} = X^{new} T, \tag{3.3}$$

where this linear transformation is described by

$$T = \begin{bmatrix} 1 & t_{12} & t_{13} \\ 0 & 1 & t_{23} \\ 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

with

$$t_{12} = \frac{(X_2^{old}, X_1^{new})}{(X_1^{new}, X_1^{new})}, \quad t_{13} = \frac{(X_3^{old}, X_1^{new})}{(X_1^{new}, X_1^{new})}, \quad t_{23} = \frac{(X_3^{old}, X_2^{new})}{(X_2^{new}, X_2^{new})}. \tag{3.5}$$

Further details concerning Gram-Schmidt algorithms and their properties are given by Lawson and Hanson (1974).

It is important that the introduction of orthogonalization does not change the value of the dispersion function from that which would result, without orthogonalization, were precision loss absent. If for upward (downward) integration we denote increasing (decreasing) r by increasing values of its subscript, then each step of a numerical integration procedure such as a Runge-Kutta method can be written for our system as

$$X(r_{m+1}) = V(r_m)\, X(r_m), \tag{3.6}$$

where the 6×6 matrix $V(r_m)$ describes the integration from r_m to r_{m+1}. If we begin the integration at r_1 and perform α steps, X becomes

$$X(r_{1+\alpha}) = V(r_\alpha) V(r_{\alpha-1}) \cdots V(r_2)\; V(r_1) X(r_1); \tag{3.7}$$

and after an orthogonalization at $r_{1+\alpha}$,

$$X^{new}(r_{1+\alpha}) = V(r_\alpha) V(r_{\alpha-1}) \cdots V(r_2) V(r_1) X(r_1) T^{-1}(r_{1+\alpha}), \tag{3.8}$$

where α vanishes if an orthogonalization is applied before integration is initiated. To continue the demonstration, we proceed with another β integration steps followed by an orthogonalization, γ further steps and an orthogonalization, $\cdots$, ψ steps plus orthogonalization, and a final ω integration steps, from which we have

$$X^{\text{orth}}(r_{1+\alpha+\beta+\cdots+\psi+\omega}) = V(r_{\alpha+\beta+\cdots+\psi+\omega})\cdots V(r_2)V(r_1)X(r_1) \times T^{-1}(r_{1+\alpha})T^{-1}(r_{1+\alpha+\beta})\cdots T^{-1}(r_{1+\alpha+\beta+\cdots+\psi}), \quad (3.9)$$

or

$$X^{\text{orth}}(r_{1+\alpha+\beta+\cdots+\psi+\omega}) = X(r_{1+\alpha+\beta+\cdots+\psi+\omega}) \times T^{-1}(r_{1+\alpha})T^{-1}(r_{1+\alpha+\beta})\cdots T^{-1}(r_{1+\alpha+\beta+\cdots+\psi}). \quad (3.10)$$

We can then define a matrix U, composed of the first, third, and fifth rows of X, which of course must also satisfy (3.10). Taking the determinant of both sides of the resulting relation, we have

$$\det[U^{\text{orth}}(r_{\text{end}})] = \det[U(r_{\text{end}})], \quad (3.11)$$

where $r_{1+\alpha+\beta+\cdots+\psi+\omega}$ is taken to be the final point of integration when forming the dispersion function, i.e., r_{end}. The right-hand side of (3.11) is exactly the dispersion function defined by equation (2.10), in terms of the vector components *without* orthogonalization; therefore, (3.11) demonstrates the invariance of the dispersion function under orthogonalization, when a rigid boundary is used to terminate the integration at depth. Simple variations of this demonstration, by redefining the matrix U, establish the invariance of other forms of the dispersion function for either upward or downward integration. The importance of this invariance is that it ensures a smooth, regular variation of the dispersion function as roots are sought by altering the value of the dispersion variable.

Another important point concerning the introduction of orthogonalization into our integration process is the question of the modifications then required to compute the actual components of motion, i.e., the components of $Y(r)$. These modifications are straightforward, and merely require the reversal of the above procedure. In the following description, details concerning normalization (see section 5 of the additional notes) will be neglected.

Assume that the last iteration has been performed to obtain the value of the dispersion variable at the root of the dispersion function. After each integration step the values of $[X_i]_j$ have been stored; at the depths where orthogonalization has been applied, $[X_i^{old}]_j$ are the components stored, along with t_{12}, t_{13}, and t_{23}. Since each orthogonalization is just a linear transformation, Y at the starting point of integration

$$Y = e_1X_1 + f_1X_2 + h_1X_3 \quad (3.12)$$

is only transformed into

$$Y = e_\psi X_1^{\text{orth}} + f_\psi X_2^{\text{orth}} + h_\psi X_3^{\text{orth}} \quad (3.13)$$

at the terminating point of the integration. The constants e_ψ, f_ψ, h_ψ are obtained by

application of the boundary conditions at the last point of integration, and with these constants, relation (3.13) yields the components of motion for the last ω steps of the integration, i.e., at

$$r_{1+\alpha+\beta+\ldots+\psi+1}$$

$$r_{1+\alpha+\beta+\ldots+\psi+2}$$

$$\vdots$$

$$r_{1+\alpha+\beta+\ldots+\psi+\omega}.$$

The effect of the last orthogonalization is removed by applying

$$X^{new} = X^{old}T^{-1} = X^{old}\begin{bmatrix} 1 & -t_{12} & t_{12}t_{23} - t_{13} \\ 0 & 1 & -t_{23} \\ 0 & 0 & 1 \end{bmatrix} \tag{3.14}$$

to (3.13) at $r_{1+\alpha+\beta+\ldots+\psi}$, which yields

$$Y = e_\chi X_1^{\text{orth}} + f_\chi X_2^{\text{orth}} + h_\chi X_3^{\text{orth}}, \tag{3.15}$$

where

$$\begin{aligned} e_\chi &= e_\psi - t_{12}f_\psi + (t_{12}t_{23} - t_{13})h_\psi \\ f_\chi &= f_\psi - t_{23}h_\psi \\ h_\chi &= h_\psi. \end{aligned} \tag{3.16}$$

Relation (3.15) can then be used to determine the components of Y at the ψ points

$$r_{1+\alpha+\beta+\ldots+\chi+1}$$

$$r_{1+\alpha+\beta+\ldots+\chi+2}$$

$$\vdots$$

$$r_{1+\alpha+\beta+\ldots+\chi+\psi}.$$

The same procedure is repeated up to the point of the first orthogonalization, $r_{1+\alpha}$, where (3.12) applies, and e_1, f_1, h_1 are obtained from e_α, f_α, h_α, and (3.16). Relation (3.12) is then employed to specify the components of motion at the $1 + \alpha$ points r_1, r_2, $\ldots$, $r_{1+\alpha}$. Of course, e_1, f_1, h_1 contain a common, multiplicative factor, which means that no generality is lost by applying the form of normalization described in the "General Results" of section 5 in the additional notes. In our initial numerical tests of the above algorithm we compared the calculated components of $Y(r)$, from programs both with and without orthogonalization, for cases in which loss of precision is not dominant. These tests were satisfied by obtaining the same results

from both programs for the components of motion. The tests with cases for which precision loss is completely dominant, in the original AJP formulation, produced well behaved $y_i(r)$ as soon as orthogonalization was introduced into the computations.

The numerical tests in the present section were again performed on an IBM 360/91 computer, using a precision equivalent to 16 to 17 decimal digits. The first of our preliminary tests showed that even when orthogonalization is applied after *every* integration step, the cost of adding this operation is quite reasonable. In routine computations, the general requirement for control of precision loss appears to be an orthogonalization about every third step, which amounts to an increase in cost of roughly 5 to 10 per cent over that for the original algorithms. The second preliminary check concerned the invariance of the dispersion function relative to orthogonalization. Successful tests of this feature involved: (1) duplicate runs of the routines with and without orthogonalization, for cases in which precision loss is not significant; and (2) duplicate runs of the routines containing this process, in which orthogonalization was applied different numbers of times during the integration over depth. The latter runs were performed with cases dominated by precision loss.

In the "Connection Between Loss of Precision and Instability" (section 4 of the additional notes) we mention that numerical instability cannot be eliminated, in an *absolute* sense, by any linear transformation applied to our three vectors. Thus it is a matter of some importance to us to determine how well our linear transformation—orthogonalization—suppresses this inherent instability, in a *practical* sense. Of course the critical aspect of this suppression is the degree of success we have in eliminating loss of precision, which has its quantitative expression in how well orthogonalization succeeds in transforming the situation in Figure 16a, to that in Figure 16b. More specifically, for each radial node n, treated without orthogonalization in Figure 16a, acceptable (r_0, σ) pairs must fall below the upper portion of the curve—structural limitation—and above the lower portion—limitation due to precision loss. The quantitative, practical success of orthogonalization in eliminating precision loss is therefore measured by its ability to remove this lower limitation. Pleasantly, if somewhat surprisingly, for all periods and radial mode numbers that were tested with orthogonalization included in the algorithms, the loss-of-precision problem appeared to be *completely* under control. For a given value of n, once the limit of integration, r_0, fell below the solid line in Figure 16a, σ, the number of stabilized figures in the computed dispersion variable did not decrease no matter how much further r_0 was reduced; the practical limitation on the reduction of r_0 is the point at which our restricted form of normalization ceases to control overflow, but this is always several hundred kilometers below the structural limitation depicted by the solid lines in Figure 16b. Our most detailed stability tests were performed at a period of 10 sec, with radial modes 4 and 15 of a continental structure. Once having passed any given (r_0, σ) point on the structural-limitation curve, the stability of the computed dispersion variable did not decrease as r_0 was decreased by at least 1500 km.

The dispersion function, computed with and without orthogonalization, is illustrated in Figure 17 for two realistic structures and a range of r_0 values. This figure represents the fourth mode at a period of 10 sec; a case in which precision loss is completely dominant in the original AJP formulation. With orthogonalization included in the computations, we find the dispersion function to be very smooth and well behaved as a function of the dispersion variable. An interesting, and apparently quite general aspect of dispersion computations, is illustrated very strikingly in this

figure. We first note that the vertical scale must be compressed more and more as an increasing amount of structure—decreasing r_0—is included in the computations. As this happens, the dispersion function will be seen to become more and more "L shaped." In our past experience we have noted that this is also a feature of dispersion computations with flat structures composed of a sequence of homogeneous layers. Knowledge of this property of the dispersion function can be of importance in practical computations, especially as an aid in selecting an efficient scheme for refining a root once it has been bracketed. Obviously, from the curves for $r_0 = 4500$ km, point-slope, linear interpolation based on a point at the right side of the "L" would be a very poor choice for root refinement. A two-point (with opposite signs) scheme of linear interpolation would also be a poor choice for root refinement since it would converge very slowly toward the root from the right. A detailed, numerical example of this last case is given by Schwab and Knopoff (1970, Table 2). Since we also find the "L-shaped" form (reversed in this case) of the dispersion function in the vicinity of each root in Figure 18, a common cause for this feature is suggested: a multiplicative, exponential dependence of the dispersion function on the parameter that is being varied.

As final confirmation—based on absolute accuracy of the computed dispersion variable—that orthogonalization solves the loss-of-precision problem represented by the lower portions of the curves in Figures 12 and 13, two tests were performed: (1) at a period of about 50 sec the phase velocity for radial mode 13 was computed with the spheroidal-mode program, and was then checked successfully to four-figure accuracy by comparing with the result calculated from the Rayleigh-wave routine; (2) at a period of about 25 sec, this same test was performed successfully with radial mode 24. Thus there is no question that orthogonalization solves the numerical problem to the extent that it was possible to define it in the second section, for realistic oceanic and continental models of the earth.

Still dealing with the realistic models of the earth that are given in Figure 5, the tests of our solution for precision loss were extended down to the minimum period of interest: 10 sec. With the spheroidal-mode program the phase velocities were computed for increasing radial mode numbers, at periods near 10 sec. At the period corresponding to each of these spheroidal modes, the Rayleigh-wave program was then used to obtain a successful check of the phase velocity to four significant figures. This test was performed successfully up to $n = 30$, at which point the grouping of dispersion results indicates an interesting new problem in seismic interpretation for realistic models of the earth's radial heterogeneity. Since this problem is unrelated to the present study, it will be treated elsewhere. As mentioned previously, to obtain the desired theoretical seismograms for Rayleigh waves, the results of Liao *et al.* (1978) indicate that we will need a solution to precision loss for radial mode numbers up to 90 to 100, at a period of 10 sec. To obtain unquestionable verification of the accuracy of our new algorithms we decided to check the results from these two routines, to this minimum period and maximum n, against a set of dispersion values that could be obtained independently, to an arbitrary absolute accuracy. For this purpose we replaced our realistic models of the earth with a gravitating, homogeneous sphere of radius 6371 km, density 5.52 gm/cm^3, Lamé constants given by

$$\lambda/\mu = 2.402$$

$$\mu = 1.463 \times 10^{12} \text{ gm/cm sec}^2,$$

and a gravitational constant G of 6.67×10^{-8} cm^3/gm sec^2. The reference dispersion

results are given in Table 3. With orthogonalization included in our algorithms, the dispersion values from the spheroidal-mode program agree with the reference results over the entire range of n values, with the arbitrary accuracy of six significant figures selected in our free-mode program. As n increases from zero, the agreement of the results from our Rayleigh-wave routine decreases to somewhat better than four significant figures by the time $n = 5$ is reached, when the maximum step size of our integration is held at 12.5 km; with the maximum step size decreased to 6.25 km at $n = 6$, the accuracy of the computed results does not decrease again to four figures until n reaches 26; from 27 on, the maximum step size must again be decreased to maintain the desired accuracy.

In the above descriptions of computations with realistic models of the earth, we have concentrated on cases in which the modal energy does not penetrate significantly into the core. To complete the present study we focus on the dispersion of long-period spheroidal modes. The practical consideration here is the ability to perform successful dispersion computations for *all* of those modes that make up the present data set used in inversion studies (Gilbert and Dziewonski, 1975). Although most of the modes in this data set can be handled with the original AJP formulation, precision loss is encountered with some when using the 16 to 17 decimal digits employed by our IBM 360/91 computer. That is, even for relatively small values of l, loss-of-precision problems can eventually become significant as n increases and the modes strongly sample the core of the earth. The cases of interest here are those with l less than about 30, and with periods of the order of, or less than 80 sec. For successful handling of these cases, e.g., ${}_{25}S_{18}$, ${}_{27}S_{16}$, ${}_{15}S_{24}$, ${}_{17}S_{12}$, ${}_{30}S_{10}$, ${}_{34}S_{1}$, it was found completely satisfactory to introduce orthogonalization in the mantle and outer section of the core. For these types of modes, no need was found to orthogonalize the vectors within the inner section of the core; however, this might be necessary for periods of 50 sec or less.

Conclusions. When computing the dispersion of Rayleigh waves or spheroidal modes for a spherical model of the earth, algorithms based on the original AJP formulation have an intrinsic problem with precision loss which becomes increasingly severe as period decreases and/or radial mode number increases. This report presents, we believe for the first time with direct methods, a demonstrated solution to this problem for *gravitating* structures.

The basic task of these computations is the numerical integration, over depth, of three independent vectors. The angles between these vectors were computed as a function of depth. The results show a persistent tendency of the "independent" vectors to become parallel. This spurious behavior renders the dispersion function meaningless, causing the calculations to lose their significance. Since the vector integrations are governed by equation (2.1), the numerical properties of the calculations can be inferred from those of the matrix A. If we note that over its interior the properties of the earth vary rather smoothly, this allows us to make a simplified eigenvalue analysis of this matrix, which provides a very simple interpretation of the parallelism of our vectors. Over a range of r, the eigenvalues yield vector solutions containing both exponential and oscillatory terms; the combination of a finite-precision computer and the dominance of exponential over oscillatory tendencies in the solutions leads, eventually, to only this dominant tendency being left in the solutions; hence, the spurious parallelism. The dispersion function is a determinant—roots of which yield the desired dispersion results—each of whose columns is composed of elements from one of the solution vectors, which demonstrates the equivalence of this parallelism and the equally artificial roots that define the loss-of-precision problem.

The detailed analysis of the eigenvalues of A shows that integration in *both* upward and downward directions exhibits the above problem; there is no preferred direction of integration relative to the loss-of-precision problem. The numerical tests of the second and third sections of this report fully support this conclusion. Another set of tests (section 5 of the additional notes) was carried out to determine whether there is a preferred direction of integration relative to overflow problems. The results show a striking agreement between the overflow features for upward and downward integration; these features are precisely the same in both form and magnitude.

Like precision loss, the overflow problem is also a result of the inherent instability (see section 4 of the additional notes) of the original AJP formulation. When dealing with the large radial order numbers that the solution to precision loss permits us to treat, even routine computations encounter overflow. However, the problem is not so severe that its control necessitates as powerful and expensive a normalization procedure as that described in the second section. A restricted form of normalization, the cost of which is trivial, has been found sufficient for radial mode numbers up to 90 to 100, and for periods down to 10 sec. This method, the full details of which are given in section 5 of the additional notes, requires normalization only at the first and last points of integration. It also yields a dispersion function that has, as well as a meaningful sign, magnitudes that vary smoothly and regularly from iteration to iteration, i.e., as a function of the dispersion variable.

To avoid the tendency toward parallelism of the solution vectors, and hence the problem of precision loss, we have combined Gram-Schmidt orthogonalization with the actual integration. To provide as complete a set of tests as possible for this procedure, separate programs for propagating Rayleigh waves and for free-mode oscillations were employed. By so doing, orthogonalization has been shown effective over a broad range of physical and numerical techniques. We obtained five basic results from the analysis of orthogonalization.

1. The dispersion function is invariant under orthogonalization. This is demonstrated formally, and checked numerically. The importance of this invariance is that it ensures a smooth, regular variation of the dispersion function as roots are sought by altering the value of the dispersion variable. To be sure of this smoothness, overflow is controlled with the previously mentioned form of restricted normalization, rather than by introducing orthonormalization at each point where we orthogonalize the vectors.

2. The introduction of orthogonalization into the integration process leads to modifications in the means of obtaining the actual components of motion from the vector solutions. These modifications are shown to be quite straightforward. Specific details are given in (3.12) to (3.16).

3. In routine computations, we have found it sufficient to orthogonalize about every third integration step. This amounts to an increase in cost of only about 5 to 10 per cent over that of the original algorithms.

4. Our numerical tests for all periods down to 10 sec, and for all radial mode numbers up to 90 to 100, indicate that orthogonalization brings the loss-of-precision problem under *complete* control. These tests employed: (1) realistic models of continental and oceanic structures, for which absolute accuracy of the computed dispersion variable was checked by comparing the results from our two independent programs; and (2) a gravitating, homogeneous sphere. The latter case permitted us to obtain unquestionable verification of the accuracy of our new algorithms, since reference dispersion results can be computed independently, to arbitrary absolute accuracy.

5. The first group of our orthogonalization tests were focused on cases in which the seismic energy is confined to the mantle of the earth. In the second group of these tests we concentrated on modes with significant energy within the core. In the latter case, we again found orthogonalization to be a completely satisfactory solution to precision loss.

The combination of orthogonalization to control precision loss, and the improved algorithm for high-speed computations that is described in the following section, now makes it possible to consider the routine calculation of complete, spheroidal-wave seismograms. These are designed for direct comparison with records from the long-period instruments of the WWSSN, where the theoretical records contain *all* of the seismic energy generated at periods above 10 sec. By comparing the results of our present tests at periods of 50, 25, and 10 sec, with those from the comparable program for complete, torsional-wave seismograms, we find that the latter routine can be used to estimate the maximum required values of the radial mode numbers. Approximately 6 to 7 significant figures in the stabilized dispersion values are desirable—this is *not* absolute accuracy (see "Numerical Testing" in section 6 of the additional notes for an explanation of this point)—which indicates that at 50, 25, and 10 sec we can go up to radial mode numbers of 10, 23, and 62, respectively, without entering the core. Thus these numbers are representative requirements for the spheroidal-wave program if it is to generate only those seismic arrivals that do not quite penetrate to the mantle-core boundary. If reflections from this boundary, and phases that penetrate somewhat deeper are also desired, then the approximate numbers of radial modes that will be required at the above periods are 18, 37, and 93, respectively.

With orthogonalization included, short-period computations were possible here with much higher radial mode numbers, n, than could be treated in the tests described in the second and fourth sections. It was therefore possible to examine the maximum integration step size required—for an absolute accuracy of four significant figures in the computed phase velocities—in the calculation of complete theoretical seismograms down to a period of 10 sec: (1) a maximum step size of 12.5 km can be used for all periods down to 25 sec. (2) Somewhere between 25 and 10 sec, it will be necessary to begin decreasing this maximum step size; the value of n at which this decrease is to be effected, decreases with decreasing period. (3) At 10 sec, this maximum must be decreased at about $n = 6$—the precise value of n probably being dependent upon the specified structure; if this decrease is to 6.25 km, this will suffice until n reaches about 27; one additional decrease should be sufficient to complete the computations at this period.

4. Improved Algorithm for High-Speed Computations

Simplification of equations of motion from AJP formulation. Up to the present time, the sixth-order formulation (second section) has been the basis of direct computations with gravitating structures. The simplification to a fourth-order system involves the reduction of

$$y_5(r) = \frac{4\pi G}{2l+1}\left\{\frac{1}{r^{l+1}}\int_o^r \rho(\zeta)[l y_1(\zeta) + l(l+1)\,y_3(\zeta)]\zeta^{l+1}\,d\zeta + r^l\int_r^a \rho(\zeta)[-(l+1)\,y_1(\zeta) + l(l+1)\,y_3(\zeta)]\zeta^{-l}\,d\zeta\right\} \quad (4.1)$$

[equation (35), Hoskins, 1920; section 5, Pekeris and Jarosch, 1958] and

$$y_6(r) = \frac{4\pi G}{2l+1}\left\{-\frac{l+1}{r^{l+2}}\int_o^r \rho(\zeta)[l y_1(\zeta) + l(l+1)\,y_3(\zeta)]\zeta^{l+1}\,d\zeta \right.$$

$$\left. + l r^{l-1}\int_r^a \rho(\zeta)[-(l+1)\,y_1(\zeta) + l(l+1)\,y_3(\zeta)]\zeta^{-l}\,d\zeta\right\} \quad (4.2)$$

to approximations

$$y_5(r) = c_5 y_1(r) + d_5 y_3(r) \quad (4.3)$$

$$y_6(r) = c_6 y_1(r) + d_6 y_3(r), \quad (4.4)$$

which are then used to effect the desired simplification from sixth-order form [equation (2.1)]. The most difficult part of the present work was to determine the best technique for forming approximations (4.3) and (4.4). Thus this will be treated in some detail.

As a preliminary question, one might ask whether highly accurate values of y_5 and y_6 are actually required for dispersion computations. If it is noted that these quantities are directly proportional to the gravitational constant G, while y_1, y_2, y_3, and y_4 do not have this property, this provides an easy means for obtaining the rough estimate needed to answer this question. Since

$$\delta G/G \approx \delta y_5/y_5 \approx \delta y_6/y_6, \quad (4.5)$$

an estimate of the effect of errors in y_5 and y_6 can be obtained by varying G and noting the effect on phase velocity when executing the program described in the second section. The results of this test are shown in Table 4. They indicate quite clearly that relatively large errors in y_5 and y_6 have small effects on the accuracy of computed phase velocities. Since this is intended only as a preliminary test, the illustration is limited to one period and eight (radial) modes.

From these results it is apparent that approximations (4.3) and (4.4) need not be highly accurate; fairly good graphical agreement between the approximations and the true values of y_5 and y_6 should ensure satisfactory accuracy in phase velocities computed with algorithms based on (4.3) and (4.4). Thus the results of various approaches to constructing these expressions will be presented in graphical form. The test computations are again based on the sixth-order algorithm. At the correct value of c,

$$\begin{bmatrix} y_3(a)/y_1(a) \\ y_5(a)/y_1(a) \end{bmatrix} = \frac{1}{[X_2(r_0)]_1[X_3(r_0)]_3 - [X_2(r_0)]_3[X_3(r_0)]_1} \times \begin{bmatrix} -[X_1(r_0)]_1\ [X_3(r_0)]_3 + [X_1(r_0)]_3\ [X_3(r_0)]_1 \\ [X_1(r_0)]_1\ [X_2(r_0)]_3 - [X_1(r_0)]_3\ [X_2(r_0)]_1 \end{bmatrix} \quad (4.6)$$

for rigid-boundary termination. Integration is performed with starting vector $\boldsymbol{X_S(a)}$

$$\begin{bmatrix} y_1(a) \\ y_2(a) \\ y_3(a) \\ y_4(a) \\ y_5(a) \\ y_6(a) \end{bmatrix} = y_1(a) \begin{bmatrix} 1 \\ 0 \\ y_3(a)/y_1(a) \\ 0 \\ y_5(a)/y_1(a) \\ -\dfrac{l+1}{a}\dfrac{y_5(a)}{y_1(a)} \end{bmatrix} = y_1(a) X_S(a) \quad (4.7)$$

to determine $y_i(r)/y_1(a)$ and $\bar{y}_i(r)/y_1(a)$, from which any desired approximations (4.3) and (4.4) can be evaluated and then compared with the directly computed dependences $y_5(r)/y_1(a)$ and $y_6(r)/y_1(a)$. Relations (4.6) and (4.7), of course, apply to a continental structure. For an oceanic model, the left-hand side of (4.6) becomes

$$\begin{bmatrix} y_3(r_1)/y_1(a) \\ \\ y_5(a)/y_1(a) \end{bmatrix}, \tag{4.6a}$$

and integration is performed with starting vector $\boldsymbol{Z_L(a)}$

$$\begin{bmatrix} y_1(a) \\ y_2(a) \\ y_5(a) \\ y_6(a) \end{bmatrix} = y_1(a) \begin{bmatrix} 1 \\ 0 \\ y_5(a)/y_1(a) \\ -\dfrac{l+1}{a}\dfrac{y_5(a)}{y_1(a)} \end{bmatrix} = y_1(a)Z_L(a); \tag{4.7a}$$

at the bottom of the liquid oceanic layer,

$$\begin{bmatrix} y_1(r_1) \\ y_2(r_1) \\ y_5(r_1) \\ y_6(r_1) \end{bmatrix} = y_1(a) \begin{bmatrix} [Z_L(r_1)]_1 \\ [Z_L(r_1)]_2 \\ [Z_L(r_1)]_5 \\ [Z_L(r_1)]_6 \end{bmatrix}; \tag{4.7b}$$

and integration is continued below this point with starting vector $\boldsymbol{X_S(r_1)}$

$$\begin{bmatrix} y_1(r_1) \\ y_2(r_1) \\ y_3(r_1) \\ y_4(r_1) \\ y_5(r_1) \\ y_6(r_1) \end{bmatrix} = y_1(a) \begin{bmatrix} [Z_L(r_1)]_1 \\ [Z_L(r_1)]_2 \\ y_3(r_1)/y_1(a) \\ 0 \\ [Z_L(r_1)]_5 \\ [Z_L(r_1)]_6 \end{bmatrix} = y_1(a)X_S(r_1). \tag{4.7c}$$

Approximation 1: if, for example, we write (4.1) in the form

$$\begin{aligned} y_5(r) = \frac{4\pi G}{2l+1}\Bigg\{ & l\int_o^r \rho(\zeta)y_1(\zeta)\left(\frac{\zeta}{r}\right)^{l+1} d\zeta \\ & + l(l+1)\int_o^r \rho(\zeta)y_3(\zeta)\left(\frac{\zeta}{r}\right)^{l+1} d\zeta \\ & - (l+1)\int_r^a \rho(\zeta)y_1(\zeta)\left(\frac{\zeta}{r}\right)^{-l} d\zeta \\ & + l(l+1)\int_r^a \rho(\zeta)y_3(\zeta)\left(\frac{\zeta}{r}\right)^{-l} d\zeta\Bigg\}, \end{aligned} \tag{4.8}$$

it is clear that the desired approximations require $y_1(\zeta)$ and $y_3(\zeta)$ to be removed from the integrals in the forms $y_1(r)$ and $y_3(r)$, and the remaining integrals to be

evaluated. Since the contributions of the integrands are concentrated toward the points $\zeta = r$, the obvious approximation to try first is that arising from Taylor's series expansions of $\rho(\zeta)\, y_i(\zeta)$ and $\rho(\zeta)\, y_3(\zeta)$ about $\zeta = r$

$$\rho(\zeta)\, y_i(\zeta) \approx \rho(r)\, y_i(r) + [\bar{\rho}(r)\, y_i(r) + \rho(r)\, \bar{y}_i(r)](\zeta - r) + \ldots . \tag{4.9}$$

The test results from approximation 1 are shown in Figure 20. When only one term of the Taylor's series expansions is used, the results are unacceptable; if two terms are employed, the results improve except for the large spikes that are introduced in the second term by abrupt variations in $\bar{\rho}(r)$; the results of the three-term expansions are much worse, owing to the additional spikes introduced by $\bar{\lambda}(r)$ and $\bar{\mu}(r)$ which appear in the third terms. Because of these abrupt variations with depth, approximation 1 was abandoned.

Approximation 2: to improve on the first approximation, a bit of generality is sacrificed by leaving $\rho(\zeta)$ under the integral signs in (4.8) and assigning it an explicit specification—a sequence of linear functions of ζ—to permit the evaluation of the integrals. Thus the Taylor's series are simplified to

$$y_i(\zeta) \approx y_i(r) + \bar{y}_i(r)(\zeta - r). \tag{4.10}$$

The results of the second approximation are also given in Figure 20, where we see that although the first-term expression is unacceptable, the expression with two terms no longer has the large spikes of approximation 1.

Although the smoothness of the second approximation shows improvement, the absolute agreement with the true result is not too good. Thus some iterative procedure is indicated for improving this agreement. Once the initial approximations to $y_1(r)$ and $y_3(r)$ have been obtained—by running the algorithm based on (4.10) and then evaluating the eigenfunctions $y_i(r)$—an iterative technique can be set up to improve the agreement. Rewrite (4.6) as

$$\begin{aligned} y_5(r) = \frac{4\pi G}{2l+1} \{ & l I_1(r)\, y_1(r) + l(l+1) I_3(r)\, y_3(r) \\ & -(l+1) I_2(r)\, y_1(r) + l(l+1) I_4(r)\, y_3(r) \} \end{aligned} \tag{4.11}$$

with the approximations

$$\begin{aligned} I_1(r) &\approx \frac{1}{p y_1(r)} \int_0^r \rho(\zeta)\, p y_1(\zeta) \left(\frac{\zeta}{r}\right)^{l+1} d\zeta \\ I_2(r) &\approx \frac{1}{p y_1(r)} \int_r^a \rho(\zeta)\, p y_1(\zeta) \left(\frac{\zeta}{r}\right)^{-l} d\zeta \\ I_3(r) &\approx \frac{1}{p y_3(r)} \int_o^r \rho(\zeta)\, p y_3(\zeta) \left(\frac{\zeta}{r}\right)^{l+1} d\zeta \\ I_4(r) &\approx \frac{1}{p y_3(r)} \int_r^a \rho(\zeta)\, p y_3(\zeta) \left(\frac{\zeta}{r}\right)^{-l} d\zeta \end{aligned} \tag{4.12}$$

where py_i are the approximate eigenfunctions predicted from the algorithm based on (4.10)—or from successive improvements—and expressions (4.12) arise from the assumption that in the neighborhood of $\zeta = r$

$$y_i(\zeta) \approx \text{constant} \times py_i(\zeta); \tag{4.13}$$

since the contributions of the integrands are concentrated toward $\zeta = r$, the constants are evaluated at that point

$$y_i(\zeta) \approx \frac{y_i(r)}{py_i(r)} py_i(\zeta). \tag{4.14}$$

The proposed iterative procedure for improving approximation 2 is given in Figure 21. With the program described in the second section, this scheme has been tested; the results are shown in Figure 22, where it is seen that the procedure is convergent. Thus these tests indicate that the outline in Figure 21 is valid, and this approach is discussed in detail in section 8 of the additional notes. However, this is not the most efficient way of employing the fourth-order simplification of the AJP formulation. Reference to Figure 21 shows that the key stage in the procedure—step 10—requires evaluation of $py_i(r)$, and then $I_i(r)$, each time the dispersion function is computed. A much more efficient algorithm would be one in which the recalculation of $py_i(r)$ and $I_i(r)$ were not required with every evaluation of the dispersion function; this is the type of algorithm we seek.

A high-speed procedure of this kind will be found possible when many, equally spaced frequencies are treated. Here it is possible to approach computation at each new frequency with accurate predictions $py_i(r)$ and l, and therefore to make use of the procedure given in Figure 23. There it will be seen that $py_i(r)$ and $I_i(r)$ need be computed but once for each frequency. The question of interest then, is how best to extrapolate, from data at previous frequencies, to accurate predictions $py_i(r)$ at the new ω.

Approximation 3: the results of the first attempt to predict accurate eigenfunctions from data at preceding frequencies are shown in Figure 24. In this test, (4.14) and the eigenfunctions at the (single) preceding frequency—0.0005 cps below each of those shown—are used to predict the desired depth dependences for use in (4.12). The obvious problems reflect the fact that py_3 and y_3 have cross-over points at depths which differ significantly.

Approximation 4: Figure 25 illustrates the results when two preceding frequencies, and various extrapolation methods, are combined with (4.14). These results are much improved over those of Figure 24, but the fit at the longer period can still use some improvement.

Approximation 5: with the predictions py_i based on three preceding frequencies spaced 0.0005 cps apart, the test results in Figure 26 appear to justify full development of the algorithm outlined in Figure 23. After the first few (radial) modes, say 0 to 2, there should be enough features of $y_1(r)$ and $y_3(r)$—depths and magnitudes of extrema, depths of zeros, and depth of deepest point to be retained—to permit extrapolation based on these features alone. The first few modes, however, require an additional framework upon which to base the extrapolation procedure. For mode 0, at each depth this framework is the empirically determined, three-point extrapolation formulas.

$$py_1(\omega) \approx y_1(\omega - 3\Delta\omega)[y_1(\omega - \Delta\omega)/y_1(\omega - 2\Delta\omega)]^3, \tag{4.15}$$

and a similar expression for $py_3(\omega)$ at depths below the deepest extremum; above that last peak in displacement,

$$py_3(\omega) \approx x\, y_3(\omega - 3\Delta\omega) - (1 + 2x)\, y_3(\omega - 2\Delta\omega) + (2 + x)\, y_3(\omega - \Delta\omega), \tag{4.16}$$

where

$$x = \{[\Lambda(\omega) - \Lambda(\omega - 2\Delta\omega)]/[\Lambda(\omega - \Delta\omega) - \Lambda(\omega - 3\Delta\omega)]\}^{1.33}. \tag{4.17}$$

Λ denotes wavelength, and $\Delta\omega$ is the constant interval between frequencies. These formulas are based on a wavelength-scaled forward difference formula, and "forward ratio" generalizations of the usual forward difference formula. For modes 1 and 2, both $py_1(\omega)$ and $py_3(\omega)$ are obtained as $py_3(\omega)$ is for mode 0.

We now derive simplified equations of motion based on the use of approximation 5. In this case the reduction is quite easy. The integrals $I_i(r)$ are approximated by (4.12) and y_5 and y_6 are given by

$$\begin{aligned} y_5(r) = &\left\{\frac{4\pi G}{2l+1}[lI_1(r) - (l+1)I_2(r)]\right\} y_1(r) \\ &+ \left\{\frac{4\pi G}{2l+1}\, l(l+1)[I_3(r) + I_4(r)]\right\} y_3(r) \end{aligned} \tag{4.18}$$

$$\begin{aligned} y_6(r) = &\left\{\frac{4\pi G}{2l+1}\frac{l(l+1)}{r}[-I_1(r) - I_2(r)]\right\} y_1(r) \\ &+ \left\{\frac{4\pi G}{2l+1}\frac{l(l+1)}{r}[-(l+1)I_3(r) + lI_4(r)]\right\} y_3(r). \end{aligned} \tag{4.19}$$

Substitution into the sixth-order equations of motion [equation (2.1)] yields the fourth-order system

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \dot{y}_3 \\ \dot{y}_4 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ -A_{21} & a_{22} & A_{23} & a_{24} \\ a_{33} & 0 & a_{33} & a_{34} \\ A_{41} & a_{42} & A_{43} & a_{44} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}, \tag{4.20}$$

where the only modified elements of the coefficient matrix are

$$\begin{aligned} A_{21} &= a_{21} + \frac{4\pi G\rho}{2l+1}\frac{l(l+1)}{r}[I_1(r) + I_2(r)] \\ A_{23} &= a_{23} - \frac{4\pi G\rho}{2l+1}\frac{l(l+1)}{r}[-(l+1)I_3(r) + lI_4(r)] \\ A_{41} &= a_{41} - \frac{4\pi G\rho}{2l+1}\frac{1}{r}[lI_1(r) - (l+1)I_2(r)] \\ A_{43} &= a_{43} - \frac{4\pi G\rho}{2l+1}\frac{l(l+1)}{r}[I_3(r) + I_4(r)]. \end{aligned} \tag{4.21}$$

The preceding development, of course, applies to the solid portions of the earth. The application of the above method to an ocean modeled by a single, homogeneous, liquid layer, reduces the original fourth-order system [equation (2.16)] to one of the second order

$$\begin{bmatrix} \bar{y}_1 \\ \bar{y}_2 \end{bmatrix} = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}, \tag{4.22}$$

where

$$\begin{aligned} B_{11} &= b_{11} + b_{15} g \frac{\mathscr{A} + \mathscr{B}}{1 + \mathscr{B}} \\ B_{12} &= b_{12} - b_{15} \frac{1}{\rho} \frac{\mathscr{B}}{1 + \mathscr{B}} \\ B_{21} &= b_{21} + b_{25} g \frac{\mathscr{A} + \mathscr{B}}{1 + \mathscr{B}} - \rho \left[\mathscr{C} + g \mathscr{D} \left(1 - \frac{\mathscr{A} + \mathscr{B}}{1 + \mathscr{B}} \right) \right] \\ B_{22} &= b_{22} - b_{25} \frac{1}{\rho} \frac{\mathscr{B}}{1 + \mathscr{B}} + \mathscr{D} \left(1 - \frac{\mathscr{B}}{1 + \mathscr{B}} \right) \end{aligned} \tag{4.23}$$

with

$$\begin{aligned} \mathscr{A} &= \frac{4\pi G}{2l+1} \frac{1}{g} [l I_1(r) - (l+1) I_2(r)] \\ \mathscr{B} &= \frac{4\pi G}{2l+1} \frac{l(l+1)}{r\omega^2} [I_3(r) + I_4(r)] \\ \mathscr{C} &= \frac{4\pi G}{2l+1} \frac{l(l+1)}{r} [-I_1(r) - I_2(r)] \\ \mathscr{D} &= \frac{4\pi G}{2l+1} \frac{l(l+1)}{r^2\omega^2} [-(l+1) I_3(r) + l I_4(r)] \end{aligned} \tag{4.24}$$

and

$$\begin{aligned} b_{11} &= \frac{1}{r} \left(\frac{l(l+1)g}{r\omega^2} - 2 \right) \\ b_{12} &= \frac{1}{\lambda} - \frac{l(l+1)}{r^2\omega^2\rho} \\ b_{15} &= -\frac{l(l+1)}{r^2\omega^2} \\ b_{21} &= \rho \left(\frac{l(l+1)g^2}{r^2\omega^2} - \omega^2 - \frac{4g}{r} \right) \\ b_{22} &= -\frac{l(l+1)g}{r^2\omega^2} \\ b_{25} &= -\frac{l(l+1)\rho g}{r^2\omega^2}. \end{aligned} \tag{4.25}$$

Computational algorithm for high-speed procedure. We begin with a continental structure. In this case, y_2 and y_4 vanish at $r = a$. Thus we can write the starting vector as

$$Y_S(a) = \begin{bmatrix} y_1(a) \\ y_2(a) \\ y_3(a) \\ y_4(a) \end{bmatrix} = \begin{bmatrix} y_1(a) \\ 0 \\ y_3(a) \\ 0 \end{bmatrix} = y_1(a) \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + y_3(a) \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \tag{4.26}$$

or

$$Y_S(a) = y_1(a)X_1(a) + y_3(a)X_2(a), \tag{4.27}$$

and for $r < a$

$$Y_S(r) = y_1(a)X_1(r) + y_3(a)X_2(r). \tag{4.28}$$

The two quantities which are known—$y_1(a)$ and $y_3(a)$—can be carried implicitly in the computations, while we integrate the vectors whose starting values are known exactly: X_1 and X_2. That is, we integrate to obtain X_1 and X_2 individually, thus using equation (4.20) in the form $\bar{X}_i = AX_i$ to integrate from the surface $r = a$ to the depth at which the boundary conditions are to be applied: $r = r_0$, where we can again express Y_S in terms of the undetermined coefficients by using (4.28).

If we define a rigid boundary at depth by

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} y_1(r_0) \\ y_3(r_0) \end{bmatrix} = \begin{bmatrix} [X_1(r_0)]_1 [X_2(r_0)]_1 \\ [X_1(r_0)]_3 [X_2(r_0)]_3 \end{bmatrix} \begin{bmatrix} y_1(a) \\ y_3(a) \end{bmatrix}, \tag{4.29}$$

then the determinant of the coefficient matrix must vanish if we are to have a nontrivial solution. Thus the dispersion function takes the form

$$F(c, \omega) = \begin{vmatrix} [X_1(r_0)]_1 [X_2(r_0)]_1 \\ [X_1(r_0)]_3 [X_2(r_0)]_3 \end{vmatrix}, \tag{4.30}$$

zeros of which define valid (c, ω) dispersion pairs. As in the second section, we adopt the procedure of fixing ω and computing l, or c as defined in (2.5).

For an oceanic structure, we assume that the liquid oceanic layer is a single, homogeneous layer, i.e., that $\rho(r)$ and $\lambda(r)$ are constant. The second-order formulation is then applied, with the starting vector, $Z(a)$, specified by

$$\begin{bmatrix} y_1(a) \\ y_2(a) \end{bmatrix} = y_1(a) \begin{bmatrix} 1 \\ 0 \end{bmatrix} = y_1(a)Z(a). \tag{4.31}$$

At the bottom of the liquid layer: $r = r_1$, continuity of y_1 and y_2, and the vanishing of y_4, yield the starting vector within the solid mantle:

$$Y_S(r_1) = \begin{bmatrix} y_1(r_1) \\ y_2(r_1) \\ y_3(r_1) \\ y_4(r_1) \end{bmatrix} = \begin{bmatrix} y_1(a)[Z(r_1)]_1 \\ y_1(a)[Z(r_1)]_2 \\ y_3(r_1) \\ 0 \end{bmatrix} = y_1(a) \begin{bmatrix} [Z(r_1)]_1 \\ [Z(r_1)]_2 \\ 0 \\ 0 \end{bmatrix} + y_3(r_1) \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \tag{4.32}$$

or

$$Y_S(r_1) = y_1(a)X_1(r_1) + y_3(r_1)X_2(r_1). \tag{4.33}$$

The integration from this point on is handled exactly as in the continental case.

Although the information could be extremely useful to anyone hoping to implement the high-speed procedure, limitations on space have forced us to relegate the details concerning numerical testing of this procedure to the additional notes (section 6). The results of this testing are included in the next paragraph, and in the following "Conclusions."

The full timing details describing the algorithm for the numerical integration of our system of differential equations are quite complicated relative to the details when the homogeneous-layer approximation is used. In the latter case, the time per frequency is given precisely by

$$\tau \times n \times \mathcal{N}, \tag{4.34}$$

where τ is the single characteristic time (to treat one layer in each iteration over c, n is the number of layers, and $\mathcal{N}$ is the required number of iterations over c to obtain the desired accuracy in the phase velocity. Precise timing details for the present high-speed procedure are given in section 7 of the additional notes; here, however, we deal with a simplified, "effective" characteristic time, τ_{eff} (see additional notes, section 7), that will allow us to make immediate comparisons with times for the optimized, sixth-order algorithm described in the second section, and with the fastest of the previously published Rayleigh-wave techniques: the optimized form of Knopoff's method for nongravitating, homogeneous layers. Section 7 of the additional notes also contains the explanation of how the characteristic time of the most important program segment—the predictor-corrector portion of HPCMTL—was improved over the 143×10^{-6} sec/step/iteration reported in the second section for Figure 4c. As explained there, where complete timing details (as given in section 7 of the additional notes) were considered unnecessary, n and $\mathcal{N}$ of (4.34) can be equated approximately with the total number of integration steps and iterations over l, respectively, when using the optimized, fourth-order, Runge-Kutta and predictor-corrector methods outlined in the second section and in the additional notes (section 2). Thus, approximate relations among the computation times of the various algorithms can be obtained by simple comparison of the associated characteristic times. The times are given in Table 5, where all of the timing results from this report are summarized. These results show that the fourth-order, high-speed procedure reduces τ_{eff} to 164×10^{-6} sec/step/iteration, thus allowing gravity to be included in Rayleigh-wave computations on a sphere for only 5/4 the cost (or execution time) of calculations with nongravitating, spherical structures. For nongravitating, spherical models, the fourth-order, high-speed procedure—with τ_{eff} reduced to 132×10^{-6} sec/step/iteration—requires 6/5 the time the "fast" form of Knopoff's method needs to treat a sequence of flat, homogeneous, nongravitating layers. (The speed for nongravitating, spherical models is the same as that for our approximation with y_5 and y_6 set equal to zero while $g(r)$ is retained in the elements of the coefficient matrix. This approximation is described in the last three paragraphs of section 6 in the additional notes.) For gravitating structures, the high-speed, fourth-order procedure is roughly 1.54 times the speed of the most highly optimized form of the sixth-order algorithm. Section 7 of the additional notes contains the

explanation of how the effective characteristic time of the sixth-order algorithm was improved to 252×10^{-6} sec/step/iteration, from the lower bound of 266×10^{-6} sec/step/iteration given in the second section of the main text.

How do we control loss of precision with the improved algorithm? For computations in which the computer carries 16 to 17 decimal digits, the numerical details of this problem are given in the second section. A general discussion of the methods for treating this problem—orthogonalization and delta-matrix representations—is given in section 4 of the additional notes and in the third section of the main text.

A simplified form of the orthogonalization process described in the third section, is probably the easiest means of controlling the loss-of-precision problem in the high-speed procedure. The treatment of the oceanic layer should be satisfactory as it is, in its second-order form. Since the fourth-order simplification of the algorithm for handling solid, gravitating structures involves just two vectors X_i, the application of orthogonalization requires modification of only one vector in this case.

The application of delta-matrix representations to the sixth-order formulation would be a formidable task as it involves a twentieth-order system. The details for the delta-matrix extension of a fourth-order formulation are well known, and the above simplification, from sixth to fourth order, now makes the practical application of delta matrices possible for Rayleigh waves on a spherical, gravitating structure.

For a continental structure, instead of integrating the two vectors $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ separately, we treat the compound elements

$$\begin{aligned} W_{01} &= (X_1)_1(X_2)_2 - (X_1)_2(X_2)_1 \\ W_2 &= (X_1)_1(X_2)_3 - (X_1)_3(X_2)_1 \\ W_3 &= (X_1)_1(X_2)_4 - (X_1)_4(X_2)_1 \\ W_4 &= (X_1)_2(X_2)_3 - (X_1)_3(X_2)_2 \\ W_5 &= (X_1)_2(X_2)_4 - (X_1)_4(X_2)_2 \\ W_{06} &= (X_1)_3(X_2)_4 - (X_1)_4(X_2)_3. \end{aligned} \tag{4.35}$$

The equations of motion are still governed by the elements of the coefficient matrix of relation (4.20)

$$\begin{bmatrix} \bar{W}_{01} \\ \bar{W}_2 \\ \bar{W}_3 \\ \bar{W}_4 \\ \bar{W}_5 \\ \bar{W}_{06} \end{bmatrix} = \begin{bmatrix} a_{11}+a_{22} & A_{23} & a_{24} & -a_{13} & 0 & 0 \\ 0 & a_{11}+a_{33} & a_{34} & a_{12} & 0 & 0 \\ a_{42} & A_{43} & a_{11}+a_{44} & 0 & a_{12} & a_{13} \\ a_{33} & A_{21} & 0 & a_{22}+a_{33} & a_{34} & -a_{24} \\ -A_{41} & 0 & A_{21} & A_{43} & a_{22}+a_{44} & A_{23} \\ 0 & -A_{41} & -a_{33} & -a_{42} & 0 & a_{33}+a_{44} \end{bmatrix} \begin{bmatrix} W_{01} \\ W_2 \\ W_3 \\ W_4 \\ W_5 \\ W_{06} \end{bmatrix} \tag{4.36}$$

(Gantmacher, Chapter I.4, 1959; Gilbert and Backus, Section 2, 1966; Takeuchi and Saito, Section II.D.4, 1972). This system can be further reduced, i.e., four additional zeros can be introduced, by the transformations

$$\begin{aligned} W_1 &= \tfrac{1}{2}[W_{01} - l(l+1)W_{06}] \\ W_6 &= \tfrac{1}{2}[W_{01} + l(l+1)W_{06}], \end{aligned} \tag{4.37}$$

which yield

$$\begin{bmatrix} \bar{W}_1 \\ \bar{W}_2 \\ \bar{W}_3 \\ \bar{W}_4 \\ \bar{W}_5 \\ \bar{W}_6 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} & 0 & 0 \\ 0 & w_{22} & w_{23} & w_{24} & 0 & 0 \\ w_{31} & w_{32} & w_{33} & 0 & w_{35} & 0 \\ w_{41} & w_{42} & 0 & w_{44} & w_{45} & 0 \\ w_{51} & 0 & w_{53} & w_{54} & w_{55} & w_{56} \\ 0 & w_{62} & 0 & 0 & 0 & w_{66} \end{bmatrix} \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \\ W_5 \\ W_6 \end{bmatrix} \tag{4.38}$$

with

$$w_{11} = -2/r$$

$$w_{12} = \frac{l(l+1)}{r}\left[\rho g - \frac{2\mu(3\lambda + 2\mu)}{(\lambda + 2\mu)r}\right] - \frac{4\pi G\rho}{2l+1}\frac{l(l+1)}{2r}[l(I_1 + I_4) - (l+1)(I_2 + I_3)]$$

$$w_{13} = l(l+1)/r$$

$$w_{14} = -l(l+1)\lambda/(\lambda + 2\mu)r$$

$$w_{22} = \frac{1}{r}\left(1 - \frac{2\lambda}{\lambda + 2\mu}\right)$$

$$w_{23} = 1/\mu$$

$$w_{24} = 1/(\lambda + 2\mu)$$

$$w_{31} = -2\lambda/(\lambda + 2\mu)r$$

$$w_{32} = -\rho\omega^2 + \frac{2\mu[\lambda(2l^2 + 2l - 1) + 2\mu(l^2 + l - 1)]}{(\lambda + 2\mu)r^2} - \frac{4\pi G\rho}{2l+1}\frac{l(l+1)}{r}(I_3 + I_4)$$

$$w_{33} = -\frac{1}{r}\left(3 + \frac{2\lambda}{\lambda + 2\mu}\right)$$

$$w_{35} = w_{24}$$

$$w_{41} = -w_{11}$$

$$w_{42} = -\rho\omega^2 - \frac{4\rho g}{r} + \frac{4\mu(3\lambda + 2\mu)}{(\lambda + 2\mu)r^2} + \frac{4\pi G\rho}{2l+1}\frac{l(l+1)}{r}(I_1 + I_2)$$

$$w_{44} = \frac{1}{r}\left(1 - \frac{4\mu}{\lambda + 2\mu}\right)$$

$$w_{45} = w_{23}$$

$$w_{51} = w_{11}w_{12}/w_{13}$$

$$w_{53} = w_{42}$$

$$w_{54} = w_{32}$$

$$w_{55} = -\frac{1}{r}\left(3 + \frac{4\mu}{\lambda + 2\mu}\right)$$

$$w_{56} = \frac{4\pi G\rho}{2l+1}\frac{1}{r}[l(I_1 - I_4) - (l+1)(I_2 - I_3)]$$

$$w_{62} = \frac{l(l+1)}{2} w_{56}$$

$$w_{66} = w_{11}. \tag{4.39}$$

With this formulation for a continental structure, the starting vector is given by

$$W(a) = \begin{bmatrix} W_1(a) \\ W_2(a) \\ W_3(a) \\ W_4(a) \\ W_5(a) \\ W_6(a) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.40}$$

since all $[X_i(a)]_j$ vanish except for

$$[X_1(a)]_1 = [X_2(a)]_3 = 1. \tag{4.41}$$

Comparison of (4.30) and (4.35) will show that the dispersion function in this case is just

$$F(c, \omega) = W_2(r_0), \tag{4.42}$$

if we again employ a rigid boundary at depth to terminate the integration.

For an oceanic structure, we handle the homogeneous liquid layer in the manner described earlier in this section. Within the solid mantle, the starting values of X_i are given by (4.32) and (4.33). Thus from (4.35),

$$W(r_1) = \begin{bmatrix} 0 \\ [Z(r_1)]_1 \\ 0 \\ [Z(r_1)]_2 \\ 0 \\ 0 \end{bmatrix} \tag{4.43}$$

and the integration proceeds and terminates exactly as in the continental case.

The relative speeds of (4.20) and its delta-matrix extension, (4.38), are easily

estimated: each of the vectors $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ requires 23 elementary operations—additions, multiplications—when (4.20) is used; the single vector $\boldsymbol{W}$ requires 38 operations when (4.38) is applied. Thus, to first approximation, the delta-matrix algorithm would take roughly $38/(2 \times 23)$, or 83 per cent of the time required by (4.20) if we assume that both approaches need the same number of iterations over l. However, it must be noted that this estimate applies only to the formation of the dispersion function: subroutines HPCSAC and HPCMTL in Figure 27, and that part of this advantage is lost in the formation of $I_i(r)$ and in the eigenfunction evaluations required by the high-speed procedure. As explained in the last paragraph of section 7 in the additional notes, the *actual* reduction is to 93 per cent of the time of (4.20), or an improvement in the effective characteristic time from 164×10^{-6}, to 153×10^{-6} sec/step/iteration by using (4.38).

When r_0 increases beyond the position of the "400-km discontinuity," the approximation with y_5 and y_6 equal to zero is used. Orthogonilization can be applied to this approximation in exactly the same manner as with the unmodified high-speed procedure. The delta-matrix formulation appropriate to this approximation is obtained from (4.38) and (4.39) by letting $I_i(r)$ go to zero. This reduces the system to fifth order: $\bar{W}_6$, W_6, and the last row and column of the coefficient matrix can be deleted from (4.38). Thus the reduced, single vector $\boldsymbol{W}$ now requires only 33 operations, and the conversion to the delta-matrix algorithm with vanishing y_5 and y_6 reduces computation time to $33/(2 \times 23)$, or 72 per cent of that required by (4.20). The effective characteristic time is therefore improved from 132×10^{-6}, to 95×10^{-6} sec/step/iteration.

The methods used in the numerical integration of differential equations include one-step and multi-step procedures; the initiation of a multi-step method requiring the results of a specified number of applications of a self-starting, one-step method. The high-speed procedure described here and in section 6 of the additional notes, is strongly analogous to such a multi-step method, and for each (radial) mode requires the results of three applications of a self-starting procedure that does not need an *a priori* prediction of $y_1(r)$ and $y_3(r)$ at the frequency being treated. The most obvious choice for this "starting" procedure is the full, sixth-order algorithm; ideally, the augmented algorithm of the third section which controls the loss-of-precision problem by means of orthogonalization. In section 8 of the additional notes we present an investigation of the usefulness of a fourth-order "starting" procedure, which is based on "assumption 2" of the present section, and look for any significant advantages over the sixth-order algorithm.

Conclusions. Previous algorithms for the computation of Rayleigh-wave dispersion on a spherical, gravitating earth, have been based on the integration of a sixth-order system of differential equations. In its most highly optimized form (Table 5), the algorithm for treating this system is still about five times slower ($\tau_{eff} = 252 \times 10^{-6}$ sec/step/interation) than the comparable Love-, or torsional-wave calculations, and about 2.3 times slower than the fastest Rayleigh-wave calculations based on the nongravitating, homogeneous-layer approximation. In this section it is shown that the sixth-order formulation can be reduced to one of fourth order, while still retaining the effect of gravity. This simplification makes it possible to compute Rayleigh-wave dispersion on spherical, gravitating structures, with 4/5 the speed ($\tau_{eff} = 164 \times 10^{-6}$ sec/step/iteration) of the optimized algorithm for the comparable calculation with a nongravitating sphere ($\tau_{eff} = 132 \times 10^{-6}$ sec/step/iteration). The new high-speed procedure for Rayleigh-wave computations on a spherical, gravitat-

ing earth is only three times slower than the torsional-wave computations we are already using to calculate complete, torsional-wave seismograms for realistic models of the earth. Thus, at least in terms of speed, it now appears feasible to attempt to compute complete, spheroidal-wave seismograms containing all of the energy generated down to a period of 10 sec.

This fourth-order, high-speed procedure is designed for the computation of many dispersion points at equally spaced frequencies. To initiate this process for each radial mode, eigenfunction and dispersion data at the three longest-period points are needed. For this purpose, a fourth-order "starting" procedure has been described (section 8 of the additional notes). Owing to the key optimization technique—evaluation of the coefficient-matrix elements of (4.20), (4.22), (65), and (67) outside of the innermost, integration loops—it is possible to use the same integration subroutines (or program segments) for the fourth-order high-speed and "starting" procedures. (Equation numbers that lack periods refer to relations given in the additional notes.) This use of the same subroutines results in considerable coding efficiency, and is justified if $[(N_l)_{4\text{th}}]_{\text{START}}$ is not too much above $(N_l)_{6\text{th}}$ (see "Computational Algorithm" in section 8 of the additional notes).

Automatic structure reduction is an intrinsic feature of the high-speed procedure as it treats shorter and shorter periods for the same radial mode. When the structure has been reduced to the point where the integration is terminated just below the low-velocity channel in the upper mantle, it is advisable to switch to an even faster algorithm which allows y_5 and y_6 to vanish while retaining $g(r)$ in the elements of the coefficient matrix. Otherwise, one would have to modify the very efficient integration in XINTEG (Figure 27) for obtaining $I_i(r)$. Since this new, faster algorithm—same speed as that for a nongravitating sphere—maintains the desired four-figure accuracy in c for these shallow depths of penetration, it should of course be used when r_0 increases beyond the depth of the "400-km discontinuity." That the faster algorithm can be utilized in the channel wave-crustal wave region of the dispersion curves is particularly fortuitous: the eigenfunction extrapolation procedure therefore need not be attempted for these period ranges in which the energy associated with a given radial mode is abruptly shifting up and down, between the low-velocity zone and the crust, as period varies.

If the value of $[(N_l)_{4\text{th}}]_{\text{START}}$ is sufficiently small (see above), then consideration of coding efficiency will not only indicate that the fourth-order "starting" procedure be used with the high-speed procedure, but this will also be an argument for treating the loss-of-precision problem by orthogonalization. For the application in which we are mainly interested, a feature for controlling precision loss must be included with the high-speed algorithm. Thus the computational speeds that are really of interest are those which include such a feature. The use of orthogonalization increases the above timing estimates by 5 to 10 per cent; the argument in favor of using delta matrices to control precision loss is that this *decreases* computation time. For a gravitating earth, the reduction is approximately 7 per cent ($\tau_{eff} = 153 \times 10^{-6}$ sec/step/iteration) by using representation (4.38) instead of (4.20), if we assume that both approaches require the same number of iterations over l to obtain the desired accuracy. With the structure reduced to the point where it is allowable to use the algorithm which permits y_5 and y_6 to vanish, delta matrices can be combined with the optimization indicated in Figure 1 to yield the fastest Rayleigh-wave algorithm ($\tau_{eff} = 95 \times 10^{-6}$ sec/step/iteration) that has been developed to date.

For ease of reference, our final timing results are summarized in the following table.

Computational Technique	Reference	Control of Precision Loss	Relative Computation Time	Wave Type	Structure
homogeneous-layer approximation	Schwab and Knopoff (1972)	—*	1.0	Love, or torsional	flat, or spherical
	Schwab and Knopoff (1972)	—†	2.0	Rayleigh	flat, nongravitating
direct-integration methods	high speed procedures			spheroidal	spherical, approximate with $y_5 = y_6 = 0$ but $g(r)$ left in a_{ij}
	(4.20)	none	2.4		
	(4.20)	orthogonalization	2.6		
	(4.38)	delta matrix	1.7		
	high-speed procedures (Figure 27)			spheroidal	spherical, gravitating
	(4.20)	none	3.0		
	(4.20)	orthogonalization	3.2		
	(4.38)	delta matrix	2.8		
	"starting" procedures			spheroidal	spherical, gravitating
	(2.1)	none	4.6		
	(2.1)	orthogonalization	4.9		
	(65)	none	4.7		
	(65)	orthogonalization	5.1		
	(76)	delta matrix	5.1		

* There is no loss-of-precision problem in this case.

† This case refers to the optimized form of Knopoff's method, which contains the equivalent of the delta-matrix approach for controlling precision loss (Schwab, 1970).

The main application envisioned for the high-speed procedure is the calculation of complete, spheroidal-wave seismograms. Only a few routine details remain to be taken care of before attempting this application.

1. The specific details, for all (radial) modes, should be investigated concerning the point at which to switch from the high-speed procedure for gravitating structures, to the special algorithm for shallow penetration. This point could be given as a specific frequency for each mode, but a more useful and general criterion might be a single, critical depth of penetration.

2. The best "starting" procedure and technique for controlling loss of precision should be decided upon. Considerations based on coding efficiency favor the fourth-order "starting" procedure, the multi-step, high-speed procedure (Figure 27) using representation (4.20), and orthogonalization to control precision loss; if computation time is governing consideration, there is little to choose between the two "starting" procedures, but the multi-step, high-speed procedure should be used in the form (4.38), i.e., with delta matrices employed to control precision loss.

3. If orthogonalization is chosen to treat loss of precision, then the point at which to begin application of this method should be investigated. This point also could be

given as a specific frequency for each (radial) mode number; however, it might be more desirable to represent this criterion as a single function. Extended to cover the entire period range of interest, this function would correspond to the line for $\sigma =$ four significant figures in Figure 15.

4. Since the difficult algorithmic problems concern the solid portions of the structure, the main emphasis in this section has been on dealing with the crust-mantle system. As Liao *et al.* (1978) show, the complete theoretical seismograms that we are interested in can exhibit important body-wave phases reflected from the mantle-core boundary. Thus we should at least touch upon the question of including the liquid, outer section of the core in our integration loops. For the fourth-order high-speed and "starting" procedures, representation (2.16)—with ρ and λ varying with r in this case—can be used in the outer core. Only a single vector need be integrated here. Its starting value is obtained from the continuity of y_1 and y_2, and the vanishing of y_4 at the mantle-core boundary; (4.18) and (4.19) are then used with the high-speed procedure, and with the "starting" procedure, (62) and (63) combined with the first and third equations of (65). Loss-of-precision problems may require that the fourth-order formulation, (2.16), be reduced to one of second order with ρ and λ allowed to vary with r.

5. For radial mode numbers greater than about five, as period decreases from 25 to 10 sec the maximum integration step size must be reduced from 12.5 km to maintain four significant figures in the computed dispersion. (See the last paragraph of the third section for further details.) Since this reduction will significantly increase computation time, the *precise* details concerning when the decrease must be effected should be investigated.

The high-speed procedure for Rayleigh waves on a spherical, gravitating earth now makes it economically feasible to do with spheroidal waves, what has recently been shown possible for torsional waves: to compute complete theoretical seismograms that are suitable for direct comparison with records from the long-period instruments of the WWSSN, i.e., to compute synthetic seismograms containing all body- and surface-wave energy that is generated down to a period of 10 sec. With the capability to synthesize both spheroidal and torsional contributions, comparisons will be possible with the experimental records of all three components.

5. Acknowledgments

This research was partly supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by the Air Force Office of Scientific Research under Contract F49620-76-C-0038, partly supported by the Earth Sciences Section, National Science Foundation Grants DES75-04376 and EAR75-04376, and partly supported by the North Atlantic Treaty Organization, NATO Grant 1497. Part of the work was carried out while F. Schwab and A. H. Liao were Visiting Scientists at the University of Chile (Santiago), supported by the Multinational Project on Earth Sciences (Regional Scientific and Technological Development Program) of the Organization of American States. The original program development and testing, and many of the algorithmic tests, were carried out at the Campus Computing Network of the University of California at Los Angeles; we wish to express our gratitude for that portion of the computations which was supported by intramural, university computing funds. A large part of the numerical testing was performed at the Dipartimento di Scienza della Terra, Università della Calabria and at the Instituto di Geodesia e Geofisica, Università di Bari; we are grateful to these institutions for supporting our computations. We also wish to express our gratitude to the Centro de Computación, Universidad de Chile (Santiago) for supporting the remainder—and not insignificant portion—of our numerical analyses.

We are grateful to Dr. J. K. Gardner for advice concerning optimum usage of IBM 360 hardware, and to Dr. F. Gilbert for the initial version of a spheroidal-mode (variable period) program that has provided results with which to compare the computational features of the various Rayleigh-wave (variable phase-velocity) algorithms described in the text.

6. References

Alterman, Z., H. Jarosch, and C. L. Perkeris (1959). Oscillations of the Earth, *Proc. Roy. Soc., A* **252,** 80–95.

Alterman, Z., H. Jarosch, and C. L. Pekeris (1961). Propagation of Rayleigh waves in the Earth, *Geophys. J.* **4,** 219–241.

Anon. (1970). *System/360 Scientific Subroutine Package, Version III Programmer's Manual* (Program Number 360A-CM-03X), IBM Corporation, White Plains, New York, 343–350.

Bhattacharya, S. N. (1976). Extension of the Thomson-Haskell method to non-homogeneous spherical layers, *Geophys. J.* **47,** 411–443.

Biswas, N. N. (1972). Earth-flattening procedure for the propagation of Rayleigh waves, *Pure and Appl. Geophys.* **96,** 61–74.

Biswas, N. N. and L. Knopoff (1970). Exact earth-flattening calculation for Love waves, *Bull. Seism. Soc. Am.* **60,** 1123–1137.

Bolt, B. A. and J. Dorman (1961). Phase and group velocities of Rayleigh waves in a spherical, gravitating earth, *J. Geophys. Res.* **66,** 2965–2981.

Dorman, J., M. Ewing, and J. Oliver (1960). Study of shear-velocity distribution in the upper mantle by mantle Rayleigh waves, *Bull. Seism. Soc. Am.* **50,** 87–115.

Gantmacher, F. R. (1959). *The Theory of Matrices*, vol. 1 (Chapter I, Section 4) and vol. 2 (Chapter XIV, Section 2), Chelsea Publishing Co., New York.

Gilbert, F. and G. E. Backus (1966). Propagator matrices in elastic wave and vibrational problems, *Geophysics* **31,** 326–332.

Gilbert, F. and A. M. Dziewonski (1975). An application of normal mode theory to the retrieval of structural parameters and source mechanisms from seismic spectra, *Phil. Trans. Roy. Soc. Lond., A* **278,** 187–269.

Hamming, R. W. (1959). Stable predictor corrector methods for ordinary differential equations, *J. Assoc. Comp. Mach.* **6,** 37–47.

Haskell, N. A. (1953). The dispersion of surface waves on multilayered media, *Bull. Seism. Soc. Am.* **43,** 17–34.

Hoskins, L. M. (1920). The strain of a gravitating sphere of variable density and elasticity. *Trans. Am. Math. Soc.* **21,** 1–43.

Kausel, E. and F. Schwab (1973). Contributions to Love-wave transformation theory: earth-flattening transformation for Love waves from a point source in a sphere, *Bull. Seism. Soc. Am.* **63,** 983–993.

Knopoff, L. (1964). A matrix method for elastic wave problems, *Bull. Seism. Soc. Am.* **54,** 431–438.

Lawson, C. L. and R. J. Hanson (1974). *Solving Least Squares Problems*, Prentice-Hall, New Jersey.

Liao, A. H., F. Schwab, and E. Mantovani (1977). Computation of complete theoretical seismograms (abstract), *Geol. Soc. Am., Abstracts* **9,** 453.

Liao, A. H., F. Schwab, and E. Mantovani (1978). Computation of complete theoretical seismograms for torsional waves, *Bull. Seism. Soc. Am.* **68,** 317–324.

Neigauz, M. G. and G. V. Shkadinskaya (1972). Method for calculating surface Rayleigh waves in a vertically inhomogeneous half-space, in *Computational Seismology*, V. I. Keilis-Borok, Editor, Consultants Bureau, New York, 88–92.

Nolet, A. M. H. (1976). *Higher Modes and the Determination of Upper Mantle Structure, Ph.D. Thesis*, University of Utrecht.

Pekeris, C. L. (1966). The internal constitution of the Earth, *Geophys. J.* **11,** 85–132.

Pekeris, C. L. and H. Jarosch (1958). The free oscillations of the Earth, in *Contributions in Geophysics*, vol. 1, Pergamon Press, London, 171–192.

Porter, L. D., F. Schwab, K. K. Nakanishi, I. F. Weeks, G. F. Panza, E. Mantovani, D. McMenamin, W. D. Smythe, A. H. Liao, J. A. Landoni, N. N. Biswas, F.-S. Chang, S. S. Bor, E. G. Kausel, P. Gasperini, and J. L. Luthey (1980). Relative computer speeds for surface-wave dispersion computations, *Bull. Seism. Soc. Am.* **70,** 1415–1420.

Press, F., D. Harkrider, and C. A. Seafeldt (1961). A fast, convenient program for computation of surface-wave dispersion curves in multilayered media, *Bull. Seism. Soc. Am.* **51,** 495–502.

Ralston, A. (1960). Numerical integration methods for the solution of ordinary differential equations, in *Mathematical Methods for Digital Computers*, vol. 1, A. Ralston and H. S. Wilf, Editors, John Wiley and Sons, New York, 95–109.

Ralston, A. (1962). Runge-Kutta methods with minimum error bounds, *Math. Comp.* **16,** 431–437.

Randall, M. J. (1967). Fast programs for layered half-space problems, *Bull. Seism. Soc. Am.* **57,** 1299–1316.

Schwab, F. (1970). Surface-wave dispersion computations: Knopoff's method, *Bull. Seism. Soc. Am.* **60,** 1491–1520.

Schwab, F. and E. Kausel (1976). Long-period surface wave seismology: Love wave phase velocity and polar phase shift, *Geophys. J.* **45,** 407–435.

Schwab, F. and L. Knopoff (1970). Surface-wave dispersion computations, *Bull. Seism. Soc. Am.* **60,** 321–344.

Schwab, F. and L. Knopoff (1971). Surface waves on multilayered anelastic media, *Bull. Seism. Soc. Am.* **61,** 893–912.

Schwab, F. and L. Knopoff (1972). Fast surface wave and free mode computations, in *Methods in Computational Physics*, vol. 11, B. A. Bolt, Editor, Academic Press, New York, 87–180.

Schwab, F. and L. Knopoff (1973). Love waves and the torsional free modes of a multilayered anelastic sphere, *Bull. Seism. Soc. Am.* **63,** 1107–1117.

Takeuchi, H. and M. Saito (1972). Seismic surface waves, in *Methods in Computational Physics*, vol. 11, B. A. Bolt, Editor, Academic Press, New York, 217–295.

Thomson, W. T. (1950). Transmission of elastic waves through a stratified solid medium, *J. Appl. Phys.* **21,** 89–93.

Wiggins, R. A. (1968). Terrestrial variational tables for the periods and attenuations of the free oscillations, *Phys. Earth Planet. Interiors* **1,** 201–266.

Wiggins, R. A. (1976). A fast, new computational algorithm for free oscillations and surface waves, *Geophys. J.* **47,** 135–150.

INSTITUTE OF GEOPHYSICS AND PLANETARY PHYSICS
UNIVERSITY OF CALIFORNIA
LOS ANGELES, CALIFORNIA 90024
CONTRIBUTION NO. 1700 (F.S., J.F., A.H.L.)

UNIVERSITÀ DI TRIESTE
34100 TRIESTE, ITALIA (G.F.P)

DEPARTAMENTO DE GEOFISICA
UNIVERSIDAD DE CHILE
CASILLA 2777
SANTIAGO, CHILE
CONTRIBUTION NO. 121 (E.G.K.)

Manuscript received May 11, 1977

ADDITIONAL NOTES

1. ALGORITHMS FOR GROUP VELOCITY AND PARTIAL DERIVATIVE DETERMINATION

When treating Rayleigh waves on a spherical, gravitating earth, the variational technique is usually employed to compute group velocities and partial derivatives with respect to the structural parameters (see, for example, Takeuchi and Saito, 1972, Section III). Here, we believe for the first time, we propose direct computational techniques for the determination of these quantities on a gravitating earth. The practical value of direct algorithms for these purposes became apparent when it was necessary to solve the problem of efficient computation of an average of about 100 dispersion points for each of 90-100 torsional-wave modes, all in a single computer run. (A full description of this method is in preparation for future publication. Interested investigators can obtain the description from the program package and documentation, which are available from the senior author. Results of the first application have been given by Liao et al., 1977; 1978.) The advantage of the direct approach is is rather obvious: Roots of a dispersion function, F, are sought to determine phase velocities; the speed with which these roots can be determined, at each frequency, is improved if it is possible to calculate

$$\left(\frac{\partial}{\partial c}\right)_{\omega} F \tag{1}$$

as well as F at each iteration over c; hence, it is extremely useful to be able to compute this partial derivative at the same time as F. This is possible with a direct computational algorithm. Since

$$\left(\frac{\partial}{\partial \omega}\right)_{c} F \tag{2}$$

is also available from our direct computational scheme, and since the dispersion function vanishes when the point (c,ω) falls on a dispersion

curve, we can use implicit function theory to obtain

$$\frac{dc}{d\omega} = -\frac{(\partial/\partial\omega)_c F}{(\partial/\partial c)_\omega F} \tag{3}$$

which yields both the group velocity

$$u = c \Big/ \left(1 - \frac{\omega}{c}\frac{dc}{d\omega}\right) \tag{4}$$

and the slope of the dispersion curve at the frequency of interest, say ω_i. With c and the slope (3) at ω_i, we easily obtain an accurate extrapolate $c(\omega_{i+1})$, with which to begin the iterative process for the phase velocity at this next frequency. If we agree to use the rigid boundary at depth, then

$$F' = \left(\frac{\partial F}{\partial \omega}\right)_c = \begin{vmatrix} [X_1'(r_0)]_1 & [X_2'(r_0)]_1 & [X_3'(r_0)]_1 \\ [X_1(r_0)]_3 & [X_2(r_0)]_3 & [X_3(r_0)]_3 \\ [X_1(r_0)]_5 & [X_2(r_0)]_5 & [X_3(r_0)]_5 \end{vmatrix}$$

$$+ \begin{vmatrix} [X_1(r_0)]_1 & [X_2(r_0)]_1 & [X_3(r_0)]_1 \\ [X_1'(r_0)]_3 & [X_2'(r_0)]_3 & [X_3'(r_0)]_3 \\ [X_1(r_0)]_5 & [X_2(r_0)]_5 & [X_3(r_0)]_5 \end{vmatrix} + \begin{vmatrix} [X_1(r_0)]_1 & [X_2(r_0)]_1 & [X_3(r_0)]_1 \\ [X_1(r_0)]_3 & [X_2(r_0)]_3 & [X_3(r_0)]_3 \\ [X_1'(r_0)]_5 & [X_2'(r_0)]_5 & [X_3'(r_0)]_5 \end{vmatrix} \tag{5}$$

and

$$\dot{F} = \left(\frac{\partial F}{\partial c}\right)_\omega \tag{6}$$

is given by the same type of expression as (5), with dots replacing the primes. The elements $[X_i(r_0)]_j$ in (5) and (6) are obtained exactly as described in the second section of the main text. The evaluation of $[X_i'(r_0)]_j$ and $[\dot{X}_i(r_0)]_j$ requires a simple extension of the algorithm.

Continental structure. In this case we start with the sixth-order system

$$\bar{Y} = AY \tag{7}$$

and form

$$\overline{Y'} = A'Y + AY' \tag{8}$$

$$\dot{\overline{Y}} = \dot{A}Y + A\dot{Y} \quad . \tag{9}$$

Here again, we use these equations of motion in terms of the vectors $\underline{X_i}$, $\underline{X_i'}$, $\underline{\dot{X}_i}$, that we know exactly at the surface:

$$\overline{X_i'} = A'X_i + AX_i' \tag{10}$$

$$\dot{\overline{X}}_i = \dot{A}X_i + A\dot{X}_i \quad . \tag{11}$$

Since $\underline{X_i}$ can be determined independently, we can treat $\underline{A'X_i}$ and $\underline{\dot{A}X_i}$ as known vectors at each depth, and we have

$$\overline{X_i'} = AX_i' + C_i \tag{12}$$

$$\dot{\overline{X}}_i = \dot{A}X_i + D_i \quad , \tag{13}$$

where

$$C_i = A'(r)\, X_i(r) = -2\,\rho(r)\omega \begin{bmatrix} 0 \\ [X_i(r)]_1 \\ 0 \\ [X_i(r)]_3 \\ 0 \\ 0 \end{bmatrix} \tag{14}$$

$$D_i = \dot{A}(r)\, X_i(r) = -\frac{a\omega}{c^2 r}(2\ell+1) \begin{bmatrix} [\lambda/(\lambda+2\mu)]\ [X_i(r)]_3 \\ [\rho g - 2\mu(3\lambda+2\mu)/(\lambda+2\mu)r]\ [X_i(r)]_3 + [X_i(r)]_4 \\ 0 \\ [4\mu(\lambda+\mu)/(\lambda+2\mu)r]\ [X_i(r)]_3 \\ 0 \\ [-4\pi G\rho]\ [X_i(r)]_3 + (1/r)\ [X_i(r)]_5 \end{bmatrix} . \tag{15}$$

Oceanic structure. Here we begin with the fourth-order system (2.16), and form

$$\begin{bmatrix} \overline{y_1'} \\ \overline{y_2'} \\ \overline{y_5'} \\ \overline{y_6'} \end{bmatrix} = B' \begin{bmatrix} y_1 \\ y_2 \\ y_5 \\ y_6 \end{bmatrix} + B \begin{bmatrix} y_1' \\ y_2' \\ y_5' \\ y_6' \end{bmatrix} \tag{16}$$

$$\begin{bmatrix} \overline{\dot{y}_1} \\ \overline{\dot{y}_2} \\ \overline{\dot{y}_5} \\ \overline{\dot{y}_6} \end{bmatrix} = \dot{B} \begin{bmatrix} y_1 \\ y_2 \\ y_5 \\ y_6 \end{bmatrix} + B \begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \dot{y}_5 \\ \dot{y}_6 \end{bmatrix} \tag{17}$$

These equations are then used with the vectors for which we have solutions at the surface, $\underline{Z_i}$:

$$\overline{Z_i'} = B'Z_i + BZ_i' \tag{18}$$

$$\overline{\dot{Z}_i} = \dot{B}Z_i + B\dot{Z}_i \quad , \tag{19}$$

which can be written in terms of known vectors, $\underline{B'Z_i}$ and $\underline{\dot{B}Z_i}$, at each depth:

$$\overline{Z_i'} = BZ_i' + E_i \tag{20}$$

$$\overline{\dot{Z}_i} = B\dot{Z}_i + G_i \quad , \tag{21}$$

where

$$E_i(r) = B'(r)\, Z_i(r) = [2\,\ell(\ell+1)/r^2\omega^3] \begin{bmatrix} -H_i(r) \\ U_i(r) \\ 0 \\ [4\pi G\,\rho(r)]\, H_i(r) \end{bmatrix} \tag{22}$$

$$G_i(r) = \dot{B}(r)\, Z_i(r) = [-a\,(2\ell+1)/c^2r^2\omega] \begin{bmatrix} H_i(r) \\ [g(r)\,\rho(r)]\, H_i(r) \\ 0 \\ V_i(r) \end{bmatrix} \tag{23}$$

and

$$H_i(r) = g(r)\,[Z_i(r)]_1 - [1/\rho(r)]\,[Z_i(r)]_2 - [Z_i(r)]_5 \tag{24}$$

$$U_i(r) = -\rho(r)[\omega^4 r^2/\ell(\ell+1) + g^2(r)]\,[Z_i(r)]_1 + g(r)\{[Z_i(r)]_2 + \rho(r)[Z_i(r)]_5\} \tag{25}$$

$$V_i(r) = 4\pi G\{-\rho(r)g(r)\,[Z_i(r)]_1 + [Z_i(r)]_2 + [\rho(r) + \omega^2/4\pi G][Z_i(r)]_5\} \ . \tag{26}$$

Application of (20) and (21) will allow us to carry the integration to the bottom of the oceanic layer at $r = \underline{r_1}$, where we can apply the boundary conditions of continuity of $\underline{y_1}$, $\underline{y_1'}$, $\underline{\dot{y}_1}$, $\underline{y_2}$, $\underline{y_2'}$, $\underline{\dot{y}_2}$, $\underline{y_5}$, $\underline{y_5'}$, $\underline{\dot{y}_5}$, $\underline{y_6}$, $\underline{y_6'}$, $\underline{\dot{y}_6}$, and the vanishing of $\underline{y_4}$, $\underline{y_4'}$, $\underline{\dot{y}_4}$, to obtain the necessary starting values to apply in (12) and (13):

$$X_1'(r_1) = \begin{bmatrix} [Z_1'(r_1)]_1 \\ [Z_1'(r_1)]_2 \\ 0 \\ 0 \\ [Z_1'(r_1)]_5 \\ [Z_1'(r_1)]_6 \end{bmatrix} \quad , \quad X_2'(r_1) = 0 \quad , \quad X_3'(r_1) = \begin{bmatrix} [Z_2'(r_1)]_1 \\ [Z_2'(r_1)]_2 \\ 0 \\ 0 \\ [Z_2'(r_1)]_5 \\ [Z_2'(r_1)]_6 \end{bmatrix} \tag{27}$$

and a like set of starting values with primes replaced by dots.

The direct computation of partial derivatives of $\underline{c}$ with respect to structural parameters follows the same lines as the direct evaluation of $\underline{u}$. For example,

$$\left(\frac{\partial c}{\partial \rho(r_k)}\right)_{\omega,p_k} = -\frac{\left(\frac{\partial}{\partial \rho(r_k)}\right)_{\omega,c,p_k} F}{\left(\frac{\partial}{\partial c}\right)_{\omega,p_k,\rho(r_k)} F} \tag{28}$$

where

$$p_k = (\ \rho(r_i), i=1(1)k-1, k+1(1)N\ ), (\ \lambda(r_i), \mu(r_i), i=1(1)N\ ) \quad , \tag{29}$$

$\underline{\lambda}$ and $\underline{\mu}$ are the Lamé constants, and $\underline{N}$ is the number of points at which the structure is defined. The partial derivative in the numerator of (28) is defined by an expression having the form of (5), and the expressions for the determination of

$$\left(\frac{\partial X_i(r)}{\partial \rho(r_k)}\right)_{\omega,c,p_k} \tag{30}$$

are obtained exactly as above, with the obvious simplifications introduced by the fact that here

$$\left(\frac{\partial A(r_j)}{\partial \rho(r_k)}\right)_{\omega,c,p_k} = 0 \tag{31}$$

when $r_j \neq r_k$.

2. RUNGE-KUTTA TECHNIQUE FOR STARTING PREDICTOR-CORRECTOR METHOD

In terms of a single, first-order differential equation:

$$\dot{\eta} = f(r,\eta) \quad , \quad \eta(r_{start}) = \eta_{start} \quad , \tag{32}$$

at $r_{start+1}$, $r_{start+2}$, $\cdots$, the Runge-Kutta method is given by

$$\eta_{m+1} - \eta_m = \sum_{i=1}^{4} w_i W_i \quad . \tag{33}$$

Here, $\eta_m = \eta(r_m)$, the w_i are constants, and

$$W_i = s_m f\left(r_m + \alpha_i s_m, \eta_m + \sum_{j=1}^{i-1} \beta_{ij} W_j\right) \tag{34}$$

with $s_m = r_{m+1} - r_m$, and

$$\begin{aligned} \alpha_1 &\quad 0 \\ \alpha_2 &\quad 2/5 \\ \alpha_3 &\quad (14 - 3\sqrt{5})/16 \\ \alpha_4 &\quad 1 \end{aligned} \tag{35}$$

$$\begin{aligned} \beta_{21} &= \alpha_2 \\ \beta_{31} &= \alpha_3 - \beta_{32} \\ \beta_{32} &= [\alpha_3(\alpha_3 - \alpha_2)]/[2\alpha_2(1 - 2\alpha_2)] \\ \beta_{41} &= \alpha_4 - \beta_{42} - \beta_{43} \\ \beta_{42} &= \frac{(1 - \alpha_2)[\alpha_2 + \alpha_3 - 1 - (2\alpha_3 - 1)^2]}{2\alpha_2(\alpha_3 - \alpha_2)[6\alpha_2\alpha_3 - 4(\alpha_2 + \alpha_3) + 3]} \\ \beta_{43} &= \frac{(1 - 2\alpha_2)(1 - \alpha_2)(1 - \alpha_3)}{\alpha_3(\alpha_3 - \alpha_2)[6\alpha_2\alpha_3 - 4(\alpha_2 + \alpha_3) + 3]} \end{aligned} \tag{36}$$

$$
\begin{aligned}
w_1 &= \frac{1}{2} + [1 - 2(\alpha_2 + \alpha_3)]/12\alpha_2\alpha_3 \\
w_2 &= (2\alpha_3 - 1)/[12\alpha_2(\alpha_3 - \alpha_2)(1 - \alpha_2)] \\
w_3 &= (1 - 2\alpha_2)/[12\alpha_3(\alpha_3 - \alpha_2)(1 - \alpha_3)] \\
w_4 &= \frac{1}{2} + [2(\alpha_2 + \alpha_3) - 3]/[12(1 - \alpha_2)(1 - \alpha_3)] \quad .
\end{aligned} \tag{37}
$$

3. TERMINATING BOUNDARY CONDITIONS

When the structure used to form the dispersion function is terminated within the mantle by a gravitating, homogeneous, solid sphere below $\underline{r_0}$, above $\underline{r_0}$ we have

$$Y_+(r_0) = y_1(a)\, X_1(r_0) + y_3(b)\, X_2(r_0) + y_5(a)\, X_3(r_0) \tag{38}$$

where

$$b = \begin{cases} a & \text{for a continental structure} \\ r_1 & \text{for an oceanic structure.} \end{cases} \tag{39}$$

For the homogeneous, gravitating, solid sphere below $\underline{r_0}$, there are three classes of solutions: $\underline{Y_1}(r)$, $\underline{Y_2}(r)$, $\underline{Y_3}(r)$; thus, just below $\underline{r_0}$,

$$Y_-(r_0) = D\, Y_1(r_0) + E\, Y_2(r_0) + M\, Y_3(r_0) \quad , \tag{40}$$

where $\underline{D}$, $\underline{E}$, and $\underline{M}$ are undetermined coefficients. Applying the boundary conditions of continuity of $\underline{y_i}$ at $\underline{r_0}$, we obtain

$$\begin{bmatrix} [X_1(r_0)]_1 & [X_2(r_0)]_1 & [X_3(r_0)]_1 & -[Y_1(r_0)]_1 & -[Y_2(r_0)]_1 & -[Y_3(r_0)]_1 \\ [X_1(r_0)]_2 & [X_2(r_0)]_2 & [X_3(r_0)]_2 & -[Y_1(r_0)]_2 & -[Y_2(r_0)]_2 & -[Y_3(r_0)]_2 \\ [X_1(r_0)]_3 & [X_2(r_0)]_3 & [X_3(r_0)]_3 & -[Y_1(r_0)]_3 & -[Y_2(r_0)]_3 & -[Y_3(r_0)]_3 \\ [X_1(r_0)]_4 & [X_2(r_0)]_4 & [X_3(r_0)]_4 & -[Y_1(r_0)]_4 & -[Y_2(r_0)]_4 & -[Y_3(r_0)]_4 \\ [X_1(r_0)]_5 & [X_2(r_0)]_5 & [X_3(r_0)]_5 & -[Y_1(r_0)]_5 & -[Y_2(r_0)]_5 & -[Y_3(r_0)]_5 \\ [X_1(r_0)]_6 & [X_2(r_0)]_6 & [X_3(r_0)]_6 & -[Y_1(r_0)]_6 & -[Y_2(r_0)]_6 & -[Y_3(r_0)]_6 \end{bmatrix} \begin{bmatrix} y_1(a) \\ y_3(b) \\ y_5(a) \\ D \\ E \\ M \end{bmatrix} = 0 \tag{41}$$

or

$$C\, W_{\text{SOLID SPHERE}} = 0 \quad , \tag{42}$$

and the dispersion function has the form

$$F(\omega, c) = \det(C) \quad . \tag{43}$$

The components of $\underline{Y_i}(r)$ are given in convenient form by Takeuchi and Saito (1972): $\underline{Y_1}(r)$ by their equations (98), with the negative sign in (99);

$\underline{Y}_2(r)$ by (98), with the positive sign in (99); and $\underline{Y}_3(r)$ by their equations (100). Note that their definition of $\underline{y}_6$ differs slightly from that used here.

When the structure used to form the dispersion function is terminated at the mantle-core boundary by the conditions for a homogeneous, liquid sphere below $\underline{r}_0$, above $\underline{r}_0$ we have (38); below,

$$\underline{Y}_-(r_0) = P\, Y_1(r_0) + Q\, Y_3(r_0) \quad , \tag{44}$$

where $\underline{P}$ and $\underline{Q}$ are undetermined constants, and $\underline{Y}_i$ in (44) have the form

$$Y_i(r_0) = \begin{bmatrix} y_1(r_0) \\ y_2(r_0) \\ y_5(r_0) \\ y_6(r_0) \end{bmatrix} \tag{45}$$

Again, Takeuchi and Saito (1972) give the form of these vector components for the gravitating, homogeneous, liquid sphere. From the conditions of continuity of $\underline{y}_1$, $\underline{y}_2$, $\underline{y}_5$, $\underline{y}_6$ at $\underline{r}_0$, and the vanishing of $\underline{y}_4(\underline{r}_0)$, we have

$$\begin{bmatrix} [X_1(r_0)]_1 & [X_2(r_0)]_1 & [X_3(r_0)]_1 & -[Y_1(r_0)]_1 & -[Y_3(r_0)]_1 \\ [X_1(r_0)]_2 & [X_2(r_0)]_2 & [X_3(r_0)]_2 & -[Y_1(r_0)]_2 & -[Y_3(r_0)]_2 \\ [X_1(r_0)]_4 & [X_2(r_0)]_4 & [X_3(r_0)]_4 & 0 & 0 \\ [X_1(r_0)]_5 & [X_2(r_0)]_5 & [X_3(r_0)]_5 & -[Y_1(r_0)]_5 & -[Y_3(r_0)]_5 \\ [X_1(r_0)]_6 & [X_2(r_0)]_6 & [X_3(r_0)]_6 & -[Y_1(r_0)]_6 & -[Y_3(r_0)]_6 \end{bmatrix} \begin{bmatrix} y_1(a) \\ y_3(b) \\ y_5(a) \\ P \\ Q \end{bmatrix} = 0 \tag{46}$$

or

$$R\, S_{\text{LIQUID SPHERE}} = 0 \quad , \tag{47}$$

and the dispersion function takes the form

$$F(\omega, c) = \det (R) \quad . \tag{48}$$

To obtain the group velocity we still employ (3) and (4). The forms of $\underline{F}'$ and $\underline{\dot{F}}$ which result from (43), or (48), can be obtained by analogy with the way in which (5) is obtained from (2.10). In the present case, however, the analog of (5) will comprise the sum of six, sixth-order determinants when (43) is used to form the dispersion function, and the sum

of five, fifth-order determinants when (48) is used.

The efficiency of our simplified boundary conditions at depth was checked by direct numerical experiments. For the same dispersion calculations, both the simplified and the usual boundary conditions were used, and the different amounts of structure required in the two cases were noted. As would be expected the results showed that, to obtain a given accuracy in c, less structure had to be retained with the usual conditions than when terminating with free or rigid boundaries within the solid mantle. A complete set of tests, comparable to those illustrated in Figure 12, was performed for a period of 50 seconds. The results showed that when switching to the usual boundary conditions the increase in the maximum r_0 values was surprisingly independent of mode number and σ: (300±25) km, or in most cases, about 24 fewer integration steps when terminating with a homogeneous sphere. Our tests with mode 7 at 25 seconds showed the increase in r_0 to be approximately (100±10) km; about nine fewer integration steps. Thus, from our limited number of tests, it appears that $\delta r_0/(a - r_0)$ is roughly constant for a given mode and value of σ; r_0 is the maximum value allowed by the structural limitation when the condition of a rigid terminating boundary is applied, and δr_0 is the increase in r_0 when one employs the condition of a terminating homogeneous sphere. For example, the above ratio has about the value 0.14±0.02 for computations with mode 7 when 4 significant figures in c are required.

The results of these tests show the use of the usual terminating boundary conditions to be important only for the lowest (radial) mode numbers. For example, with mode 0 at 50 seconds, the required number of integration steps is improved about 42 percent by employing the usual boundary conditions, but this improvement drops rapidly with increasing mode number: to 18 percent by mode 4, and to 11 percent by mode 9. Relative to the main application intended for the high-speed algorithm developed in this report--computation of complete theoretical seismograms

containing all of the energy generated down to a period of 10 seconds--these tests lead to a very interesting result. When computing extremely large numbers of dispersion points for the first 90-100 radial modes--down to a minimum period of 10 seconds for each of these modes--careful comparison of the present results with the r_0 values of the 90-100 modes already being treated for complete, torsional-wave, theoretical seismograms, indicates an upper bound to the overall improvement of only 12 percent by switching to the usual terminating boundary conditions. We estimate the actual improvement to be, perhaps, half this value. If it is recalled that this "improvement" involves only the required number of integration steps, then the true improvement vanishes because the switch also requires the addition of the difficult evaluation of spherical Bessel functions of non-integral order number each time that the usual terminating boundary conditions are applied. Thus we conclude that relative to the main application intended for the results of our present work, no improvement can be expected by employing the usual boundary conditions at depth, and that the use of simplified terminating boundary conditions is fully justified.

In regard to the loss-of-precision problem, the above tests also yield comparative numerical results for the usual, and simplified boundary conditions at depth. After the switch to the usual conditions, the upper portion of the curve for each mode in Figures 12 and 13 is translated upward as described above, while the right-hand extreme of each curve occurs at the same depth as for the simplified boundary condition. As a result of the upward translation while the depth of this extreme remains fixed, homogeneous-sphere termination improves the maximum attainable accuracy for any given mode. However, this is only an advantage when one is limited to the original AJP formulation, and is dealing with relatively low (radial) mode numbers. For the problem of main interest at present, which requires results down to a period of 10 seconds for about 90-100

modes, this original formulation is unsatisfactory due to the limitation illustrated in Figures 14 and 15. The original formulation must be modified to obtain control over this loss-of-precision difficulty before all of these modes can be treated successfully down to this short a period. Hence the right-hand extremes in Figures 12 and 13 will not exist in the computed results from the final algorithm, which means that the last arguments in the preceding paragraph also apply here. Thus consideration of the loss-of-precision problem does not alter our previous conclusion: In the main application intended for the results of our present work, the use of simplified terminating boundary conditions is fully justified.

4. DESCRIPTION OF THE LOSS-OF-PRECISION PROBLEM

As stated in the main text, in the description of this problem there are four main points of interest: the numerical aspects of the difficulty, the interpretive aspects, the connection between loss of precision and numerical instability, and the possible methods for overcoming this precision loss.

Numerical description. Dispersion computations for spheroidal modes are a particular case of the numerical solution to a two-point boundary value problem. That is, in general the numerical methods applied to the AJP formulation represent the integration of three independent vectors, subject to boundary conditions at the top and at the bottom of the structure. One set of conditions will constrain the initial values of these vectors; the second will be used to form a "dispersion function," which is also called a secular or characteristic function. The dispersion calculation consists mainly of the determination of roots of this dispersion function. This is usually done in a trial and error procedure where one of two parameters is maintained fixed, and the second is varied systematically until a root is found. An algorithm for standing-wave computations will determine a natural period for a fixed polar order number

ℓ; in the case of propagating waves, a phase velocity will be found as a function of frequency. In the second section of the main text, we describe the results from our numerical experiments with loss of precision in the second type of determination. A simplified illustration of these results is given in Figure 16a. For a particular radial mode and fixed period, a maximum precision, σ_{max}, is obtained when using the optimum value $(r_0)_{optimum}$, in the numerical integration, where this is performed between the surface of the earth and the radius r_0. For several modes at a fixed period, the results show that when n increases, $(r_0)_{optimum}$ and σ_{max} decrease. Finally, σ_{max} decreases to the point where the dispersion computations become meaningless. The smoothness of the lower tail of each curve in Figure 16a has been emphasized. Actual results, as they are presented in Figures 12 and 13, show a more erratic behavior.

The situation when loss of precision completely dominates the dispersion computations is shown in Figure 17, where the dependence of the dispersion function on phase velocity and r_0 is illustrated. Solid lines represent the dispersion function when a numerical algorithm without any provision for avoiding the loss-of-precision problem is used to integrate

$$\overline{Y} = A\,Y \quad ; \tag{49}$$

dashed lines represent the same dispersion function when the algorithm has been modified to control this problem. The important thing to note is the almost completely random behavior of the dispersion function represented by the solid curve. The obvious conclusion here is that the calculation has lost all its significance. The connection between the erratic behavior of the lower end of the curves in Figures 12 and 13, and the random appearance of the solid curves in Figure 17 will be clarified later; however, it is obvious that an increasing loss of precision will increase the importance of the role played by the random fluctuations observed in each figure.

It is not difficult to understand from a physical point of view, why the accuracy attainable in dispersion computations increases as more

structure is included in the integration, i.e. as more and more of the specified structure is used in the actual computations. It is also reasonable from a numerical point of view to expect that, at some point in the addition of structure to the computational process, rounding and truncation errors will become important limitations to the attainable accuracy. That a picture that includes only these two factors--one physical and one numerical--may not be complete, becomes clear when it is noted how the computational errors also increase with mode number: the point at which the numerical factor becomes dominant over the physical one, occurring at smaller and smaller values of σ as n increases. We also know from the numerical results of the second and fourth sections of the main text that this problem exists independently of whether the integrations are performed downward from the top of the structure, or upward from a homogeneous sphere at depth.

To investigate this situation from another perspective, the angles between the three independent vectors were computed at different points of the structure as the integration was performed. This was done for modes where loss of precision was beginning to become, or already was, dominant. The results emphasize a persistent tendency of the three "independent" vectors to become more and more closely parallel as more and more structure is included in the integration; the parallelism becoming more marked as period decreases and/or radial mode number increases. The dispersion function is a determinant whose columns comprise components of these vectors evaluated at the terminating point of the integration. Therefore, when their parallelism becomes predominant, the value of the determinant will tend, spuriously, to be zero. Small, random errors in the numerical computations will then be increasingly important in this evaluation. Finally, the dispersion function will behave like a random variable with a null average value. This is the tendency which is observed in the solid lines of Figure 17, and which causes the erratic behavior of the lower

portions of the curves in Figures 12 and 13.

Interpretation of problem. A simplified, therefore qualitative interpretation of the above parallelism can be obtained if we model the earth by a sequence of thin layers. For our present purpose, in each layer the matrix $\underline{A}$ can be reduced to a constant by using suitable approximations to the actual depth dependences of the structural parameters. This makes it possible to avoid the complications that would be involved in the use of spherical Bessel functions, which would be introduced had it only been assumed that the properties of each layer were constant with depth. For each of these layers with constant matrix $\underline{A}$, a vector solution to (49) of the form

$$Y = Y_{oi}\, e^{p(r - r_{oi})} \tag{50}$$

is suggested by the first-order (scalar) case. Here, r_{oi} corresponds to the reference boundary of the layer. The introduction of this expression into (49) results in the eigenvalue problem for the matrix $\underline{A}$:

$$A\, Y_{oi} = p\, Y_{oi} \tag{51}$$

and, since $\underline{A}$ is not symmetric, in general we will have complex eigenvalues $\underline{p}$. Let us consider the sixth-order case from this point on. In most of its interior, the properties of the earth vary rather smoothly. This allows us to infer that the radial dependence of these eigenvalues will indicate just what is causing the numerical difficulties. To obtain this information a complete set of numerical tests was performed.

The results of these tests show the following general pattern. At the deepest points of the structure, three of the six eigenvalues have negative, and three have positive real values; at a smaller depth, two of these eigenvalues become complex conjugates; at still shallower depths, two more may become complex conjugates, but at least one real negative and one real positive eigenvalue will always remain. All real parts of complex eigenvalues are negative and equal at a given depth. The real eigenvalues can be arranged in pairs; each member of a given pair having about the same

magnitude as the other, but the opposite sign. Finally, these magnitudes are significantly greater than those of the real parts of the corresponding complex eigenvalues. One straightforward interpretation of these results is that, as a function of the radius, complex eigenvalues represent oscillations of the vector solutions, whereas real eigenvalues represent monotonically (exponentially) increasing or decreasing tendencies. Thus, toward the end of the integration of the three, initially independent vectors, a monotonically increasing tendency will predominate over the smaller oscillatory and decreasing functions. The vectors therefore become dependent, i.e. the three solutions are most strongly affected by a single tendency, the dominance of which increases rapidly as integration proceeds. This situation is independent of the direction in which one chooses to integrate: (1) The positive real eigenvalues will dominate in the exponential tendency when the integration is performed from below, i.e. when $r > r_{oi}$ in equation (50). (2) When the integration is done from the top downward--$r < r_{oi}$ in (50)--the negative real eigenvalues will take on this role.

The description above was motivated by similar discussions in the analysis of elementary vibration problems. One of these simple problems is sufficient to provide more insight into the occurrence of loss of precision. The analysis will yield a more quantitative understanding concerning the source of this numerical difficulty by dealing with a situation that is much simpler, but analytically similar to our dispersion computations. Consider the problem of finding the natural frequencies of a horizontal, homogeneous beam that is rigidly supported at its two ends. In this case, the dispersion function for the bending of the beam in the vertical plane is

$$F = \Omega^2[(\sinh \Upsilon + \sin \Upsilon)^2/4\Upsilon^2 - (\sinh \Upsilon - \sin \Upsilon)^2/4\Upsilon^2] \quad , \quad (52)$$

where Ω is the length of the beam, Υ is proportional to $\Omega\omega^{1/2}$, and ω is the natural frequency being sought (Pestel and Leckie, 1963, page 192). In

this simple problem, it is possible to note directly that (52) reduces to

$$F = \frac{\Omega^2}{\gamma^2} \sinh \gamma \sin \gamma \quad . \tag{53}$$

However, this reduction would not be immediately apparent in more complicated problems, and an expression similar to (52) would be used in the numerical evaluation of the dispersion function since this expression arises naturally from the matrix formulation of the problem. From (53) it is obvious that roots of the dispersion function occur at

$$\gamma_k = k\pi \quad , \quad k = 1,2,3, \cdots \quad , \tag{54}$$

where k is the mode number. However, if the dispersion function is evaluated using expression (52), beyond a certain magnitude of γ these roots will not be found due to loss of precision. This point is reached when γ attains a magnitude such that, for the number of digits carried in the calculations,

$$\sinh \gamma \pm \sin \gamma = \sinh \gamma \quad . \tag{55}$$

Another interesting feature of these computations occurs as γ increases, but before it has gotten large enough to produce complete dominance of loss of precision. In the evaluation of the dispersion function, the values of $\sin \gamma$ will be magnified by the factor $\sinh \gamma$, and roots of F will be bracketed by increasingly steep portions of this function. The situation is illustrated in Figure 18, which represents the first three roots of F/Ω^2. One can see that the dispersion function, after the first root, begins to oscillate with rapidly increasing amplitude as γ increases. Therefore, the slope also increases rapidly in the neighborhood of successive roots. Actual computations of this function, continued to higher frequencies, show that the maximum amplitude of each oscillation increases by about an order of magnitude for each successive mode. From this one concludes that, when refining a root to a given accuracy after bracketing it with a change of sign of F, one must stress the number of stabilized figures in the natural frequency, and not the corresponding

values of the dispersion function; the point being that for large γ, F can take on exceedingly large magnitudes even though the trial frequency agrees, to several significant figures, with that at a root. The validity of this conclusion is unaffected by the consideration that γ is proportional to the square root of the frequency. In the third section of the main text it is demonstrated that the behavior of the dispersion function for this simple vibration problem is very similar to the behavior of the corresponding function for spheroidal-wave computations.

This simple example again illustrates that the explanation of most of the numerical difficulties in dispersion computations can be found by considering the relative values of exponential (hyperbolic) and oscillatory (circular) functions when both appear at the same time in one step, or in a consecutive set of steps, of the numerical procedure. Consider, for example, the case of surface waves on a flat, multilayered structure when the Thomson(1950)-Haskell(1953) technique is applied to obtain the dispersive properties. For Love waves, either a hyperbolic or trigonometric function appears at each particular step, never both at the same time, and loss of precision never develops. On the other hand, in the corresponding formulation for Rayleigh waves the transfer matrix for a given layer may contain both kinds of functions, and we do indeed find the expected numerical problems in this case. (See Schwab and Knopoff, 1970, for specific numerical details.) Transformations analogous to that which produces (53) from (52) must be introduced to remove this loss-of-precision problem. The first of these alternatives to the original Thomson-Haskell formulation was Knopoff's (1964) method; the optimization, analysis, and loss-of-precision testing of this method are given by Schwab (1970). The second alternative--delta-matrix extensions of the original formulation (Thrower, 1965; Dunkin, 1965; Watson, 1970)--is presented in optimized form by Schwab and Knopoff (1970), and is compared in detail with Knopoff's method by Schwab (1970). Recent, extensive testing of the

loss-of-precision features of both of these alternatives is described by Schwab, Nakanishi and Liang (1980).

Connection between loss of precision and instability. The expression "loss of precision" is particularly descriptive of the type of numerical difficulty which is our main concern in this study. The connection of this term with the rather general expression "inherent instability" (see, for example, Scott, 1973, page 37) should be made clear. To summarize, we are seeking the solution to a two-point boundary value problem by using an iterative method, where each iteration consists of carrying out three independent integrations of the AJP system of differential equations. Each individual integration is considered now as an initial-value problem. At the beginning of each iteration, a continuously improved estimate of the dispersion result is introduced into the numerical scheme; the independent vectors are integrated over r; and the satisfaction of a boundary condition is then tested. This test consists of computing the value of a determinant, and checking to see if it is converging toward a root. Each column of elements of the determinant comprises components of one of the vectors. The vectors may lose their independence when dominant, large real eigenvalues of the matrix $\underline{A}$ are present for consecutive steps of the integration, and this loss of independence will yield a determinant with linearly dependent columns. Thus, as more independence is lost, more loss of precision enters into the determination of the dispersion.

When integrating upwards, real positive eigenvalues are a particular condition for instability since exponentially increasing solutions will tend to be unbounded as the integration proceeds; real negative eigenvalues play this role when integrating downwards. This instability cannot be eliminated, in an absolute sense, by any linear transformation applied to the three vectors (Gantmacher, 1959, Chapter XIV, Section 2). Therefore, this instability must be considered intrinsic to the system of differential equations. Hence it is termed an "inherent" instability. In our specific

problem, we have unbounded increase in magnitudes both relative to other forms of solutions, and relative to the maximum value that the computer can accept. In the former case, this increase results in loss of precision, in the latter, overflow; inherent instability is the basis of both these difficulties.

A final point of interest here concerns the dependence of the loss-of-precision problem on the direction of integration. From preceding arguments, loss of precision can be traced back to the existence of real eigenvalues with large magnitudes. Were the complex eigenvalues (with negative real parts) to produce the predominant solutions at shallow depths, then integration from below would appear to be a way of avoiding loss of precision. This is not the case. As already indicated, both types of integration, either from below or above, will ultimately produce loss of precision due to the dominance of one exponentially increasing tendency. Numerical results from both the second and third sections of the main text amply confirm this conclusion.

Methods for overcoming loss of precision. The simplest possible solution for overcoming this inconvenience is to increase the number of digits used in the computations. Thus the alternative of using the extended-precision capabilities of modern computer software is appealing; however, one encounters the obvious problems here: Calculations become extremely expensive, or sufficiently high-precision capability is unavailable. For practical, computational purposes, the best approach would therefore appear to be to devise schemes that prevent the spurious parallelism of the vectors from developing.

Three suggestions are found in the literature, which are pertinent to direct integration of the AJP system of equations. One is to improve the conditioning of the matrix $\underline{A}$ (Wiggins, 1968) by introducing a linear transformation in the depth-dependent components of $\underline{Y}$, i.e. in $\underline{y}_i(r)$ (see second section of main text), and a non-linear transformation of the radius

$\underline{r}$. Chapman and Phinney (1972) have analyzed this approach, and conclude that while it does improve things by producing a better balanced system of equations from a numerical point of view, it does not influence the fundamental difficulty.

We are not limited, of course, to only linear transformations. A useful non-linear operation involves the application of delta matrices (Pestel and Leckie, 1963) or, what are equivalent, compound matrices (Gantmacher, 1959, Chapter I, Section 4). When this approach is applied to spheroidal-wave computations which include the effect of gravity, the three original vectors are arranged as the columns of a 6×3 matrix. The delta form of this matrix is obtained by forming the minors of order three, and arranging them as a 20×1 vector $\underline{y}$. Taking derivatives and making use of the original system of differential equations, we obtain a new system

$$\dot{\underline{y}} = \underline{E}\,\underline{y} \quad , \tag{56}$$

where the 20 elements of $\underline{y}$ are the new dependent variables, and $\underline{E}$ is a 20×20 matrix. The problem is therefore transformed into one in which only a single vector needs to be integrated. However, the 400 elements of $\underline{E}$ must be derived analytically, and it must then be shown numerically that this new formulation does indeed eliminate loss of precision. Discussions of the delta-matrix approach to the computation of dispersion for spheroidal waves, for the simpler, non-gravitating case, are given by Gilbert and Backus (1966; 1969) and Takeuchi and Saito (1972). However, the literature does not appear to contain any serious attempt to justify and apply the 20th-order technique that describes the spherical, gravitating case. In the fourth section of the main text, a method for simplifying the sixth-order, AJP formulation is developed, which makes it possible to use the delta-matrix approach for gravitating structures without requiring the use of 20th-order matrices.

The third suggestion for treating loss of precision is much simpler, although perhaps not so basic as delta matrices. This is to orthogonalize

the three vectors after every few integration steps, and thereby to maintain the independence of these solutions. The earliest references to this approach, in the geophysical literature, seem to be Pitteway (1965) and Neigauz and Shkadinskaya (1972) (see also Chapman and Phinney, 1972; Nolet, 1976). The purpose of the third section of the main text is to report the numerical results of our introduction of orthogonalization into computations for both propagating Rayleigh waves and standing spheroidal modes on a spherical, gravitating earth. This technique makes successful computations possible for short periods and/or high radial order numbers. Our results concerning orthogonalization appear to be the first that report successful treatment of the loss-of-precision problem for a gravitating earth.

As our interest in this report is in direct methods of integration of the AJP formulation, no discussion of the variational technique will be given. For a recent application of this approach we again refer the reader to Wiggins (1976).

5. COMPUTATIONS FOR RAYLEIGH WAVES AND SPHEROIDAL MODES

To provide as complete a test as possible of orthogonalization as a solution for precision loss, separate programs for propagating surface waves and for free-mode oscillations were employed. Brief descriptions of these routines follow, with the results--other than those connected with orthogonalization--that are pertinent to short-period, large-n computations.

Propagating Rayleigh waves. Our algorithm is based on that which is fully described in the second section of the main text. To generate dispersion information at equal frequency intervals, for use in the usual technique for generating time series by inverse Fourier transformation, computations are carried out at fixed frequencies to obtain the corresponding phase velocities or values of ℓ. Integration is performed

from the top downward, with the three independent vectors, $\underline{X}_i$, satisfying the boundary conditions at the free surface exactly. The integration is then carried down to a sufficient depth to make it immaterial--to the specified accuracy in $\underline{c}$ or $\underline{\ell}$--just how we terminate the integration; with a free or rigid boundary, for example. Almost all of the integration is performed using a fourth-order, predictor-corrector method, with everything possible being done to minimize the time required for each evaluation of the dispersion function. There are two main features in this optimization. The first is the specification of the required accuracy in the computed phase velocities to be as low as possible, which allows us to use the maximum possible step sizes in the numerical integration, thus minimizing the required number of integration steps and expense of computation. To decide upon the accuracy necessary to synthesize relatively short-period, complete theoretical seismograms for spheroidal waves, reference to the analogous work with torsional waves (Liao et al., 1978) indicates that 4-figure accuracy is quite sufficient. The second, and most significant feature in the optimization is the application of our knowledge about this particular problem to specify all of the depths at which the elements of $\underline{A}$ are to be evaluated. These evaluations can then be removed from the innermost, integration loops of the program. In fact, most of the procedure for evaluating these elements can even be removed from within the frequency and phase velocity loops.

In the work described in the second and fourth sections of the main text, extremely large values of $\underline{n}$ could not be used at short periods since a feature for controlling loss of precision had not yet been included in the computational algorithms. In those algorithms, a maximum integration step size of 12.5 km was found sufficient to yield the desired accuracy in computed phase velocities. It is reasonable to expect that at the shortest periods of interest here, when large values of the radial order number are treated, this maximum step size will have to be decreased at some point.

That is, as these short-period integrations are carried down to greater and greater depths, the natural error growth will have to be offset by a decreased step size. With orthogonalization included in the computations it is possible to determine just when the steps must be decreased. Although detailed analysis of this question is beyond the scope of the present study, our results do provide a few estimates that establish guidelines for future work: (1) A maximum step size of 12.5 km can be used for all periods down to at least 25 seconds. (2) Somewhere between 25 and 10 seconds, it will be necessary to begin decreasing this maximum step size; the value of n at which this decrease is to be effected, decreases with decreasing period. (3) At 10 seconds, this maximum must be decreased at about $n = 6$--the precise value of n probably being dependent upon the specified structure; if this decrease is to 6.25 km, this will suffice until n reaches about 27; one additional decrease should be sufficient to complete the computations at this period.

Standing spheroidal modes. Our algorithm here corresponds to the techniques devised mainly by Backus and Gilbert (1967) and Gilbert and Backus (1969) (see also Takeuchi and Saito, 1972). The computations are carried out at fixed, integral polar order numbers ℓ, and the dispersion results are the corresponding natural frequencies. Integration is performed from the bottom of the structural segment, $r = r_0$, upward toward the free surface where the exact boundary conditions are applied to form the dispersion function. The initial values of the three independent vectors, $\underline{Y}_i(r_0)$, are obtained from the three independent solutions of the equations of motion for a homogeneous, gravitating sphere below r_0. A discussion of the two different ways of handling the boundary conditions at r_0 is contained in the second section of the main text, and in the additional notes (Section 3). The integration is performed with a scheme that permits the use of Runge-Kutta methods of variable order (Shanks, 1966). Essentially, this affords a means of maintaining a fixed accuracy

in the computed dispersion results for different values of n and ℓ, i.e. the order of the Runge-Kutta method, and hence the number of evaluations of the elements of A, is automatically increased as radial and polar order numbers increase. This, of course, is equivalent to decreasing the step size of the fixed, fourth-order predictor-corrector method employed in the Rayleigh-wave algorithm.

Other than for use in computing the acceleration due to gravity, $g(r)$, the core is not of great interest in our present Rayleigh-wave computations. For the spheroidal-mode algorithm it is important, and orthogonalization has been added to our integrations both above and below the mantle-core boundary. With the core included in the integrations, a third possibility for the boundary condition at r_0 is a power series expansion in r (Crossley, 1975). In this last reference, a numerical comparison will be found between this expansion and the rigid-boundary approximation at r_0.

With orthogonalization included in our algorithms, loss of precision is no longer a problem and ℓ can be increased to the (short-period) point where the depth of penetration becomes dependent on the shallower structural features. Since these features can vary strongly from model to model, when dealing with shallow penetration some provision must be made for modifying the value of r_0 determined by the usual empirical rules (see, for example, Bolt and Dorman, 1961, page 2967). When integrating downward, as in our Rayleigh-wave algorithm, it is possible to determine the optimum value of r_0 right as the computations proceed (see Section 6 of the additional notes for details), since this is the point at which the displacement-depth functions minimize and begin to increase without bound as depth increases. When integrating upward in our spheroidal-mode algorithm, r_0 must be specified a priori. Thus a method must be included in this algorithm to correct for values that are initially too small, which result in wasted computation time and expense, or values that are initially

too large, which may decrease the accuracy below that desired. For this purpose, after the first iteration over period at fixed n and ℓ, the depth dependence of the vertical component of displacement, $y_1(r)$, is computed and normalized to unity at the free surface. The location of r_0 is then adjusted upwards or downwards depending upon whether $|y_1|$ at the original location is less than, or greater than 10^{-6}.

General results. By using both of the above algorithms to test orthogonalization as a remedy for precision loss, we have a rather satisfying range of physical and numerical approaches over which this technique will be shown effective: computations based on free-mode oscillations and surface-wave propagation, with integral ℓ values being used to obtain standing-wave periods, but with the phase velocities for propagating waves corresponding to non-integral ℓ; physical conditions at r_0 approximated by free or rigid boundaries, or by three independent solutions for a homogeneous, gravitating sphere extending from the center of the earth to $r = r_0$; numerical integration carried out both upward and downward, either based entirely on Runge-Kutta methods, or mainly on a predictor-corrector technique; and computations with uniform, relatively high accuracy required in the dispersion results, contrasted with calculations in which the main concern is computational speed, with, for this purpose, accuracy held to the minimum acceptable value. Before describing--in the third section of the main text--the results of numerical tests that deal specifically with the orthogonalization procedure, a few general results should be covered.

A point of considerable practical interest in these algorithms is the question of overflow and its control. The computational quantities in which we are interested are the independent vectors X_i, which we are integrating over depth. These vectors are not highly oscillatory as a function of r--as is the linear combination of them that yields $Y(r)$ (second section of main text, equation (2.8))--hence the overflow

properties are easily determined by monitoring the increase in magnitude of each of the elements $[X_i(r)]_j$ as integration proceeds. We first note that these magnitudes increase during integration, independent of the direction of this process: When integrating upward, $|[X_i(r)]_j|$ increases with increasing r; when integrating downward, these magnitudes increase as r decreases. Thus the relative overflow characteristics are obtained by monitoring, and comparing the increase of $|[X_i(r_0)]_j/[X_i(a)]_j|$ for downward integration, to the increase of the reciprocal of this quantity for upward integration, where, as before, a is the radius of the earth. The results of our numerical experiments are given in Figure 19. The results for downward integration are represented by continuous lines:

$|[X_1(a - \text{depth})]_1/[X_1(a)]_1|$ long dashes

$|[X_2(a - \text{depth})]_3/[X_2(a)]_3|$ short dashes

$|[X_3(a - \text{depth})]_5/[X_3(a)]_5|$, $|[X_3(a - \text{depth})]_6/[X_3(a)]_6|$ dotted lines,

since these are the only ratios with non-vanishing denominators (second section of main text, equations (2.6) and (2.7)) when proceeding downward from the free surface. The corresponding ratios for upward integration, $|[X_i(a)]_j/[X_i(a - \text{depth})]_j|$, yield 18 results for each value of the depth since none of the denominators vanish; hence, to simplify, the range of solutions at each depth is represented by a bar, and the continuous solution range is then denoted by hatched areas. The agreement of overflow features for upward and downward integration is quite striking. The features are precisely the same in form and magnitude. The point at which overflow will occur is therefore only a function of r_0 and the maximum magnitude accepted by the computer, and is independent of the direction of integration. Thus the following remarks on control of overflow apply equally well to each of our algorithms.

The powerful normalization scheme that is described in the second section of the main text was designed mainly for use in program testing,

where it is desirable to have the capability to perform dispersion computations while retaining an extremely large number of wavelengths of structure. In routine applications, the integrations should be performed with only the minimum amount of structure required to yield the desired accuracy. Application of this restriction to the low radial order numbers treated in the second section of the main text makes it possible to avoid overflow in these cases without employing any form of normalization; for the large n values that orthogonalization now makes it possible to handle, overflow occurs even when using only the required minimum of structure. However, the problem is not so severe that its control necessitates as powerful and expensive a method as that described in the second section of the main text. The following procedure, the cost of which is trivial, has been found sufficient for radial mode numbers up to 90-100, and ℓ values corresponding to periods down to 10 seconds.

If the minimum magnitude that is accepted by the computer is

$$\text{MIN} = 10^{-m} \quad , \quad m > 0 \quad , \tag{57}$$

then at the starting point of integration, the non-zero elements of each vector $\underline{X}_i$ are conservatively scaled such that their minimum magnitudes are set at about $10^{-(m - 5)}$. The three vectors are then integrated to the final value of r with no normalizations during these operations. At the terminal point of integration, r_0, to ensure that overflow is avoided while forming the dispersion function, all elements are multiplied by

$$10^{-(m_1 + m_2)/2} \tag{58}$$

before evaluating the determinant. The quantities m_1 and m_2 describe the magnitude bounds of the 18 vector elements:

$$\begin{aligned} \text{maximum } |[X_i(r_0)]_j| &= 10^{m_1} \\ \text{minimum } |[X_i(r_0)]_j| &= 10^{m_2} \end{aligned} \tag{59}$$

The scaling factor at r_0, (58), is determined during the first evaluation

of the dispersion function. This same value is used in subsequent evaluations. By so doing, as well as a meaningful sign of the dispersion function, we will have magnitudes that vary smoothly from iteration to iteration, i.e. as a function of the dispersion variable.

The last of our general results is really a note of caution concerning root refinement in the presence of precision loss. When determining roots of the dispersion function in the most conservative manner, we usually increase the dispersion variable systematically until this function changes sign. The root is then refined to obtain a more accurate value of the dispersion variable, i.e. a value closer to the root of the determinant. However, in the presence of precision loss we have the behavior illustrated by the solid lines in Figure 17, and if root refinement is carried out with too simple a procedure we can be led into mistaking spurious stabilization for true accuracy. An example of such an oversimplified procedure is repeated bisection of the dispersion-variable interval, where only the half interval defining a sign change in the dispersion function is retained after each bisection. The error here is in not monitoring the magnitude of successive values of this function. The random behavior, with no convergence toward zero, is an obvious signal indicating precision problems. At the qualitative level, numerical testing for the existence of precision loss in dispersion computations is described by Schwab, Nakanishi and Liang (1980). The more expensive, quantitative analysis is described in the second section of the main text, and is illustrated in Figures 11-15.

6. NUMERICAL TESTING OF FOURTH-ORDER, HIGH-SPEED PROCEDURE

The subroutine structure used in the testing of the fourth-order, high-speed procedure is given in Figure 27. For each new frequency, the eigenfunctions $\underline{y}_1(r)$ and $\underline{y}_3(r)$ and the polar order numbers $\underline{\ell}$, at the three previous frequencies, are used as input to XINTEG. There, predictions

$py_1(r)$, $py_3(r)$, and ℓ are computed for the current frequency, and $I_i(r)$ are calculated in a very compact procedure using simple, trapezoidal integration. Even though $I_i(r)$ are ℓ dependent, the accuracies of the predicted $py_1(r)$, $py_3(r)$, and ℓ are good enough--and the dependence of the final ℓ value on the accuracy of $I_i(r)$ small enough--so that $I_i(r)$ need not be recomputed each time ℓ is varied to decrease the magnitude of the dispersion function in the central subroutine block of Figure 27. When ℓ, or c, has been obtained to the desired accuracy, the program prepares to treat the next frequency by obtaining the eigenfunctions at the present ω. The corresponding subroutine block--the last in Figure 27--requires only a single starting vector. With the solution for ℓ at the present frequency, we obtain

$$\frac{y_3(a)}{y_1(a)} = -\frac{[X_1(r_0)]_1}{[X_2(r_0)]_1} \tag{60}$$

for rigid-boundary termination. Integration in EIGSAC and EIGMTL is then performed with starting vector $X_E(a)$:

$$\begin{bmatrix} y_1(a) \\ y_2(a) \\ y_3(a) \\ y_4(a) \end{bmatrix} = y_1(a) \begin{bmatrix} 1 \\ 0 \\ y_3(a)/y_1(a) \\ 0 \end{bmatrix} = y_1(a)\, X_E(a) \tag{61}$$

to determine the eigenfuctions $y_i(r)/y_1(a)$. Relations (60) and (61) apply to a continental structure. For an oceanic model, the left-hand side of (60) becomes

$$\frac{y_3(r_1)}{y_1(a)} \quad , \tag{60a}$$

and integration is performed in EIGSEA starting with $Z(a)$ as in (4.31). Since (60a) is now specified, for an oceanic structure integration is continued in the solid mantle with (4.32) modified to start with the single vector $X_E(r_1)$:

$$\begin{bmatrix} y_1(r_1) \\ y_2(r_1) \\ y_3(r_1) \\ y_4(r_1) \end{bmatrix} = y_1(a) \begin{bmatrix} [Z(r_1)]_1 \\ [Z(r_1)]_2 \\ y_3(r_1)/y_1(a) \\ 0 \end{bmatrix} = y_1(a)\ X_E(r_1) \quad . \qquad (61a)$$

In the subroutines of the two lower blocks in Figure 27, we use the numerical integration procedures described in the second section of the main text (Numerical Technique for Integrating the System of Differential Equations), and in Section 2 of the additional notes. Also as described in the second section of the main text, the key to optimizing the numerical integration is to apply our knowledge about this specific problem to specify all the depths at which $\underline{a}_{ij}(r)$, $\underline{A}_{ij}(r)$, $\underline{b}_{ij}(r)$, $\underline{B}_{ij}(r)$ are to be evaluated, which then makes it possible to handle all of these evaluations outside of the lower two subroutine blocks in Figure 27. Comparison of Figure 1 with (4.21), and Figure 2 with (4.23) and (4.24) will indicate the changes needed to include the modified elements, $\underline{A}_{ij}(r)$ and $\underline{B}_{ij}(r)$, of the high-speed procedure. (See Table 7 for a short-period, high mode number modification of the integration depths as given in the second section of the main text.)

The numerical tests are directed toward eventual use of the high-speed procedure to compute complete theoretical seismograms, for the spheroidal-wave excitation at the receiver, that can be compared directly with the experimental records from the long-period instruments of the WWSSN. All body-wave and surface-wave energy down to a period of 10 seconds is to be contained on the synthetic seismograms, which are obtained from standard inverse Fourier transformation of the multimode, frequency-domain information. This technique is described by Kausel and Schwab (1973) and Schwab and Kausel (1976) for torsional-wave seismograms. The resulting applications, at various stages of the development of the method, are given by Knopoff, Schwab and Kausel (1973), Knopoff et al. (1974), Nakanishi, Schwab and Kausel (1976), Nakanishi, Schwab and Knopoff

(1976), Mantovani et al. (1976), Kausel, Schwab and Mantovani (1977), Mantovani et al. (1977), Mantovani (1978), and by Liao et al. (1977; 1978). Specifically, in this application the key computation is that of the phase velocity, or ℓ, down to 10 seconds, for every (radial) mode that is excited at periods above 10 seconds; all this dispersion information is to be obtained in a single computer run. The present tests are limited to penetration of energy to no deeper than the mantle-core boundary, although penetration into the core is covered in the Conclusions of the fourth section in the main text. (The core, of course, is included in the computation of the gravitational acceleration, $g(r)$.) In the tests, the radial modes are treated in turn, starting with the fundamental. We use a frequency step size of 0.0005 cps, which corresponds to $\Delta\ell \approx 5$. For each mode, a "starting" procedure (see Section 8 of the additional notes) must be applied to obtain ℓ and the eigenfunctions at the three longest periods. For this purpose, in these numerical tests we have used the sixth-order algorithm described in the second section of the main text. The high-speed procedure is then applied to each successive frequency until the minimum period of 10 seconds is reached. At each frequency, computation begins with an accurate estimate of ℓ, or c, and the central subroutine block of Figure 27 is then executed, to compute the dispersion function, four times: The first estimate of c to be used in this subroutine block is obtained by extrapolation from the phase velocities at the three previous frequencies; this initial estimate is increased 0.005 km/sec to compute the second value of the dispersion function; the third c to be used is that yielding a zero of the linear fit to the dispersion function from the first two c values; the fourth c used is that at the zero of the quadratic fit from the first three values. The final phase velocity is that at which the dispersion function vanishes when it is represented by the cubic which is defined by the above four c values.

Since the initial estimate for ℓ, or c, is quite accurate, this amount

of root refinement may appear excessive to obtain the four significant figures we desire; however, two further considerations lead to the requirement of stability past the fourth significant figure of $\underline{k}$, although with the integration step sizes and fourth-order Runge-Kutta and predictor-corrector methods used here, this stability does not represent absolute accuracy. First, the eigenfunction routines need more than 4-figure stability in $\underline{k}$--both for successful extrapolation, from frequency to frequency, of $\underline{y_1}(\underline{r})$ and $\underline{y_3}(\underline{r})$, and to keep structure reduction (see following paragraph) from becoming excessive--even though this extra stability begins to define roots of the dispersion function in terms of algorithmic limitations as well as physical structure of the earth. Secondly, the analysis of synthetic time-series computation (Figure 1, Calcagnile et al., 1976) demonstrates that $\underline{k}$, or $\underline{c}$, should be stable (smooth) to at least the fifth figure to avoid unnecessary noise on the computed seismograms. The results of Liao et al. (1977; 1978) indicate that smoothness in the fifth, or later figures is sufficient for this purpose, and that more than four figures of absolute accuracy is unnecessary. It should be realized (Figures 20 and 21, Schwab and Knopoff, 1972; or Figures 2 and 3, Schwab and Knopoff, 1971) that it is meaningless to attempt higher absolute accuracies when dealing with perfectly-elastic models of the earth. If the intrinsic attenuation of the earth is introduced in an exact manner (same two references), this causes modification in the fourth and fifth figures of the phase velocities obtained from perfectly-elastic models.

Concerning the structure reduction mentioned above, this is effected at each frequency upon exiting EIGMTL in Figure 27. At this stage in the computations $\underline{y_i}(\underline{r})$ are in storage at the last (greatest) seven depths to which the integration has been carried. A check is therefore made to see whether $|\underline{y_1}(\underline{r})|$ has ceased to decrease with depth for one of these values of $\underline{r}$. If it has, this value of $\underline{r}$ is used as $\underline{r_0}$, the point at which

integration is to be terminated at the next higher frequency. This automatic structure reduction is quite important in minimizing computation time, preventing overflow, and in avoiding premature difficulties with the loss-of-precision problem.

Since we are interested here only in establishing the validity of the high-speed procedure, and since we know from the third section of the main text that orthogonalization can be added later to control loss of precision successfully, it will be sufficient to limit the numerical testing of the procedure to the algorithm described by (4.20) and (4.22), i.e. before any feature for controlling precision loss has been included. For four significant figures in computed values of $\underline{c}$, reference to Figure 15 will show that there is a sufficient number of modes for which it will be possible to test the high-speed procedure all the way down to a period of 10 seconds, and thus, to establish the validity of this procedure. Test details and ranges for the three modes used in these tests are given in Table 5. Loss-of-precision difficulties just begin to appear when the shortest periods of the third mode are treated. Attainment of the desired, 4-figure accuracy in computed values of $\underline{c}$ was checked by duplicate calculations with the sixth-order algorithm. This accuracy was retained by the high-speed procedure until structure reduction placed $\underline{r_0}$ within the low-velocity zone in the upper mantle. However, even in the worst case--the fundamental mode when $\underline{r_0}$ is above the axis of the channel, i.e. above a depth of 136 km--the accuracy falls only to slightly below 4 significant figures. Rather clearly, this decrease in accuracy when only a small number of integration steps is retained, is caused by the use of simple, trapezoidal integration to form $\underline{I_i}(\underline{r})$ in subroutine XINTEG. Just as clearly, however, there is no need to modify this integration method. For this shallow penetration, gravitation can be partially dropped from the computations, so that when $\underline{r_0}$ increases beyond a certain point, we switch to a much simpler approximation that does not require the use of XINTEG.

Thus 4-figure accuracy is maintained down to the shortest period treated.

When $\underline{r}_0$ increases beyond the position of the "400-km discontinuity," $\underline{y}_5$ and $\underline{y}_6$ are set equal to zero in the formulation although $\underline{g}(\underline{r})$ is retained in the elements of the coefficient matrix. The subroutine structure of Figure 27 can then be used without XINTEG or the last block of eigenfunction routines. Check computations with the sixth-order algorithm show this new procedure to remove the accuracy problem with short-period crustal waves. Checks against values given by North and Dziewonski (1975) for phase velocity differences between gravitating and non-gravitating structures, indicate that by leaving $\underline{g}(\underline{r})$ in the new algorithm, these phase velocity differences are decreased: perhaps 20 percent for the fundamental mode at a period of 300 seconds. At any rate, the accuracy checks with the sixth-order program indicate that the approximation is fully satisfactory as soon as $\underline{r}_0$ has increased beyond the "400-km discontinuity," into the region of the low-velocity channel in the upper mantle.

This is quite an important result since it means that the most difficult region in which to perform dispersion computations--that where the members of the low-velocity-zone, channel-wave family cross the members of the crustal-wave family (pages 908-910, Schwab and Knopoff, 1971; pages 166-168, Schwab and Knopoff, 1972; Panza, Schwab and Knopoff, 1972)--need not be treated with the complication of gravity. Hence, we need not apply the extrapolation technique for predictions $\underline{py}_1(\omega)$ and $\underline{py}_3(\omega)$ for these period ranges in which the energy associated with a given radial mode is abruptly shifting up and down, between the low-velocity zone and the crust, as period varies. This becomes a more and more important consideration as the minimum period to which the computations are carried becomes shorter and shorter.

7. DETAILED TIMING INFORMATION

Because of the importance and expense of the type of computations

described in this report, more detailed timing information is given here for those requiring such data. The simplified information given in the main text, however, should be sufficient for most purposes.

Before presenting the final timing details, a few words of explanation should be given concerning optimization of the subroutine structure given in Figure 27. After the numerical testing of modes 0-2, a final attempt was made to further improve the optimization. To be specific, the treatment of HPCMTL will be described; the analogous handling of the other subroutines will be obvious. The initial form of the predictor-corrector portion of HPCMTL was that of Figure 4c, with the obvious variables switched from the dimension statement into a common block, and with A(1430) increased to A(3900). The computation time was found to depend significantly upon the arrangement of the variables in the common block. To completely eliminate this difficulty, the subroutine structure was abandoned and HPCMTL was included as a program section in the main routine. The logical IF-statements within the inner nested DO-loops, e.g. loop 170 which occurs within a loop of the form 180 (Figure 4a), require an inordinate amount of time and should be deleted. This leads to loop 170 being replaced by two loops, but this is justified by the improvement in computation time. The above modifications lead to an improvement from 143×10^{-6}, to 131×10^{-6} sec/step/iteration in the characteristic time for treating solid structural units with the predictor-corrector method described in the second section of the main text. Of course a like improvement would be expected in the characteristic time of the comparable sixth-order algorithm: 266×10^{-6} sec/step/iteration (second section of the main text, Optimization of the AJP Formulation). Since three, rather than two vectors are integrated in this algorithm, the estimated improvement would be to 250×10^{-6} sec/step/iteration.

The formulas for detailed computation of timing estimates are given in Table 6. Table 7 contains the values of the parameters used in Table 6.

These tables should be studied in combination with the program-section structures in Figures 27 and 28, data given in the second section (Numerical Technique for Integrating the System of Differential Equations; Figures 6 and 8) and third section of the main text (maximum values of integration step size in the mantle), and the comparative computer speeds given by Porter et al. (1980). In Tables 6 and 7, NSED is the number of sedimentary layers, NCRUST is the number of subsedimentary crustal layers, $(n_i)_j$ is the number of integration steps in the ith part of the structure using the jth integration method: Runge-Kutta (RK) or predictor-corrector (PC) (second section of the main text, Numerical Technique for Integrating the System of Differential Equations), and $(\tau_i)_j$ and $(\varepsilon_i)_j$ are the characteristic times corresponding to $(n_i)_j$. The times $(\tau_{SOLID})_j$ are about double $(\varepsilon_{SOLID})_j$ because the former characterize the treatment of two vectors $\underline{X}_i(r)$, while the latter describe the handling of only a single vector, plus of course, the storage of $\underline{y}_1(r)$ and $\underline{y}_3(r)$. This storage is the only difference between the program segments characterized by $(\tau_{SEA})_j$ and $(\varepsilon_{SEA})_j$. Part of the reason for the Runge-Kutta characteristic times being so much larger than the predictor-corrector times is that a refinement technique (equation (23), Ralston, 1960) is applied to $\underline{y}_i(r)$ and $\underline{\bar{y}}_i(r)$ after the Runge-Kutta procedure, and is included in these characteristic times. The numbers of iterations over ℓ to obtain the required accuracy in $\ell(\omega)$: $[(N_\ell)_{4th}]_{HS}$ for the fourth-order, high-speed procedure, and $[(N_\ell)_{4th}]_{START}$ for the fourth-order, "starting" procedure, are 4 and 5-6 respectively. The latter estimate is probably safe to use, but must still be verified with numerical tests of this "starting" procedure. Assume that τ_{XINTEG} and τ_{YINTEG}, and hence T_{XINTEG} and T_{YINTEG}, are approximately the same.

The "effective" characteristic times specified in the main text for the high-speed procedures--164×10^{-6} sec/step/iteration for a gravitating structure, and 132×10^{-6} sec/step/iteration for the approximation with

vanishing $\underline{y_5}$ and $\underline{y_6}$, but with $\underline{g(r)}$ left in the expressions for the elements of the coefficient matrix--are simple averages obtained from the values in Table 8. Each of the characteristic times given in that table was obtained by equating

$$\tau_{eff} \times [(N_\ell)_{4th}]_{HS} \times \text{(average number of integration steps)}$$

and the full expression from Table 6, with the above "average number of integration steps" being approximated by the expression in footnote 2 of Table 7, where $(n_{MTL})_{PC}$ is replaced by an average value that is just half of the value indicated in Table 7. The continental structure was specified with two sedimentary layers, and granitic and basaltic subsedimentary crustal layers. For the "non-gravitating" structures, program sections XINTEG, EIGSEA, EIGSAC, and EIGMTL were ignored in calculating τ_{eff}. By analogy with the above estimates of τ_{eff}, the corresponding "effective" characteristic time for the sixth-order algorithm should increase from the predictor-corrector time (alone) of 250×10^{-6}, to about 252×10^{-6} sec/step/iteration.

When the fourth-order, high-speed procedure, (4.20), is put in its delta-matrix form, (4.38), the resulting, "effective" characteristic time is estimated by replacing $(\tau_{SOLID})_j$ in Table 6 with 38/(2×23) times the values given in Table 7, and by replacing $(\varepsilon_{SOLID})_j$ with 38/(1×23) times the values in Table 7. The conversion factor for $(\tau_{SOLID})_j$ arises from the requirement of only 38 elementary operations when using (4.38) with the single vector $\underline{W}$, whereas 23 operations with each of two vectors are needed when (4.20) is employed in the formation of the dispersion function. The different value of the conversion factor for $(\varepsilon_{SOLID})_j$ is explained by the need of just one vector when representation (4.20) is used to form the eigenfunctions. With only these changes in the values of $(\tau_{SOLID})_j$ and $(\varepsilon_{SOLID})_j$--τ_{XINTEG} remains the same--the procedure described in the preceding paragraph can then be used to estimate τ_{eff} for the delta-matrix version of the high-speed technique. The result is an improvement to

153×10^{-6} sec/step/iteration. The delta-matrix form of the approximation with vanishing $\underline{y}_5$ and $\underline{y}_6$, and $\underline{g}(r)$ retained in the elements of the coefficient matrix, again allows us to ignore program sections XINTEG, EIGSEA, EIGSAC, and EIGMTL. With the appropriate conversion factor of 33/(2×23) applied to $(\tau_{\underline{SOLID}})_{\underline{j}}$, the "effective" characteristic time of this approximation in its delta-matrix form becomes 95×10^{-6} sec/step/iteration. When the fourth-order, "starting" procedure, (65), is cast in its delta-matrix form, (76), a like approach is used to obtain an "effective" characteristic time of 283×10^{-6} sec/step/iteration. In this case, $(\tau_{\underline{SOLID}})_{\underline{j}}$ retain the same values as those given in Table 7 since both (65) and (76) require the same number of elementary operations--2×23 and 46, respectively; $(\varepsilon_{\underline{SOLID}})_{\underline{j}}$ are replaced by 46/(1×23) times the values given in Table 7. This fourth-order, "starting" procedure is described in detail in the following section of the additional notes.

8. COMPUTATIONAL ALGORITHM FOR A FOURTH-ORDER "STARTING" PROCEDURE

Simplified equations of motion based on assumption no. 2. If we apply the two-term expansion (4.10) to relations (4.1) and (4.2), we obtain

$$\begin{aligned} y_5(r) \approx & \left\{\frac{4\pi G}{2\ell+1}[\ell J_1(r) - (\ell+1)J_2(r)]\right\} y_1(r) \\ & + \left\{\frac{4\pi G}{2\ell+1}\ell(\ell+1)[J_1(r) + J_2(r)]\right\} y_3(r) \\ & + \left\{\frac{4\pi G}{2\ell+1}[\ell K_1(r) - (\ell+1)K_2(r)]\right\} \bar{y}_1(r) \\ & + \left\{\frac{4\pi G}{2\ell+1}\ell(\ell+1)[K_1(r) + K_2(r)]\right\} \bar{y}_3(r) \end{aligned} \tag{62}$$

$$\begin{aligned} y_6(r) \approx & \left\{-\frac{4\pi G}{2\ell+1}\frac{\ell(\ell+1)}{r}[J_1(r) + J_2(r)]\right\} y_1(r) \\ & + \left\{-\frac{4\pi G}{2\ell+1}\frac{\ell(\ell+1)}{r}[(\ell+1)J_1(r) - \ell J_2(r)]\right\} y_3(r) \\ & + \left\{-\frac{4\pi G}{2\ell+1}\frac{\ell(\ell+1)}{r}[K_1(r) + K_2(r)]\right\} \bar{y}_1(r) \\ & + \left\{-\frac{4\pi G}{2\ell+1}\frac{\ell(\ell+1)}{r}[(\ell+1)K_1(r) - \ell K_2(r)]\right\} \bar{y}_3(r) \end{aligned} \tag{63}$$

where

$$
\begin{aligned}
J_1(r) &= \int_0^r \rho(\zeta)\ \left(\frac{\zeta}{r}\right)^{\ell+1} d\zeta \\
J_2(r) &= \int_r^a \rho(\zeta)\ \left(\frac{\zeta}{r}\right)^{-\ell} d\zeta \\
K_1(r) &= \int_0^r \rho(\zeta)\ (\zeta - r)\ \left(\frac{\zeta}{r}\right)^{\ell+1} d\zeta \\
K_2(r) &= \int_r^a \rho(\zeta)\ (\zeta - r)\ \left(\frac{\zeta}{r}\right)^{-\ell} d\zeta
\end{aligned}
\tag{64}
$$

which can be evaluated directly since $\underline{\rho}$ is assigned an explicit specification: a sequence of linear functions of $\underline{\zeta}$. Substitution of (62) and (63) into the sixth-order equations of motion (equation (2.1)) yields the fourth-order system

$$
\begin{bmatrix} \overline{y}_1 \\ \overline{y}_2 \\ \overline{y}_3 \\ \overline{y}_4 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ C_{21} & C_{22} & C_{23} & C_{24} \\ -a_{33} & 0 & a_{33} & a_{34} \\ C_{41} & C_{42} & C_{43} & C_{44} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}
\tag{65}
$$

where the modified elements of the coefficient matrix are

$$
\begin{aligned}
C_{21} &= a_{21} + \frac{4\pi G}{2\ell+1}\frac{\ell(\ell+1)\rho}{r}\Big\{J_1(r) + J_2(r) + a_{11}[K_1(r) + K_2(r)] \\
&\qquad - a_{33}[(\ell+1)K_1(r) - \ell K_2(r)]\Big\} \\
C_{22} &= a_{22} + \frac{4\pi G}{2\ell+1}\frac{\ell(\ell+1)\rho}{r}\, a_{12}[K_1(r) + K_2(r)] \\
C_{23} &= a_{23} + \frac{4\pi G}{2\ell+1}\frac{\ell(\ell+1)\rho}{r}\Big\{(\ell+1)J_1(r) - \ell J_2(r) + a_{13}[K_1(r) + K_2(r)] \\
&\qquad + a_{33}[(\ell+1)K_1(r) - \ell K_2(r)]\Big\} \\
C_{24} &= a_{24} + \frac{4\pi G}{2\ell+1}\frac{\ell(\ell+1)\rho}{r}\, a_{34}[(\ell+1)K_1(r) - \ell K_2(r)] \\
C_{41} &= a_{41} - \frac{4\pi G}{2\ell+1}\frac{\rho}{r}\Big\{\ell J_1(r) - (\ell+1)J_2(r) + a_{11}[\ell K_1(r) - (\ell+1)K_2(r)] \\
&\qquad - a_{33}\ell(\ell+1)[K_1(r) + K_2(r)]\Big\} \\
C_{42} &= a_{42} - \frac{4\pi G}{2\ell+1}\frac{\rho}{r}\, a_{12}[\ell K_1(r) - (\ell+1)K_2(r)] \\
C_{43} &= a_{43} - \frac{4\pi G}{2\ell+1}\frac{\rho}{r}\Big\{\ell(\ell+1)[J_1(r) + J_2(r)] + a_{13}[\ell K_1(r) - (\ell+1)K_2(r)] \\
&\qquad + a_{33}\ell(\ell+1)[K_1(r) + K_2(r)]\Big\}
\end{aligned}
\tag{66}
$$

$$C_{44} = a_{44} - \frac{4\pi G}{2\ell+1}\frac{\rho}{r}\, a_{34}\ell(\ell+1)[K_1(r) + K_2(r)] \quad .$$

The preceding equations of motion apply to the solid portions of the earth. If (4.10) is applied to an ocean modeled by a single, homogeneous, liquid layer, the original fourth-order system (equation (2.16)) reduces to

$$\begin{bmatrix} \bar{y}_1 \\ \bar{y}_2 \end{bmatrix} = \begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \tag{67}$$

with

$$\begin{aligned}
E_{11} &= (1+H)^{-1}\left\{b_{11}(1-b_{25}A_4+\rho B_4) + b_{15}[A_1+b_{21}A_4+\rho(A_1B_4-A_4B_1)]\right\} \\
E_{12} &= (1+H)^{-1}\left\{b_{12}(1-b_{25}A_4+\rho B_4) + b_{15}[A_3+b_{22}A_4+\rho(A_3B_4-A_4B_3)]\right\} \\
E_{21} &= (1+H)^{-1}\left\{b_{21}(1-b_{15}A_2) + b_{11}(b_{25}A_2-\rho B_2) + b_{25}A_1 - \rho B_1 + \rho b_{15}(A_2B_1-A_1B_2)\right\} \\
E_{22} &= (1+H)^{-1}\left\{b_{22}(1-b_{15}A_2) + b_{12}(b_{25}A_2-\rho B_2) + b_{25}A_3 - \rho B_3 + \rho b_{15}(A_2B_3-A_3B_2)\right\}
\end{aligned} \tag{68}$$

and

$$H = -b_{25}A_4 + \rho B_4 + b_{15}[\rho(A_4B_2-A_2B_4)-A_2] \quad . \tag{69}$$

The quantities $\underline{A_i}$ and $\underline{B_i}$ are given by

$$\begin{aligned}
A_1 &= \left[\frac{g}{r}A_5 + \omega^2\frac{M_0}{N_1} + \bar{g} - 4\pi G\rho - \frac{1}{\omega^2/\rho+S_1}\left(R_0\omega^2+S_1\bar{g}-4\pi G\rho S_1\right)\right]\left[\frac{1}{r}\left(\frac{\omega^2}{\rho N_1}+A_5\right)\right]^{-1} \\
A_2 &= \left[r\omega^2\frac{M_1}{N_1} + g - \frac{1}{\omega^2/\rho+S_1}\left(r\omega^2R_1+S_1g\right)\right]\left[\frac{1}{r}\left(\frac{\omega^2}{\rho N_1}+A_5\right)\right]^{-1} \\
A_3 &= \left[-A_5/r\rho\right]\left[\frac{1}{r}\left(\frac{\omega^2}{\rho N_1}+A_5\right)\right]^{-1} \\
A_4 &= \frac{-\omega^2/\rho^2}{\omega^2/\rho+S_1}\left[\frac{1}{r}\left(\frac{\omega^2}{\rho N_1}+A_5\right)\right]^{-1} \\
A_5 &= B_5 - B_6 \\
B_1 &= \left\{\left[\frac{g}{r}B_5 + \omega^2\frac{M_0}{N_1} + \bar{g} - 4\pi G\rho\right]\left[\frac{\omega^2}{\rho N_1} + B_5\right]^{-1}\right. \\
&\quad \left.-\left[\frac{g}{r}B_6 + \frac{1}{\omega^2/\rho+S_1}\left(R_0\omega^2+S_1\bar{g}-4\pi G\rho S_1\right)\right]\frac{1}{B_6}\right\}\left[\left(\frac{\omega^2}{\rho N_1}+B_5\right)^{-1} - \frac{1}{B_6}\right]^{-1} \\
B_2 &= \left\{\left[r\omega^2\frac{M_1}{N_1} + g\right]\left[\frac{\omega^2}{\rho N_1} + B_5\right]^{-1} - \left[\frac{1}{\omega^2/\rho+S_1}\left(r\omega^2R_1+gS_1\right)\right]\frac{1}{B_6}\right\} \\
&\quad \times\left[\left(\frac{\omega^2}{\rho N_1}+B_5\right)^{-1} - \frac{1}{B_6}\right]^{-1} \\
B_3 &= \frac{1}{\rho r}\left[1 - B_5\left(\frac{\omega^2}{\rho N_1}+B_5\right)^{-1}\right]\left[\left(\frac{\omega^2}{\rho N_1}+B_5\right)^{-1} - \frac{1}{B_6}\right]^{-1}
\end{aligned} \tag{70}$$

$$B_4 = \frac{1}{\rho}\left[-\left(\frac{\omega^2}{\rho N_1}+B_5\right)^{-1} + \left(\frac{S_1}{\omega^2/\rho+S_1}\right)\frac{1}{B_6}\right]\left[\left(\frac{\omega^2}{\rho N_1}+B_5\right)^{-1} - \frac{1}{B_6}\right]^{-1}$$

$$B_5 = \frac{N_0}{N_1} - 1$$

$$B_6 = \frac{S_0 - S_1}{\omega^2/\rho + S_1}$$

where

$$M_0 = \frac{4\pi G}{2\ell+1}\left\{\frac{\ell}{\ell+2} - \frac{\ell+1}{\ell-1}\left[1 - \left(\frac{r}{a}\right)^{\ell-1}\right]\right\}$$

$$M_1 = \frac{4\pi G}{2\ell+1}\left\{-\frac{\ell}{(\ell+2)(\ell+3)} - \frac{\ell+1}{(\ell-1)(\ell-2)}\left[1 - \left(\frac{r}{a}\right)^{\ell-2}\left(\ell-1-(\ell-2)\frac{r}{a}\right)\right]\right\}$$

$$N_0 = \frac{4\pi G}{2\ell+1}\,\ell(\ell+1)\left\{\frac{1}{\ell+2} + \frac{1}{\ell-1}\left[1 - \left(\frac{r}{a}\right)^{\ell-1}\right]\right\}$$

$$N_1 = \frac{4\pi G}{2\ell+1}\,\ell(\ell+1)\left\{-\frac{1}{(\ell+2)(\ell+3)} + \frac{1}{(\ell-1)(\ell-2)}\left[1 - \left(\frac{r}{a}\right)^{\ell-2}\left(\ell-1-(\ell-2)\frac{r}{a}\right)\right]\right\} \tag{71}$$

$$R_0 = -N_0$$

$$R_1 = -N_1$$

$$S_0 = \frac{4\pi G}{2\ell+1}\,\ell(\ell+1)\left\{-\frac{\ell+1}{\ell+2} + \frac{\ell}{\ell-1}\left[1 - \left(\frac{r}{a}\right)^{\ell-1}\right]\right\}$$

$$S_1 = \frac{4\pi G}{2\ell+1}\,\ell(\ell+1)\left\{\frac{\ell+1}{(\ell+2)(\ell+3)} + \frac{\ell}{(\ell-1)(\ell-2)}\left[1 - \left(\frac{r}{a}\right)^{\ell-2}\left(\ell-1-(\ell-2)\frac{r}{a}\right)\right]\right\} .$$

Computational algorithm. The details concerning the initiation and termination of integration with the fourth-order, "starting" procedure are the same as those described in Computational Algorithm for High-Speed Procedure (fourth section of main text). From the subroutine structure of this procedure, which is shown in Figure 28, it is seen that such a "starting" method is highly compatible with the high-speed procedure of Figures 23 and 27. Only one new subroutine need be added to the program package. This compatibility favors the use of the fourth-order, "starting" procedure over the full, sixth-order method since the latter would necessitate the addition of six new subroutines to the program package. Thus, if the timing efficiency of the fourth-order approach is not significantly worse than that of the sixth-order algorithm, the former

"starting" procedure should be used with the high-speed method of Figures 23 and 27. Coding efficiency (physical length of code) is more important than timing efficiency here, since the "starting" procedure is applied to only about three percent of the frequencies treated; the rest being handled by the high-speed approach.

With all possible optimization included in the sixth-order method (see Section 7 of the additional notes), the total time per frequency is about

$$(252\times10^{-6} \text{ sec/step/iteration}) \times n_{TOTAL} \times (N_{\underline{\ell}})_{6th} \qquad (72)$$

(Table 5), where n_{TOTAL} is the number of integration steps and $(N_{\underline{\ell}})_{6th}$ is the required number of iterations over $\underline{\ell}$ to obtain the desired accuracy in $\underline{\ell}(\omega)$. For the fourth-order method, reference to Figure 28 shows that each iteration over $\underline{\ell}$ requires a time of about

$$T_{XINTEG} + T_{HPC} + T_{EIG} \qquad (73)$$

(see Section 7 of the additional notes), where we assume that

$$\tau_{YINTEG} \approx \tau_{XINTEG} \quad . \qquad (74)$$

Thus the total time per frequency for the fourth-order "starting" procedure is about

$$[(\tau_{eff})_{4th}]_{START} \times n_{TOTAL} \times [(N_{\underline{\ell}})_{4th}]_{START} \quad , \qquad (75)$$

where this effective characteristic time is about 260×10^{-6} sec/step/iteration (Table 5; Table 8). Therefore, if the required number of fourth-order iterations, $[(N_{\underline{\ell}})_{4th}]_{START}$, is not much above $(N_{\underline{\ell}})_{6th}$, the fourth-order, "starting" procedure is to be preferred.

Loss-of-precision control. If the sixth-order method of starting is selected, the augmented algorithm described in the third section of the main text will provide this control. If the fourth-order technique is chosen, a simplified form of the orthogonalization which is discussed there, can also be used to control the loss-of-precision problem.

As in the high-speed procedure, the simplification to a fourth-order "starting" procedure also makes it possible to use the delta-matrix

approach to control precision loss. The development of the delta-matrix form of the equations of motion follows that given in the fourth section of the main text for the high-speed procedure; the last simplification given there--equations (4.37)-(4.38)--however, is not possible here. The result is

$$\begin{bmatrix} \overline{W}_{01} \\ \overline{W}_2 \\ \overline{W}_3 \\ \overline{W}_4 \\ \overline{W}_5 \\ \overline{W}_{06} \end{bmatrix} = \begin{bmatrix} a_{11}+C_{22} & C_{23} & C_{24} & -a_{13} & 0 & 0 \\ 0 & a_{11}+a_{33} & a_{34} & a_{12} & 0 & 0 \\ C_{42} & C_{43} & a_{11}+C_{44} & 0 & a_{12} & a_{13} \\ a_{33} & C_{21} & 0 & C_{22}+a_{33} & a_{34} & -C_{24} \\ -C_{41} & 0 & C_{21} & C_{43} & C_{22}+C_{44} & C_{23} \\ 0 & -C_{41} & -a_{33} & -C_{42} & 0 & a_{33}+C_{44} \end{bmatrix} \begin{bmatrix} W_{01} \\ W_2 \\ W_3 \\ W_4 \\ W_5 \\ W_{06} \end{bmatrix} . \quad (76)$$

The relative speeds of (65) and its delta-matrix extension, (76), are estimated by noting that each of $\underline{X}_1$ and $\underline{X}_2$ requires 23 elementary operations when (65) is used, whereas the single vector in (76) requires 46 operations. Thus, to a first approximation, the computation time of the delta-matrix algorithm would be 46/(2×23) times that of (65), i.e. the computation times would be approximately the same if we assume that both approaches need a like number of iterations over $\underline{\ell}$. This estimate, however, applies only to the formation of the dispersion function: HPCSAC and HPCMTL in Figure 28. As explained in the last paragraph of Section 7 of the additional notes, the formation of $\underline{I}_i(\underline{r})$ and the eigenfunctions will increase the characteristic time from that of (65)--260×10^{-6}--to 283×10^{-6} sec/step/iteration. This increase of about 9 percent when using the delta-matrix extension of the fourth-order, "starting" method, means that considerations based on computation speed yield no clear preference between delta matrices and orthogonalization for controlling precision loss with

this "starting" method.

If we assume that further testing will show that $[(N_\ell)_{4th}]_{START}$ is not significantly greater than $(N_\ell)_{6th}$, then the consideration of coding efficiency will not only indicate that the fourth-order, "starting" procedure be used in preference to the sixth-order algorithm, but will also indicate that the loss-of-precision problem be treated by orthogonalization rather than delta matrices; this, because orthogonalization can be included in the subroutine structure of combined Figures 27 and 28. The high-speed procedure based on (4.38), and the "starting" procedure based on (76), would require separate subroutine structures for each of these delta-matrix procedures, since (4.38) and (76) do not have the same form.

TABLE 1

CONSTANTS FOR INTEGRATION THROUGH SUCCESSIVE STEP-SIZE REGIONS ILLUSTRATED IN PROGRAM SEGMENTS GIVEN IN FIGURE 4. CONSTANTS CORRESPOND TO 4 SIGNIFICANT FIGURES IN COMPUTED PHASE VELOCITY.

I	N1(I)	N2(I)	Integration Step Size (km)
1	5	11	-1.5625
2	12	16	-3.1250
3	17	21	-6.2500
4	22	*	-12.5000

* N2(4) is specified so as to allow integration to proceed to the deepest point within the solid mantle, while maintaining a step size of -12.5 km. NEND is determined, at each period, by the input value of $\underline{r}_0$; it must satisfy NEND $\leq$ N2(4).

TABLE 2

RESULTS OF NUMERICAL TESTS OF THE OVERFLOW PROBLEM WHEN NORMALIZATION IS NOT INCLUDED IN OUR OPTIMIZATION OF THE BASIC AJP FORMULATION. AN AVERAGE (OCEANIC) EARTH STRUCTURE (WIGGINS, 1968), AND A PERIOD OF 50 SECONDS, WERE USED IN THE TESTS. THE VALUE OR $\underline{r}_0$ IS THE MAXIMUM AT WHICH OVERFLOW OCCURS; $(\underline{a} - \underline{r}_0)/\underline{\Lambda}$ IS THE CORRESPONDING NUMBER OF WAVELENGTHS OF STRUCTURE, FROM THE SURFACE OF THE EARTH DOWN TO $\underline{r} = \underline{r}_0$. THESE, AND LARGER VALUES OF $(\underline{a} - \underline{r}_0)/\underline{\Lambda}$, YIELD OVERFLOW.

Mode Number	r_0 (km)	$(a - r_0)/\Lambda$
0	4600	8.7
1	4300	8.3
2	4000	8.2
3	3800	7.9

TABLE 3

REFERENCE DISPERSION RESULTS FOR GRAVITATING, HOMOGENEOUS SPHERE.

STRUCTURAL PARAMETERS ARE GIVEN IN THE THIRD SECTION OF THE MAIN TEXT.

PHASE VELOCITIES ARE OBTAINED FROM (2.5).

Radial Mode No., n	Polar Order No., ℓ	Period (sec)	Radial Mode No., n	Polar Order No., ℓ	Period (sec)
0	820	10.105 546 697	47	495	10.256 530 851
1	745	10.154 934 672	48	495	10.200 889 049
2	730	10.188 149 987	49	490	10.222 011 606
3	720	10.184 634 500	50	485	10.243 720 617
4	710	10.197 342 089	51	485	10.189 089 890
5	700	10.220 909 004	52	480	10.211 461 419
6	695	10.183 478 806	53	475	10.234 408 821
7	685	10.221 023 634	54	475	10.180 683 044
8	680	10.194 301 019	55	470	10.204 254 136
9	670	10.241 289 518	56	465	10.228 394 789
10	665	10.221 983 680	57	460	10.253 108 028
11	660	10.205 676 104	58	460	10.200 214 171
12	655	10.191 999 296	59	455	10.225 514 169
13	645	10.251 261 784	60	450	10.251 385 363
14	640	10.242 082 839	61	450	10.199 197 873
15	635	10.234 830 574	62	445	10.225 632 605
16	630	10.229 339 415	63	440	10.252 639 960
17	625	10.225 469 526	64	440	10.201 087 375
18	620	10.223 101 663	65	435	10.228 639 321
19	615	10.222 133 300	66	435	10.177 701 662
20	610	10.222 475 643	67	430	10.205 783 707
21	605	10.224 051 306	68	425	10.234 439 569
22	600	10.226 792 468	69	425	10.184 028 516
23	595	10.230 639 411	70	420	10.213 198 285
24	590	10.235 539 321	71	415	10.242 945 251
25	585	10.241 445 331	72	415	10.192 999 742
26	580	10.248 315 713	73	410	10.223 242 864
27	575	10.256 113 223	74	405	10.254 064 602
28	575	10.191 424 481	75	405	10.204 523 596
29	570	10.200 679 017	76	400	10.235 820 512
30	565	10.210 758 353	77	398	10.218 958 427
31	560	10.221 638 585	78	394	10.234 716 475
32	555	10.233 298 231	79	392	10.218 270 984
33	550	10.245 717 987	80	388	10.234 553 568
34	550	10.184 642 793	81	386	10.218 505 023
35	545	10.198 169 627	82	382	10.235 299 002
36	540	10.212 398 148	83	380	10.219 631 725
37	535	10.227 315 929	84	376	10.236 930 863
38	530	10.242 911 896	85	374	10.221 637 113
39	530	10.184 166 212	86	370	10.239 449 868
40	525	10.200 680 807	87	368	10.224 538 681
41	520	10.217 836 041	88	366	10.209 849 784
42	515	10.235 625 269	89	362	10.228 434 933
43	510	10.254 042 788	90	360	10.214 243 125
44	510	10.197 102 934	91	356	10.233 748 427
45	505	10.216 308 522	92	354	10.220 658 958
46	500	10.236 118 681			

TABLE 4

RESULTS OF NUMERICAL EXPERIMENT TO OBTAIN ROUGH ESTIMATE OF EFFECT OF ERRORS IN $\underline{y}_5$ AND $\underline{y}_6$ ON COMPUTED PHASE VELOCITIES. EXPERIMENT PERFORMED WITH FULL, SIXTH-ORDER ALGORITHM; ERRORS IN $\underline{y}_5$ AND $\underline{y}_6$ SIMULATED BY VARYING GRAVITATIONAL CONSTANT $\underline{G}$; AN "AVERAGE" OCEANIC STRUCTURE OF THE CIT-11 TYPE (FIGURE 5) WAS USED.

Period = 50 seconds; Radial Mode Numbers, n = 0(1)7

$\delta G/G \approx \delta y_5/y_5 \approx \delta y_6/y_6$	$\frac{1}{n_{max}+1}\sum_{n}\lvert c - c_{true}\rvert$ (km/sec)
0.1	0.00010
0.2	0.00019
0.4	0.00039
0.6	0.00060
0.8	0.00083
1.0	0.00107

TABLE 5A

SUMMARY OF RANGES USED IN NUMERICAL TESTING OF FOURTH-ORDER, HIGH-SPEED PROCEDURE. CONSTANT FREQUENCY INTERVAL BETWEEN COMPUTED DISPERSION POINTS IS 0.0005 CPS. SEE TABLE 4 AND FIGURE 5 FOR STRUCTURE USED IN THIS PARTICULAR SET OF TESTS.

Radial Mode No.	Frequencies (cps) (and Periods (sec)) used in Computational Tests		Total Number of Frequencies
	"Starting" Procedure	High-Speed Procedure	
0	0.0020 (500) 0.0025 (400) 0.0030 (333)	0.0035-0.1000	197
1	0.0035 (286) 0.0040 (250) 0.0045 (222)	0.0050-0.1000	194
2	0.0060 (167) 0.0065 (154) 0.0070 (143)	0.0075-0.1000	182

TABLE 5B

SUMMARY OF FINAL TIMING RESULTS FOR THE VARIOUS PROCEDURES DESCRIBED IN THIS REPORT. ALL CHARACTERISTIC TIMES REFER TO DOUBLE-PRECISION COMPUTATIONS ON AN IBM 360/91 COMPUTER. SEE PORTER ET AL. (1980) FOR APPROXIMATE CONVERSION FACTORS TO EQUIVALENT TIMES ON MANY OTHER COMPUTERS.

Wave Type	Structure	Computational Method	Control of Precision Loss	"Effective" Characteristic Time	Reference
Love, or torsional	flat, or spherical, homogeneous layers	Thomson-Haskell technique	[1]	55×10^{-6} sec/ layer/ iteration	Schwab and Knopoff (1972)
Rayleigh	flat, non-gravitating, homogeneous layers	Knopoff's method	[2]	110×10^{-6} sec/ layer/ iteration	Schwab and Knopoff (1972)
Rayleigh (spheroidal)	spherical, approx. with $y_5=y_6=0$ but $g(r)$ left in a_{ij}	high-speed procedures	none	132×10^{-6} sec/ step/ iteration	eq. (4.20)
			delta matrix	95×10^{-6} sec/ step/ iteration	eq. (4.38)
			orthog.	[3]	[3]
Rayleigh (spheroidal)	spherical, gravitating	high-speed procedures (Figure 27)	none	164×10^{-6} sec/ step/ iteration	eq. (4.20)
			delta matrix	153×10^{-6} sec/ step/ iteration	eq. (4.38)
			orthog.	[3]	[3]
Rayleigh (spheroidal)	spherical, gravitating	"starting" procedures (Figure 28)	none	260×10^{-6} sec/ step/ iteration	eq. (65)
			delta matrix	283×10^{-6} sec/ step/ iteration	eq. (76)
			orthog.	[3]	[3]
Rayleigh (spheroidal)	spherical, gravitating	full, sixth-order algorithm	none	252×10^{-6} sec/ step/ iteration	eq. (2.1)
			orthog.	[3]	[3]

[1] There is no loss-of-precision problem in this case.

[2] This case contains the equivalent of the delta-matrix approach for controlling precision loss (Schwab, 1970).

[3] When orthogonalization is employed to control precision loss, the "effective" characteristic time increases by 5-10 percent over that when no feature is included to eliminate this problem. The reference equation is unchanged when orthogonalization is added to the formulation without control of precision loss.

TABLE 6

FORMULAS FOR DETAILED TIMING ESTIMATES FOR RAYLEIGH-WAVE DISPERSION COMPUTATIONS WHEN THE FOURTH-ORDER FORMULATIONS ARE USED FOR A SPHERICAL, GRAVITATING EARTH

Section of Algorithm	Execution Time per Frequency in Different Sections of the Algorithm	
	Partial	Total
XINTEG	$n_{TOTAL} \times \tau_{XINTEG}$	T_{XINTEG}
HPCSEA	$(n_{SEA})_{RK} \times (\tau_{SEA})_{RK}$ $+ (n_{SEA})_{PC} \times (\tau_{SEA})_{PC}$	
HPCSAC	$N_{SED} \times n_{SED} \times (\tau_{SOLID})_{RK}$ $+ N_{CRUST} \times (n_{CRUST})_{RK} \times (\tau_{SOLID})_{RK}$ $+ N_{CRUST} \times (n_{CRUST})_{PC} \times (\tau_{SOLID})_{PC}$	T_{HPC}
HPCMTL	$(n_{MTL})_{RK} \times (\tau_{SOLID})_{RK}$ $+ (n_{MTL})_{PC} \times (\tau_{SOLID})_{PC}$	
EIGSEA	$(n_{SEA})_{RK} \times (\varepsilon_{SEA})_{RK}$ $+ (n_{SEA})_{PC} \times (\varepsilon_{SEA})_{PC}$	
EIGSAC	$N_{SED} \times n_{SED} \times (\varepsilon_{SOLID})_{RK}$ $+ N_{CRUST} \times (n_{CRUST})_{RK} \times (\varepsilon_{SOLID})_{RK}$ $+ N_{CRUST} \times (n_{CRUST})_{PC} \times (\varepsilon_{SOLID})_{PC}$	T_{EIG}
EIGMTL	$(n_{MTL})_{RK} \times (\varepsilon_{SOLID})_{RK}$ $+ (n_{MTL})_{PC} \times (\varepsilon_{SOLID})_{PC}$	

Fourth-Order Procedures	Total Time per Frequency
high-speed	$T_{XINTEG} + [(N_\ell)_{4th}]_{HS} \times T_{HPC} + T_{EIG}$
"starting"	$[(N_\ell)_{4th}]_{START} \times (T_{YINTEG} + T_{HPC} + T_{EIG})$

TABLE 7

VALUES OF PARAMETERS APPEARING IN TABLE 6

Characteristic Times[1] (10^{-6} sec/integration step)		Number of Integration Steps[2]	Long-Period[3]	Short-Period[3]
τ_{XINTEG}	58.2	$(n_{SEA})_{RK}$	3	same
$(\tau_{SEA})_{RK}$	41.6/iteration	$(n_{SEA})_{PC}$	2	same
$(\tau_{SEA})_{PC}$	31.5/iteration	n_{SED}	1	same
$(\tau_{SOLID})_{RK}$	183/iteration	$(n_{CRUST})_{RK}$	3	same
$(\tau_{SOLID})_{PC}$	131/iteration	$(n_{CRUST})_{PC}$	7	same
		$(n_{MTL})_{RK}$	3	same
$(\epsilon_{SEA})_{RK}$	43.9	$(n_{MTL})_{PC}$[4]	242	467
$(\epsilon_{SEA})_{PC}$	33.8			
$(\epsilon_{SOLID})_{RK}$	91.0			
$(\epsilon_{SOLID})_{PC}$	69.7			

[1] These times correspond to double-precision computations on an IBM 360/91 computer. Approximate conversion factors to equivalent times on many other computers will be found in Porter et al. (1980).

[2] $n_{TOTAL} = (n_{SEA})_{RK} + (n_{SEA})_{PC} + N_{SED} \times n_{SED} + N_{CRUST} \times [(n_{CRUST})_{RK} + (n_{CRUST})_{PC}] + (n_{MTL})_{RK} + (n_{MTL})_{PC}$

[3] See second section of main text (Numerical Technique for Integrating the System of Differential Equations) for further details concerning step sizes. Maximum step size must be reduced below 12.5 km, as period decreases from 25 to 10 seconds and radial mode number increases (see final paragraph of Conclusions in third section of main text).

[4] The numbers given above for $(n_{MTL})_{PC}$ correspond to the use of all of the mantle down to the mantle-core discontinuity, for an oceanic mantle. From the information given in the second section of the main text (Numerical

Technique for Integrating the System of Differential Equations; Figures 6 and 8), estimates can be obtained for the reduced values of $(n_{MTL})_{PC}$ as a function of radial mode number and frequency.

TABLE 8

"EFFECTIVE" CHARACTERISTIC TIMES FOR TYPICAL OCEANIC AND CONTINENTAL STRUCTURES, AND LONG- AND SHORT-PERIOD (SEE FOOTNOTES 3 AND 4 OF TABLE 7) COMPUTATIONS. TIMES ARE IN UNITS OF 10^{-6} SEC/INTEGRATION STEP/ITERATION ON AN IBM 360/91 COMPUTER (SEE FOOTNOTE 1 OF TABLE 7).

Fourth-Order, High-Speed Procedure	Gravitating Structures		Approximation with vanishing y_5 and y_6, but g(r) left in elements a_{ij}	
	Oceanic	Continental	Oceanic	Continental
long-period computations	162	167	130	135
short-period computations	163	164	131	133

Fourth-Order, "Starting" Procedure	Gravitating Structures	
	Oceanic	Continental
long-period computations	258	264
short-period computations	258	262

Elements to be evaluated external to both ω and c loops:

a_{11} a_{12} a_{22} a_{26} a_{33} a_{34}

a_{41} a_{42} a_{44} a_{45} a_{51} a_{66}

Auxiliary quantities to be evaluated external to both ω and c loops:

$d_{13}=\lambda a_{12}/r$ $d_{21}=4\mu(3\lambda+2\mu)a_{12}/r^2$

$d_{43}=-2\mu/r^2$ $d_{63}=-4\pi G\rho$

$d_{65}=1/r^2$ $e_{43}=4\mu(\lambda+\mu)a_{12}/r^2$

```
ω loop
    OMEGSQ=ω²
    DO 10 I=1,N
    TEMP=-a26(I)×OMEGSQ
    a21(I)=d21(I)-TEMP
10  f43(I)=d43(I)-TEMP
  c loop
    ORDER=ℓ(ℓ+1)
    DO 20 I=1,N
    a13(I)=d13(I)×ORDER
    a23(I)=a41(I)×ORDER
    a63(I)=d63(I)×ORDER
    a24(I)=a33(I)×ORDER
    a65(I)=d65(I)×ORDER
20  a43(I)=f43(I)+e43(I)×ORDER
  Integration loop, in which
    all of X1, X2, X3 are
    treated at the same time
```

FIG. 1. Schematic representation of optimized method for evaluating the matrix elements $a_{ij}(r_k)$ in the treatment of the solid sedimentary layers, the $a_{ij}(r_k)$ subsedimentary crustal layers, and the mantle. The quantities λ and μ are Lamé's constants, ρ is the density, G is the gravitational constant, and N is the number of depths at which $a_{ij}(r_k)$ must be evaluated. It is seen that most of the procedure for $a_{ij}(r_k)$ evaluating $a_{ij}(r_k)$ can even be removed from within the ω and c loops: Within the c loop, each new c value requires only 6N + 1 assignments, 6N + 1 multiplications, and N + 1 additions; within the ω loop, each new ω value requires only 3N + 1 assignments, N + 1 multiplications, and 2N subtractions. All other portions of the element determinations are performed external to these loops.

Evaluate b_{66} external to both ω and c loops. Auxiliary quantities to be evaluated external to both ω and c loops:

$p_{12} = -1/r^2\rho$ $\quad$ $p_{62} = 4\pi G/r^2$

$p_{15} = -1/r^2$ $\quad$ $p_{65} = p_{62}\rho$

$p_{22} = -g(r)/r^2$ $\quad$ $p_{61} = -p_{65}\, g(r)$

$p_{25} = p_{22}\rho$ $\quad$ $s_{21} = -4\pi g(r)/r$

$p_{21} = -p_{25}\, g(r)$ $\quad$ XLAINV $= 1/\alpha^2\rho$

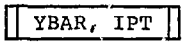
```
ω loop
    RHMOSQ=-ρω²
    OMSQIN=1/ω²
    DO 10 I=1,M
    h21(I)=p21(I)×OMSQIN
    q21(I)=s21(I)+RHMOSQ
    h61(I)=p61(I)×OMSQIN
    h12(I)=p12(I)×OMSQIN
    h22(I)=p22(I)×OMSQIN
    h62(I)=p62(I)×OMSQIN
    h15(I)=p15(I)×OMSQIN
    h25(I)=p25(I)×OMSQIN
10  h65(I)=p65(I)×OMSQIN
  c loop
    ORDER=ℓ(ℓ+1)
    DO 20 I=1,M
    b22(I)=h22(I)×ORDER
    b11(I)=b66(I)-b22(I)
    b21(I)=h21(I)×ORDER+q21(I)
    b61(I)=h61(I)×ORDER
    b12(I)=h12(I)×ORDER+XLAINV
    b62(I)=h62(I)×ORDER
    b15(I)=h15(I)×ORDER
    b25(I)=h25(I)×ORDER
20  b65(I)=[h65(I)-p15(I)]×ORDER
    Integration loop, in which both
      Z1 and Z2 are treated at the
      same time
```

FIG. 2. Schematic representation of optimized method for evaluating the matrix elements $b_{ij}(r_k)$ in the treatment of the homogeneous oceanic (liquid) layer. The $^{ij}{}^{k}$ quantity α is the compressional-wave velocity, g(r) is the acceleration due to gravity, and M is the number of depths at which $b_{ij}(r_k)$ must be evaluated.

(a)

```
      YBAR(1)=A1(IPT)*Y(1)+A7(IPT)*Y(2)+A13(IPT)*Y(3)
      YBAR(2)=A2(IPT)*Y(1)+A8(IPT)*Y(2)+A14(IPT)*Y(3)+
     1        A20(IPT)*Y(4)+A32(IPT)*Y(6)
      YBAR(3)=A15(IPT)*(Y(3)-Y(1))+A21(IPT)*Y(4)
      YBAR(4)=A4(IPT)*Y(1)+A10(IPT)*Y(2)+A16(IPT)*Y(3)+
     1        A22(IPT)*Y(4)+A28(IPT)*Y(5)
      YBAR(5)=A5(IPT)*Y(1)+Y(6)
      YBAR(6)=A18(IPT)*Y(3)+A30(IPT)*Y(5)+A36(IPT)*Y(6)
```

(b)



(c)

$$\left[a_{ij}\right] = \begin{bmatrix} A1 & A7 & A13 & 0 & 0 & 0 \\ A2 & A8 & A14 & A20 & 0 & A32 \\ -A15 & 0 & A15 & A21 & 0 & 0 \\ A4 & A10 & A16 & A22 & A28 & 0 \\ A5 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & A18 & 0 & A30 & A36 \end{bmatrix}$$

FIG. 3. (a) FORTRAN IV program segment for the basic matrix multiplication in our optimization of the AJP formulation for solid layers; (b) symbolic representation of (a), which is used in Figure 4a; and (c) definition of one-dimensional array used in (a) to represent 6×6 matrix in (2.1). The integer IPT is the index specifying the value of r, and A1 through A36 are dimensioned to 300.

(Figure 4a. First of two pages)

```
C BEGIN APPLICATION OF PREDICTOR-CORRECTOR METHOD.
      DO 110 I=1,6
  110 PMNUSC(I)=0.0D+00
      HH=0.5D+00*1.5625D+00
C LOOP OVER REGIONS WITH DIFFERENT STEP SIZES.
      DO 180 IREG=1,NUMREG
      HH=HH+HH
      NSTART=N1(IREG)
      NSTOP=N2(IREG)
      NTEMP=5
      IT=0
C LOOP OVER DEPTH IN CURRENT STEP-SIZE REGION.
      DO 170 N=NSTART,NSTOP
      IF(IT.EQ.4)  GO TO 115
      IT=NTEMP-4
      ITP1=IT+1
      ITP3=IT+3
      ITP8=IT+8
      ITP9=IT+9
      ITP10=IT+10
  115 DO 120 I=1,6
C SET PREDICTOR  P(I).
      P(I)=B(IT,I)+COEFF1*(2.0D+00*(B(ITP10,I)+B(ITP8,I))
     1     -B(ITP9,I))
C SET MODIFIED PREDICTOR XM(I).
  120 XM(I)=P(I)-.9256198347107438D+00*PMNUSC(I)
      IPT=IPT+1
      CALL XMBAR(IPT)
      DO 130 I=1,6
C SET CORRECTOR C(I)
      C(I)=.125D+00*(9.0D+00*B(ITP3,I)-B(ITP1,I)
     1     +COEFF2*(XMBAR(I)+2.0D+00*B(ITP10,I)
     2     -B(ITP9,I)))
      PMNUSC(I)=P(I)-C(I)
C SET SOLUTION VECTOR AT NTH DEPTH.
  130 Y(I)=C(I)+.07438016528925620D+00
     1     *PMNUSC(I)
      IF(N.EQ.NEND)  RETURN
C SET DERIVATIVE OF SOLUTION VECTOR AT NTH DEPTH.
      CALL YBAR(IPT)
      IF(NTEMP.GT.7)  GO TO 150
      NTMPP7=NTEMP+7
      DO 140 I=1,6
      B(NTEMP,I)=Y(I)
  140 B(NTMPP7,I)=YBAR(I)
      NTEMP=NTEMP+1
      GO TO 170
  150 DO 160 I=1,6
      B(1,I)=B(2,I)
      B(2,I)=B(3,I)
      B(3,I)=B(4,I)
      B(4,I)=B(5,I)
```

(Figure 4a. Second of two pages)

```
      B(5,I)=B(6,I)
      B(6,I)=B(7,I)
      B(7,I)=Y(I)
      B(8,I)=B(9,I)
      B(9,I)=B(10,I)
      B(10,I)=B(11,I)
      B(11,I)=B(12,I)
      B(12,I)=B(13,I)
      B(13,I)=B(14,I)
  160 B(14,I)=YBAR(I)
  170 CONTINUE
C RESET STORED VALUES OF COEFFICIENTS IN PREPARATION FOR
C DOUBLED STEP SIZE.
      COEFF1=COEFF1+COEFF1
      COEFF2=COEFF2+COEFF2
      COEFF6=COEFF6+COEFF6
C RESET STORED VALUES OF Y(I),YBAR(I),and PMNUSC(I) IN
C PREPARATION FOR DOUBLED STEP SIZE.
      DO 180 I=1,6
      PMNUSC(I)=8.962962962962963D+00*(Y(I)
     1          -B(1,I))-COEFF6*(YBAR(I)+B(8,I)
     2          +3.0D+00*(B(12,I)+B(10,I)))
      B(2,I)=B(3,I)
      B(3,I)=B(5,I)
      B(4,I)=B(7,I)
      B(9,I)=B(10,I)
      B(10,I)=B(12,I)
  180 B(11,I)=B(14,I)
```

FIG. 4a. FORTRAN IV program segment in which the predictor-corrector portion of the integration from below the Moho to r_0 is handled. Most of the computation time is spent in this segment. The boxed segments refer to the basic AJP matrix multiplication illustrated in Figure 3.

(Figure 4b. First of two pages)

```
      DIMENSION B(6,14),Y(6),YBAR(6),XM(6),XMBAR(6),P(6),C(6),PMNUSC(6),
     1          A(2090)
      EQUIVALENCE (Y(1),Y1)
      EQUIVALENCE (XM(1),XM1)
      EQUIVALENCE (XMBAR(1),XMBAR1)
      IPT=-18

           .
           .
           .

C LOOP OVER DEPTH IN CURRENT STEP-SIZE REGION.
      DO 170 N=NSTART,NSTOP
      IF(IT.EQ.4)  GO TO 115
      IT=NTEMP-4
  115 DO 120 I=1,6
C SET PREDICTOR  P(I)  .
      P(I)=B(I,IT)+COEFF1*(2.0D+00*(B(I,IT+10)+B(I,IT+8))-B(I,IT+9))
C SET MODIFIED PREDICTOR  XM(I).
  120 XM(I)=P(I)-.9256198347107438D+00*PMNUSC(I)
      IPT=IPT+19
      XMBAR(1)=A(IPT)*XM(1)+A(IPT+1)*XM(2)+A(IPT+2)*XM(3)
      XMBAR(2)=A(IPT+3)*XM(1)+A(IPT+4)*XM(2)+A(IPT+5)*XM(3)
     1         +A(IPT+6)*XM(4)+A(IPT+7)*XM(6)
      XMBAR(3)=A(IPT+8)*(XM(3)-XM(1))+A(IPT+9)*XM(4)
      XMBAR(4)=A(IPT+10)*XM(1)+A(IPT+11)*XM(2)+A(IPT+12)*XM(3)
     1         +A(IPT+13)*XM(4)+A(IPT+14)*XM(5)
      XMBAR(5)=A(IPT+15)*XM(1)+XM(6)
      XMBAR(6)=A(IPT+16)*XM(3)+A(IPT+17)*XM(5)+A(IPT+18)*XM(6)
      DO 130 I=1,6
C SET CORRECTOR  C(I)  .
      C(I)=.125D+00*(9.0D+00*B(I,IT+3)-B(I,IT+1)+COEFF2*(XMBAR(I)+2.0D+0
     1     *B(I,IT+10)-B(I,IT+9)))
      PMNUSC(I)=P(I)-C(I)
C SET SOLUTION VECTOR AT NTH DEPTH.
  130 Y(I)=C(I)+.07438016528925620D+00*PMNUSC(I)
      IF(N.EQ.NEND)  RETURN
C SET DERIVATIVE OF SOLUTION VECTOR AT NTH DEPTH.
      YBAR(1)=A(IPT)*Y(1)+A(IPT+1)*Y(2)+A(IPT+2)*Y(3)
      YBAR(2)=A(IPT+3)*Y(1)+A(IPT+4)*Y(2)+A(IPT+5)*Y(3)
     1         +A(IPT+6)*Y(4)+A(IPT+7)*Y(6)
      YBAR(3)=A(IPT+8)*(Y(3)-Y(1))+A(IPT+9)*Y(4)
      YBAR(4)=A(IPT+10)*Y(1)+A(IPT+11)*Y(2)+A(IPT+12)*Y(3)
     1         +A(IPT+13)*Y(4)+A(IPT+14)*Y(5)
      YBAR(5)=A(IPT+15)*Y(1)+Y(6)
      YBAR(6)=A(IPT+16)*Y(3)+A(IPT+17)*Y(5)+A(IPT+18)*Y(6)
      IF(NTEMP.GT.7)  GO TO 150
      DO 140 I=1,6
      B(I,NTEMP)=Y(I)
  140 B(I,NTEMP+7)=YBAR(I)
      NTEMP=NTEMP+1
      GO TO 170
```

(Figure 4b. Second of two pages)

```
C RESET STORED VALUES OF  Y  AND  YBAR  IN PREPARATION FOR NEXT INTEGRATION
C STEP.
  150 DO 160 I=1,6
      B(I,1)=B(I,2)
      B(I,2)=B(I,3)
      R(I,3)=B(I,4)
      B(I,4)=B(I,5)
      B(I,5)=B(I,6)
      B(I,6)=B(I,7)
      B(I,7)=Y(I)
      B(I,8)=B(I,9)
      B(I,9)=B(I,10)
      B(I,10)=B(I,11)
      B(I,11)=B(I,12)
      B(I,12)=B(I,13)
      B(I,13)=B(I,14)
  160 B(I,14)=YBAR(I)
  170 CONTINUE
```

FIG. 4b. FORTRAN IV program segment demonstrating subscripting and storage improvements, relative to the segment in Figure 4a, that are required to optimize computation time on an IBM 360 computer.

(Figure 4c. First of two pages)

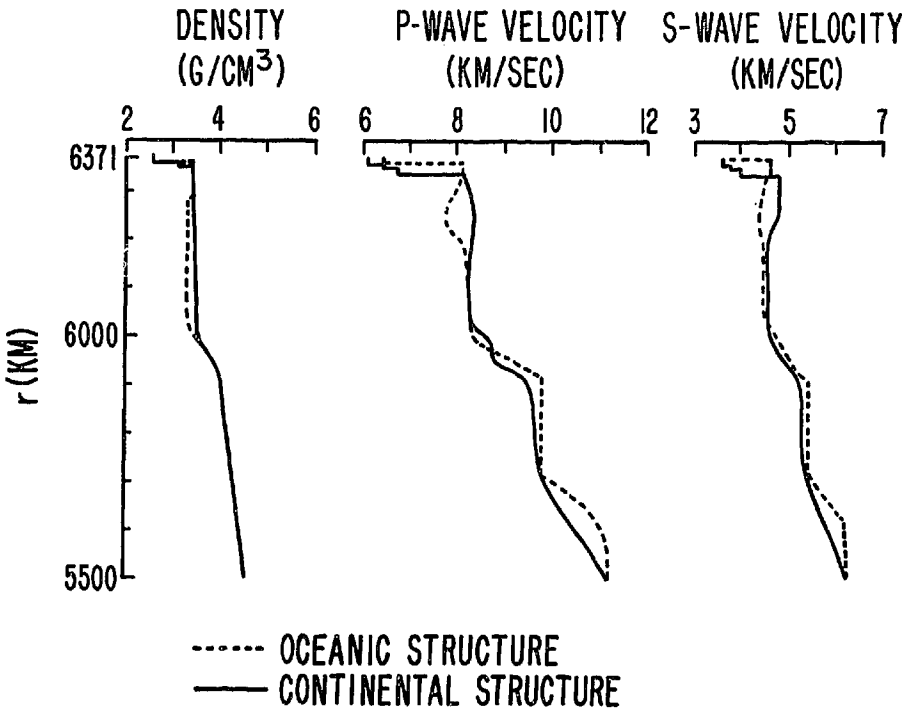
```
      DIMENSION B(4,14),D(4,14),X(4),Y(4),XBAR(4),YBAR(4),XM(4),YM(4),
     1          XMBAR(4),YMBAR(4),P(4),Q(4),C(4),F(4),PMNUSC(4),
     2          QMNUSF(4),A(1430)
      EQUIVALENCE (X(1),X1)
      EQUIVALENCE (Y(1),Y1)
      EQUIVALENCE (XM(1),XM1)
      EQUIVALENCE (YM(1),YM1)
      EQUIVALENCE (XMBAR(1),XMBAR1)
      EQUIVALENCE (YMBAR(1)YMBAR1)
      IPT=-12

           .
           .
           .

C LOOP OVER DEPTH IN CURRENT-STEP-SIZE REGION.
      DO 170 N=NSTART,NSTOP
      IF(IT.EQ.4)  GO TO 115
      IT=NTEMP-4
  115 DO 120 I=1,4
C SET PREDICTORS  P(I)  ,  Q(I) .
      P(I)=B(I,IT)+COEFF1*(2.0D+00*(B(I,IT+10)+B(I,IT+8))-B(I,IT+9))
      Q(I)=D(I,IT)+COEFF1*(2.0D+00*(D(I,IT+10)+D(I,IT+8))-D(I,IT+9))
C SET MODIFIED PREDICTORS  XM(I)  ,  YM(I)  .
      XM(I)=P(I)-.9256198347107438D+00*PMNUSC(I)
  120 YM(I)=Q(I)-.9256198347107438D+00*QMNUSF(I)
      IPT=IPT+13
      XMBAR(1)=A(IPT)*XM(1)+A(IPT+1)*XM(2)+A(IPT+2)*XM(3)
      YMBAR(1)=A(IPT)*YM(1)+A(IPT+1)*YM(2)+A(IPT+2)*YM(3)
      XMBAR(2)=A(IPT+3)*XM(1)+A(IPT+4)*XM(2)+A(IPT+5)*XM(3)
     1         +A(IPT+6)*XM(4)
      YMBAR(2)=A(IPT+3)*YM(1)+A(IPT+4)*YM(2)+A(IPT+5)*YM(3)
     1         +A(IPT+6)*YM(4)
      XMBAR(3)=A(IPT+7)*(XM(3)-XM(1))+A(IPT+8)*XM(4)
      YMBAR(3)=A(IPT+7)*(YM(3)-YM(1))+A(IPT+8)*YM(4)
      XMBAR(4)=A(IPT+9)*XM(1)+A(IPT+10)*XM(2)+A(IPT+11)*XM(3)
     1         +A(IPT+12)*XM(4)
      YMBAR(4)=A(IPT+9)*YM(1)+A(IPT+10)*YM(2)+A(IPT+11)*YM(3)
     1         +A(IPT+12)*YM(4)
      DO 130 I=1,4
C SET CORRECTORS  C(I)  ,  F(I)  .
      C(I)=.125D+00*(9.0D+00*B(I,IT+3)-B(I,IT+1)+COEFF2*(XMBAR(I)+2.0D+0
     1     *B(I,IT+10)-B(I,IT+9)))
      F(I)=.125D+00*(9.0D+00*D(I,IT+3)-D(I,IT+1)+COEFF2*(YMBAR(I)+2.0D+0
     1     *D(I,IT+10)-D(I,IT+9)))
      PMNUSC(I)=P(I)-C(I)
      QMNUSF(I)=Q(I)-F(I)
C SET SOLUTION VECTORS AT NTH DEPTH.
      X(I)=C(I)+.07438016528925620D+00*PMNUSC(I)
  130 Y(I)=F(I)+.07438016528925620D+00*QMNUSF(I)
      IF(N.EQ.NEND)  RETURN
```

(Figure 4c. Second of two pages)

```
C SET DERIVATIVES OF SOLUTION VECTORS AT NTH DEPTH.
      XBAR(1)=A(IPT)*X(1)+A(IPT+1)*X(2)+A(IPT+2)*X(3)
      YBAR(1)=A(IPT)*Y(1)+A(IPT+1)*Y(2)+A(IPT+2)*Y(3)
      XBAR(2)=A(IPT+3)*X(1)+A(IPT+4)*X(2)+A(IPT+5)*X(3)
     1        +A(IPT+6)*X(4)
      YBAR(2)=A(IPT+3)*Y(1)+A(IPT+4)*Y(2)+A(IPT+5)*Y(3)
     1        +A(IPT+6)*Y(4)
      XBAR(3)=A(IPT+7)*(X(3)-X(1))+A(IPT+8)*X(4)
      YBAR(3)=A(IPT+7)*(Y(3)-Y(1))+A(IPT+8)*Y(4)
      XBAR(4)=A(IPT+9)*X(1)+A(IPT+10)*X(2)+A(IPT+11)*X(3)
     1        +A(IPT+12)*X(4)
      YBAR(4)=A(IPT+9)*Y(1)+A(IPT+10)*Y(2)+A(IPT+11)*Y(3)
     1        +A(IPT+12)*Y(4)
      IF(NTEMP.GT.7)  GO TO 150
      DO 140 I=1,4
      B(I,NTEMP)=X(I)
      D(I,NTEMP)=Y(I)
      B(I,NTEMP+7)=XBAR(I)
  140 D(I,NTEMP+7)=YBAR(I)
      NTEMP=NTEMP+1
      GO TO 170
C RESET STORED VALUES OF  X ,  Y  AND  XBAR  ,  YBAR  IN PREPARATION FOR NEXT
C INTEGRATION STEP.
  150 DO 160 I=1,4
      B(I,1)=B(I,2)
      B(I,2)=B(I,3)
      B(I,3)=B(I,4)
      B(I,4)=B(I,5)
      B(I,5)=B(I,6)
      B(I,6)=B(I,7)
      B(I,7)=X(I)
      B(I,8)=B(I,9)
      B(I,9)=B(I,10)
      B(I,10)=B(I,11)
      B(I,11)=B(I,12)
      B(I,12)=B(I,13)
      B(I,13)=B(I,14)
      B(I,14)=XBAR(I)
      D(I,1)=D(I,2)
      D(I,2)=D(I,3)
      D(I,3)=D(I,4)
      D(I,4)=D(I,5)
      D(I,5)=D(I,6)
      D(I,6)=D(I,7)
      D(I,7)=Y(I)
      D(I,8)=D(I,9)
      D(I,9)=D(I,10)
      D(I,10)=D(I,11)
      D(I,11)=D(I,12)
      D(I,12)=D(I,13)
      D(I,13)=D(I,14)
  160 D(I,14)=YBAR(I)
  170 CONTINUE
```

FIG. 4c. FORTRAN IV program segment illustrating our final optimization, from the second section of the main text, for the non-gravitating case. Further improvement is discussed in the fourth section of the main text and in Section 7 of the additional notes.

(Figure 5. First of two pages)



(Figure 5. Second of two pages)

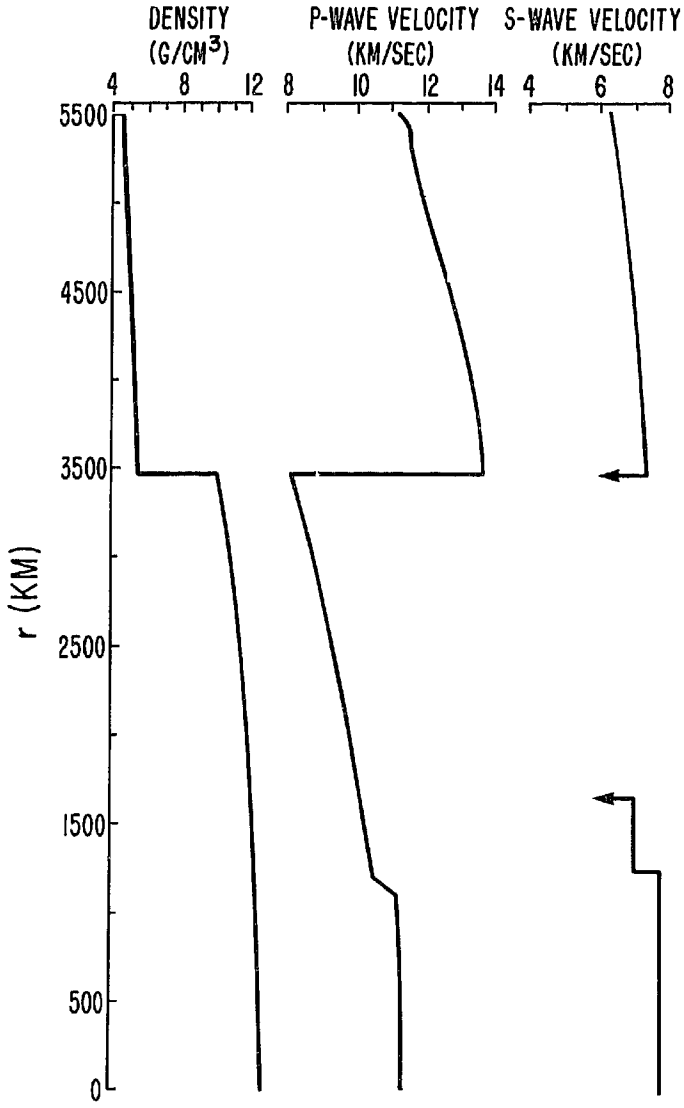


FIG. 5. Oceanic and continental (shield) models used in program testing.

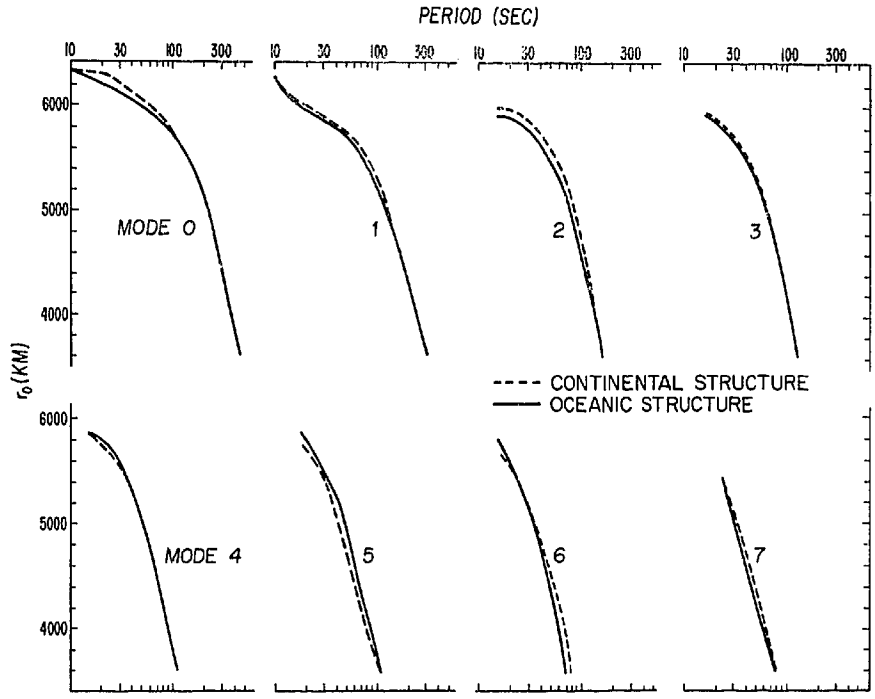


FIG. 6. Values of r_0, the depth at which integration is terminated, which yield 4-significant-figure accuracy in the computed values of c with the optimized version of the basic AJP formulation. At each period, 4-figure accuracy is attained only if r_0 is specified to be smaller than the value indicated by the curve.

```
C NORMALIZATION.
      AMXINV=1.0D+00/DMAX1(DABS(B(7,1)),DABS(B(7,2)),
     1                    DABS(B(7,3)),DABS(B(7,4)),
     2                    DABS(B(7,5)),DABS(B(7,6)))
      DO 165 I=1,6
      PMNUSC(I)=PMNUSC(I)*AMXINV
      Y(I)=Y(I)*AMXINV
      YBAR(I)=YBAR(I)*AMXINV
      DO 165 J=1,14
  165 B(J,I)=B(J,I)*AMXINV
```

FIG. 7. Normalization scheme appropriate to program segment in Figure 4a. The procedure should be included between statement numbers 160 and 170 in Figure 4a. See text for warnings concerning loss of efficiency when this general form of normalization is employed.

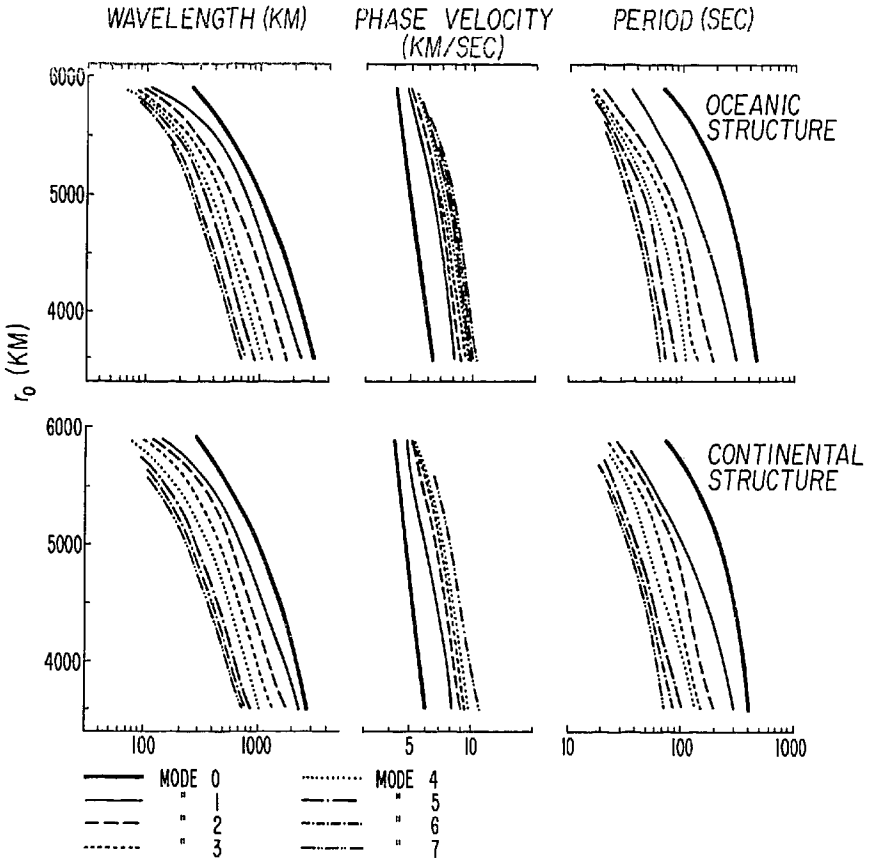


FIG. 8. Collected results from multimode tests for determining maximum permissible values of r_0, when the computed phase velocities are to be accurate to 4 significant figures.

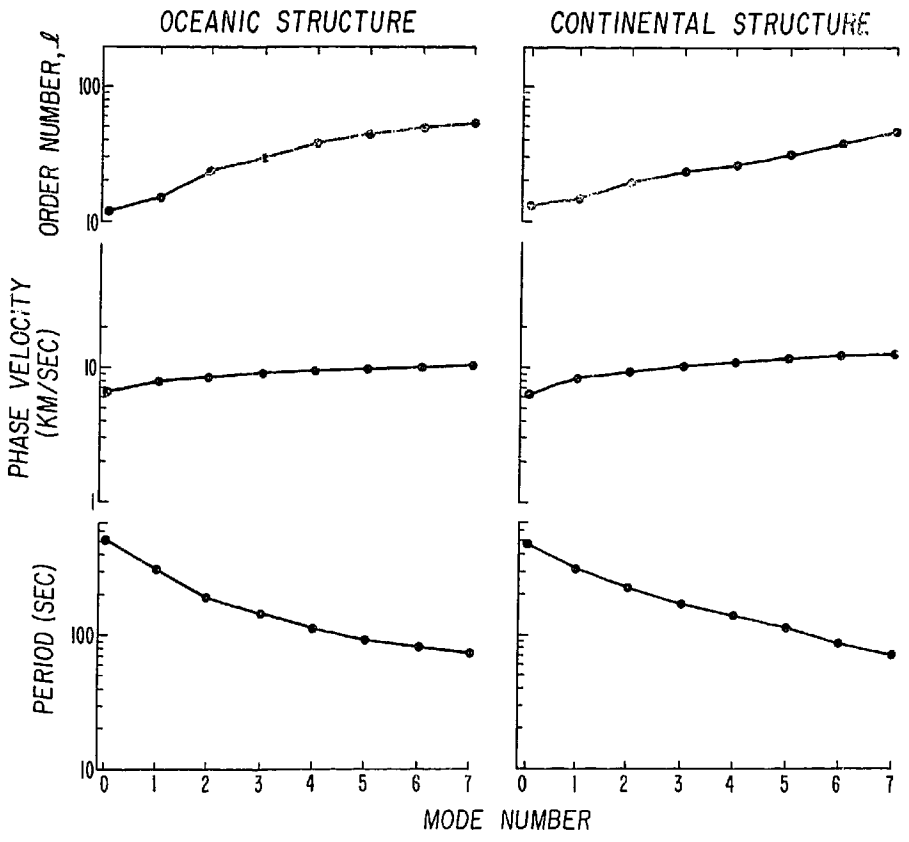
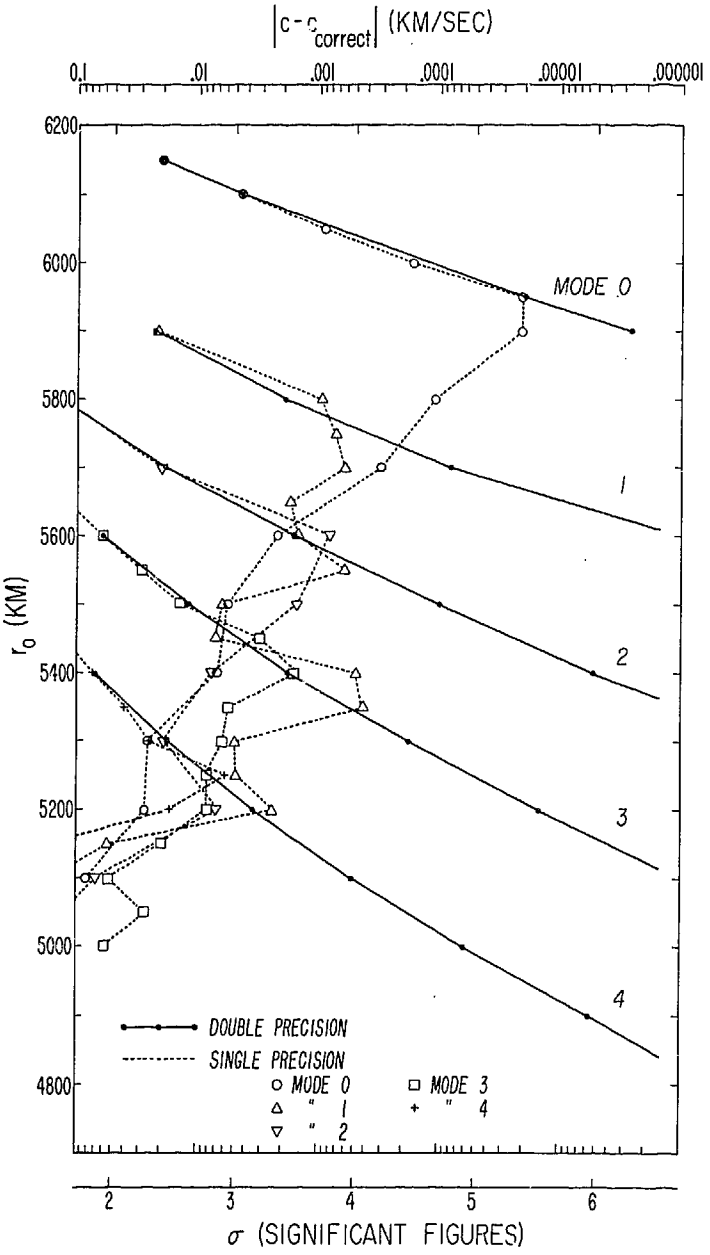


FIG. 9. Maximum periods, maximum phase velocities, and minimum polar order numbers ℓ, that can be used if c is desired to 4-figure accuracy, when the integration is limited to the mantle and the core is excluded from the computations (other than for use in the determination of g(r)).

```
150 DO 160 I=1,6
    B(1,I)=SNGL(B(2,I))
    B(2,I)=SNGL(B(3,I))
    B(3,I)=SNGL(B(4,I))
    B(4,I)=SNGL(B(5,I))
    B(5,I)=SNGL(B(6,I))
    B(6,I)=SNGL(B(7,I))
    B(7,I)=SNGL(Y(I))
    B(8,I)=SNGL(B(9 ,I))
    B(9,I)=SNGL(B(10,I))
    B(10,I)=SNGL(B(11,I))
    B(11,I)=SNGL(B(12,I))
    B(12,I)=SNGL(B(13,I))
    B(13,I)=SNGL(B(14,I))
160 B(14,I)=SNGL(YBAR(I))
```

FIG. 10. FORTRAN IV program segment used to simulate single-precision computations when using our double-precision optimization of the basic AJP formulation. The function SNGL accepts a double-precision argument, and returns the single-precision equivalent. This program segment is used to replace DO-loop 160 in Figure 4a.

(Figure 11. First of two pages)



(Figure 11. Second of two pages)

FIG. 11. Results of testing the effect of reducing IBM 360 computations from double (16-17 decimal digits) to single precision (about 6 decimal digits), while keeping the period fixed at 50 seconds. For a given radial mode, in order to obtain σ significant figures in the computed phase velocity, r_0 must not exceed the value given by the upper portion of the dashed line (structural limitation) nor fall below the lower portion (loss-of-precision limitation).

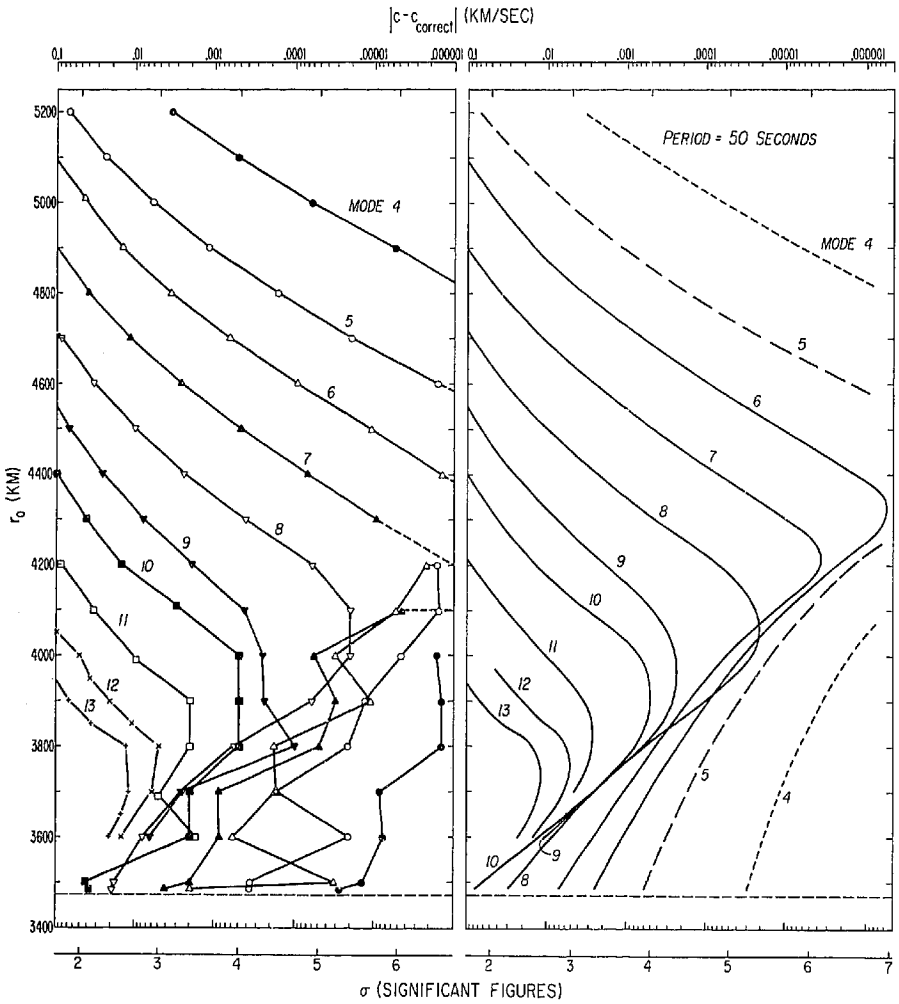


FIG. 12. Results of IBM 360, double-precision tests of loss-of-precision problem at a period of 50 seconds. At left are shown the raw results of our structure-reduction experiments; at right, the corresponding smoothed curves for each mode. The latter curves are drawn such that all data points, for a given radial mode, fall to the right of the corresponding smoothed curve.

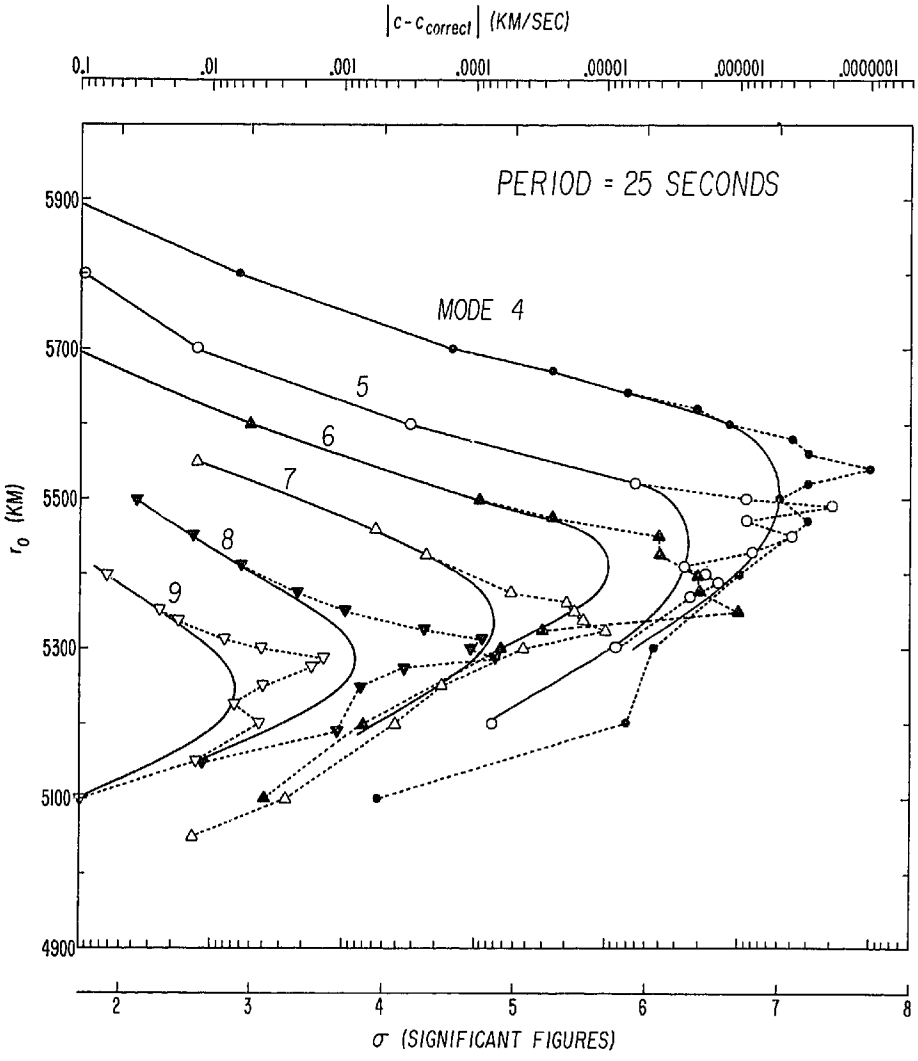


FIG. 13. Results of IBM 360, double-precision tests of loss-of-precision problem at a period of 25 seconds.

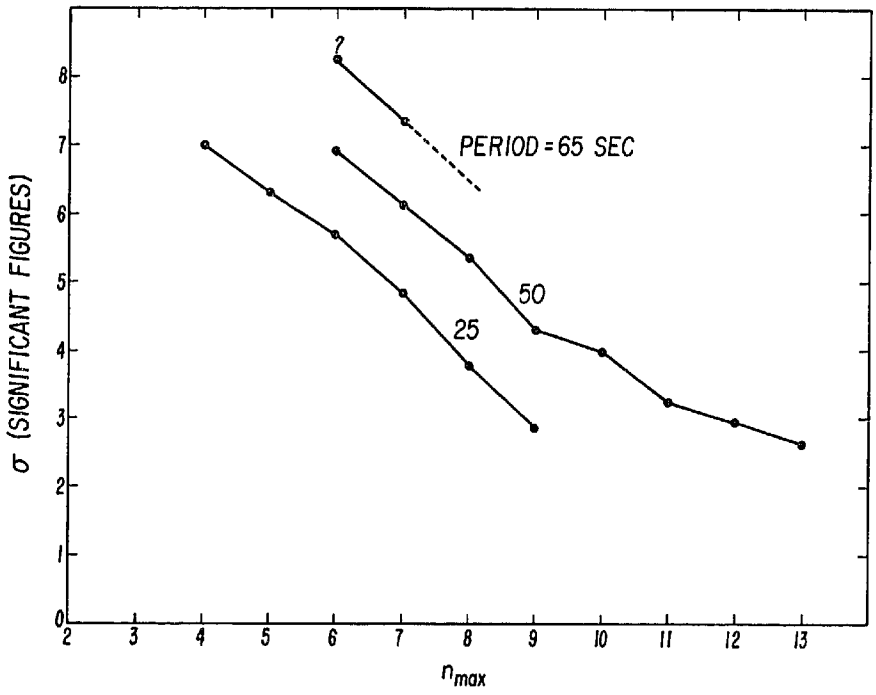


FIG. 14. Maximum possible (radial) mode number, n_{max}, for which σ significant figures can be obtained with our optimization (for speed of computation) of the AJP formulation. This limitation of the original AJP formulation is due to the loss-of-precision problem.

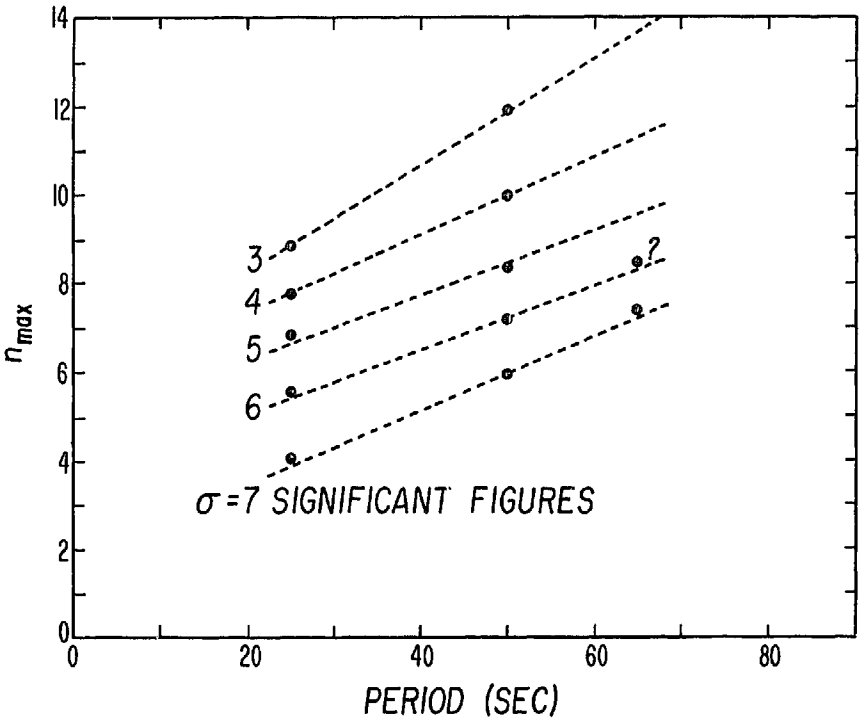


FIG. 15. Relationship between maximum possible (radial) mode number, n_{max}, and minimum period, for several values of σ. This limitation of the original AJP formulation is due to the loss-of-precision problem.

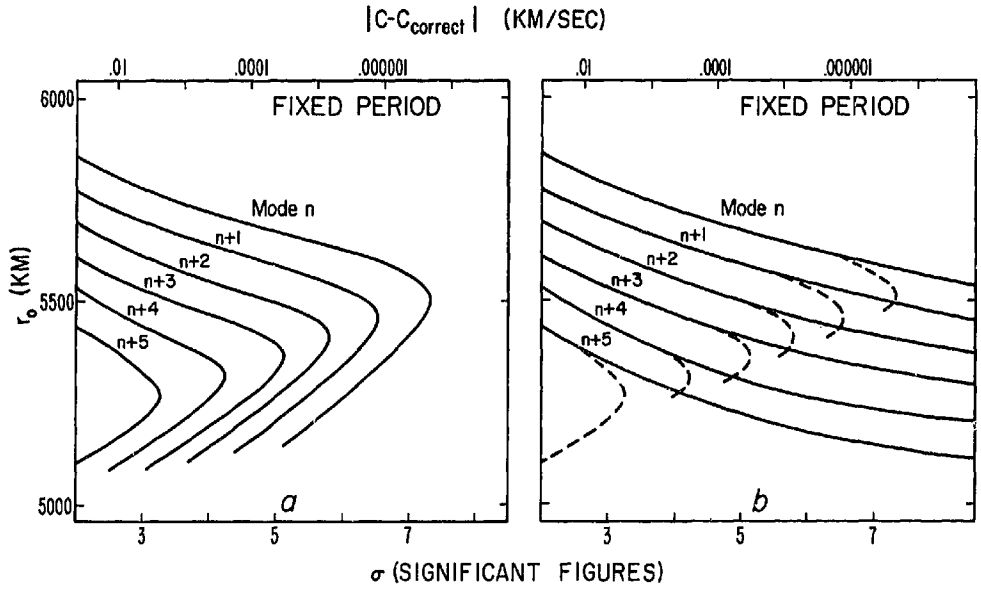
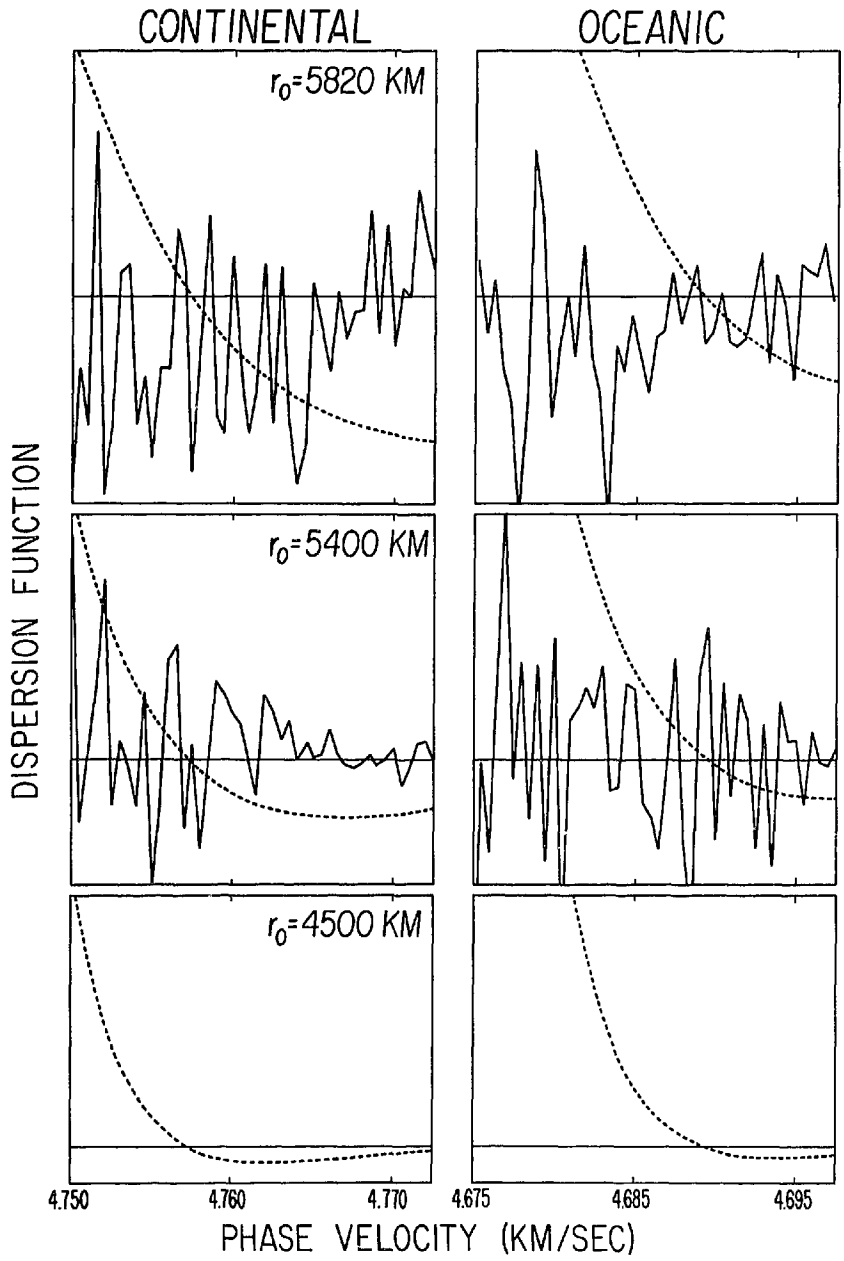


FIG. 16. Schematic illustration of results of numerical-accuracy tests applied to dispersion computations. Numerical integration is performed between the surface of the earth and radius r_0, n is the radial mode number, and c is the computed phase velocity. Insert at left represents the situation when the algorithm contains loss-of-precision problems; right-hand insert (solid lines) illustrates the situation when these problems are absent. Further details are given in second section of main text (Existence of Solutions as a Function of Numerical and Algorithmic Procedures; Figures 11-15).

(Figure 17. First of two pages)



(Figure 17. Second of two pages)

FIG. 17. Representation of detailed behavior of dispersion function when precision loss dominates the original AJP formulation for the computation of Rayleigh-wave dispersion on a spherical, gravitating model of the earth. Results are for the fourth radial mode at a period of 10 seconds, for both continental and oceanic structures; effect of decreasing r_0 is also included. Solid lines denote the results when a feature for controlling precision loss is not included in the computational algorithm. In this case only the points are significant; the continuous lines have only the purpose of connecting computed results. Dashed lines illustrate results obtained when orthogonalization is used to control precision loss. In this case, note both the stability of the solution--position of the root of the dispersion function--and the smoothness maintained by this function as r_0 is decreased by 1320 km.

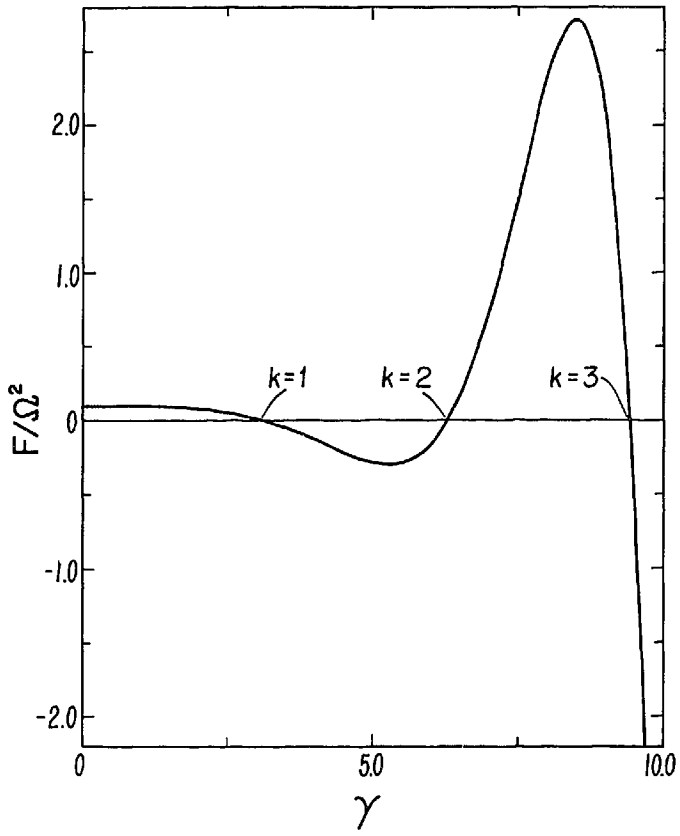


FIG. 18. Dispersion function for the case of a horizontal, homogeneous beam that is rigidly supported at its two ends. See Section 4 of the additional notes for further details.

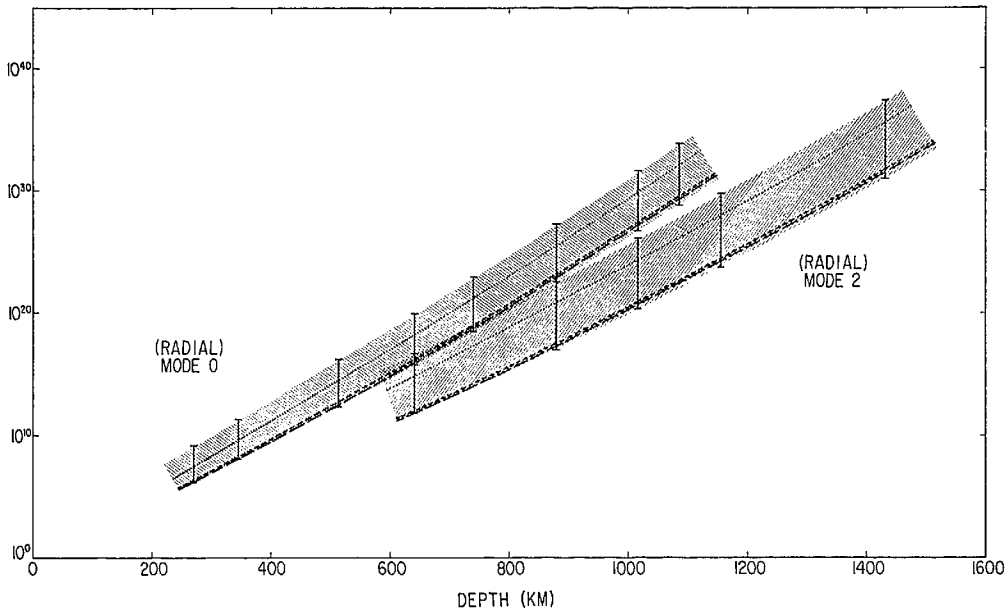


FIG. 19. Results of numerical tests comparing the overflow properties when integrating the vectors X_i over depth in the downward direction (dotted and dashed lines), and in the upward direction (hatched areas). A realistic, continental model of the earth was used in these tests of two radial modes, and a period of 25 seconds was employed.

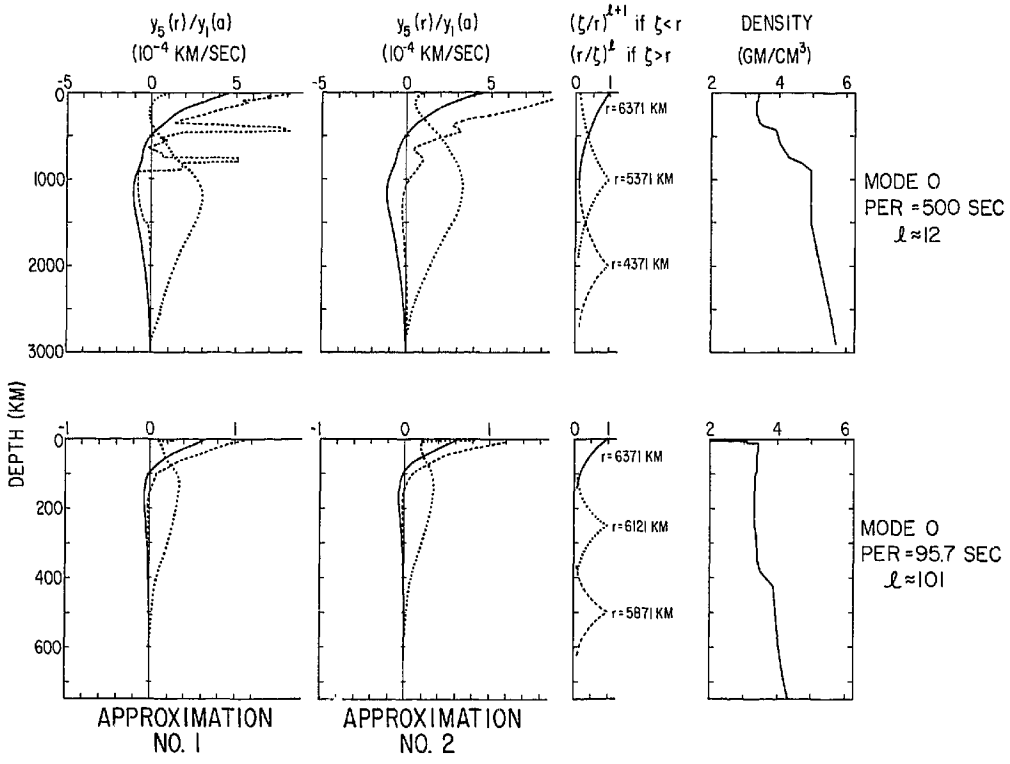


FIG. 20. Results of numerical tests of "approximation no. 1" and "approximation no. 2" in forming (4.3) and (4.4). In the representations of $y_5(r)/y_1(a)$, the true values are illustrated by solid lines, the dotted lines denote the computed results when one term of the Taylor's series expansions is used, and the dashed lines, the computed results for two terms. See Table 4 for structure used to model the earth.

Enter procedure with specified angular frequency, ω, the order number at which to start the root-bracketting (of dispersion function) procedure, $\ell_0(\omega)$, and step size, $\Delta\ell$.

1. Set $\ell = \ell_0(\omega)$.
2. Run algorithm based on (4.10).
3. Set $\ell = \ell - \Delta\ell$.
4. Set $\ell = \ell + \Delta\ell$.
5. Determine eigenfunctions $py_i(r)$.
6. Evaluate integrals $I_i(r)$ using (4.12).
7. Run algorithm based on (4.11) and (4.12)* to determine dispersion-function value.
8. If root of dispersion function has been bracketted, proceed; otherwise, return to step 4.
9. Interpolate for value of ℓ yielding a root of the dispersion function.
10. Repeat steps 5, 6, and 7.
11. If sufficient accuracy has been obtained for ℓ, computation is complete; otherwise, return to step 9.

* See "approximation no. 5."

FIG. 21. Outline of iterative procedure for improving "approximation no. 2," which is based on relation (4.10).

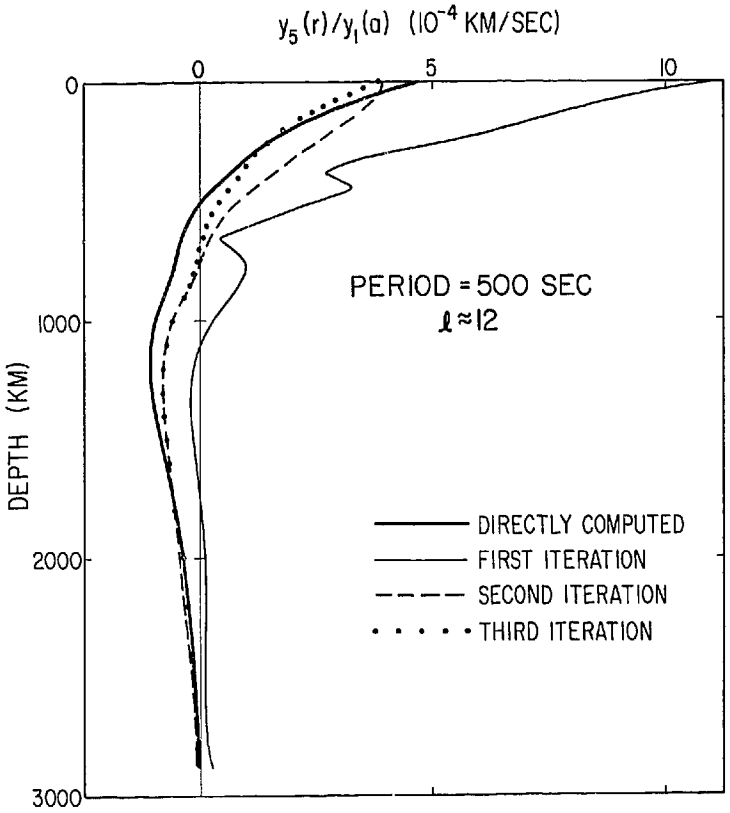


FIG. 22. Results of numerical tests of procedure outlined in Figure 21 for improving "approximation no. 2."

Enter procedure with specified angular frequency, ω, and accurate estimates (from data at previous frequencies) of ℓ and $py_i(r)$ at ω.

1. Evaluate integrals $I_i(r)$ using (4.12).
2. If this is the first iteration, set ℓ to its predicted value; otherwise, interpolate for value of ℓ yielding a root of the dispersion function.
3. Run algorithm based on (4.11) and (4.12).
4. If sufficient accuracy has been obtained for ℓ, go to step 5; otherwise, return to step 2.
5. Determine eigenfunctions $y_i(r)$.
6. Predict ℓ and $py_i(r)$ at next frequency to be treated.

FIG. 23. Outline of high-speed procedure to be used with the fourth-order simplification of the AJP formulation for Rayleigh-wave dispersion computations with a spherical, gravitating model of the earth.

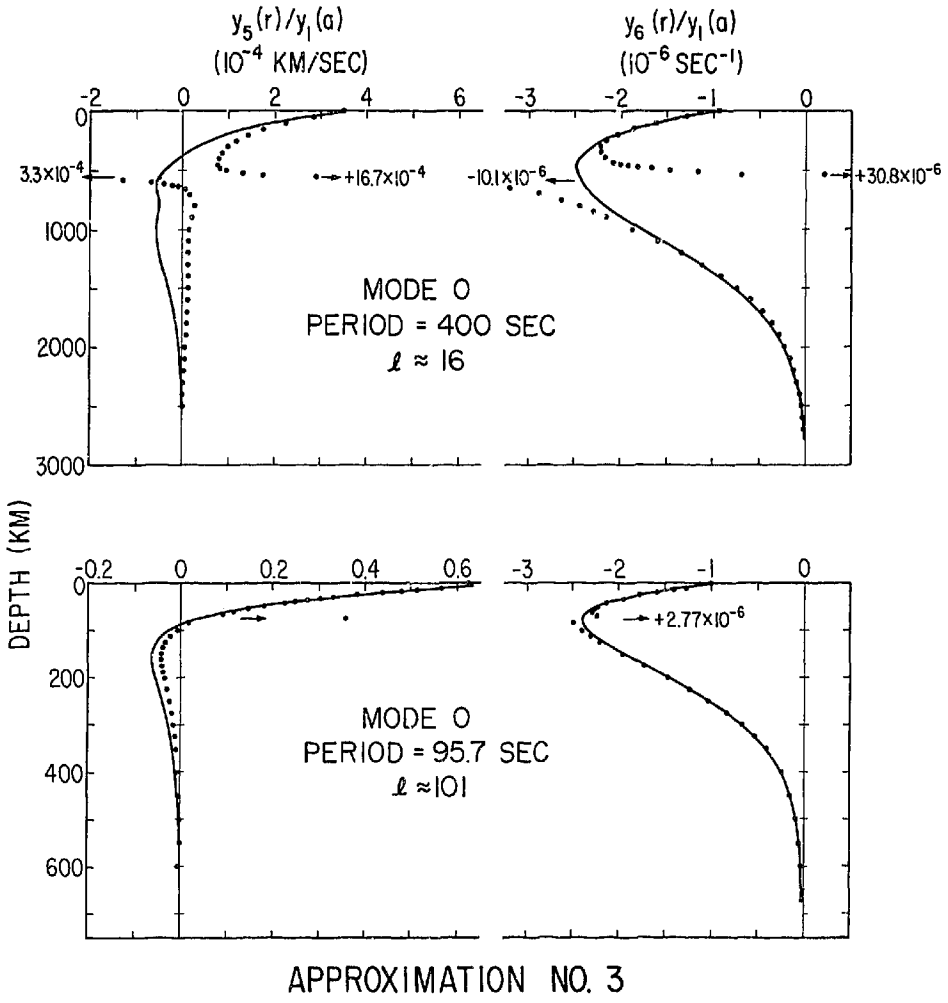


FIG. 24. Results of numerical testing of "approximation no. 3." Predictions $py_1(r)$ and $py_3(r)$ are obtained from the (single) frequency 0.0005 cps below each of those represented in the figure. Solid lines denote true depth dependences; plotted points indicate test results. See Table 4 for structure used to model the earth.

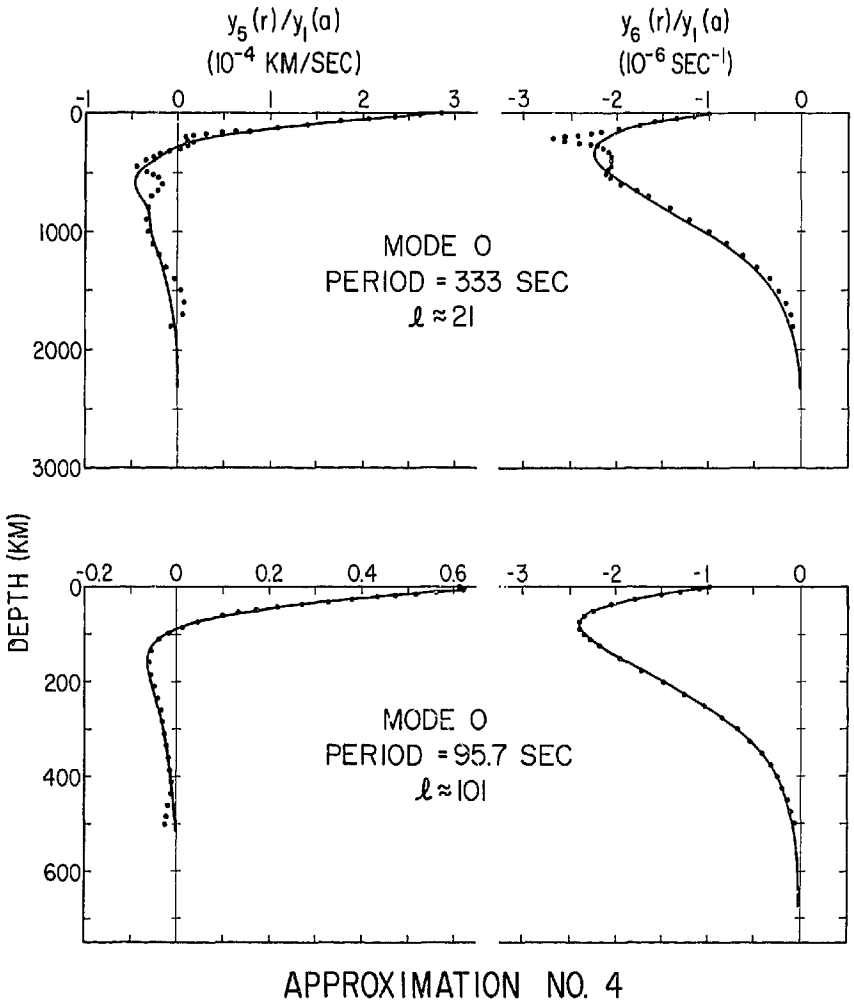


FIG. 25. Results of numerical testing of "approximation no. 4." Predictions $py_1(r)$ and $py_3(r)$ are obtained from the pair of frequencies 0.0005 and 0.0010 cps below each of those represented in the figure. See caption of Figure 24 for further details.

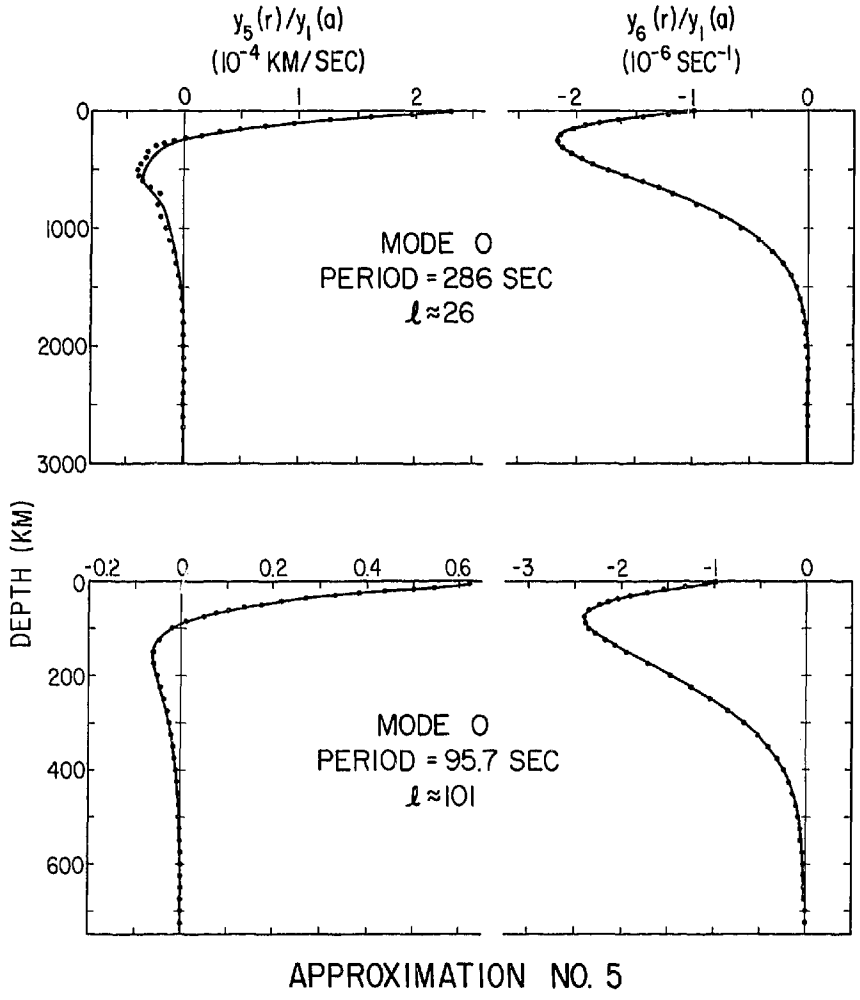


FIG. 26. Results of numerical testing of "approximation no. 5." Predictions $py_1(r)$ and $py_3(r)$ are obtained from the three frequencies 0.0005, 0.0010, and 0.0015 cps below each of those represented in the figure. See caption of Figure 24 for further details.

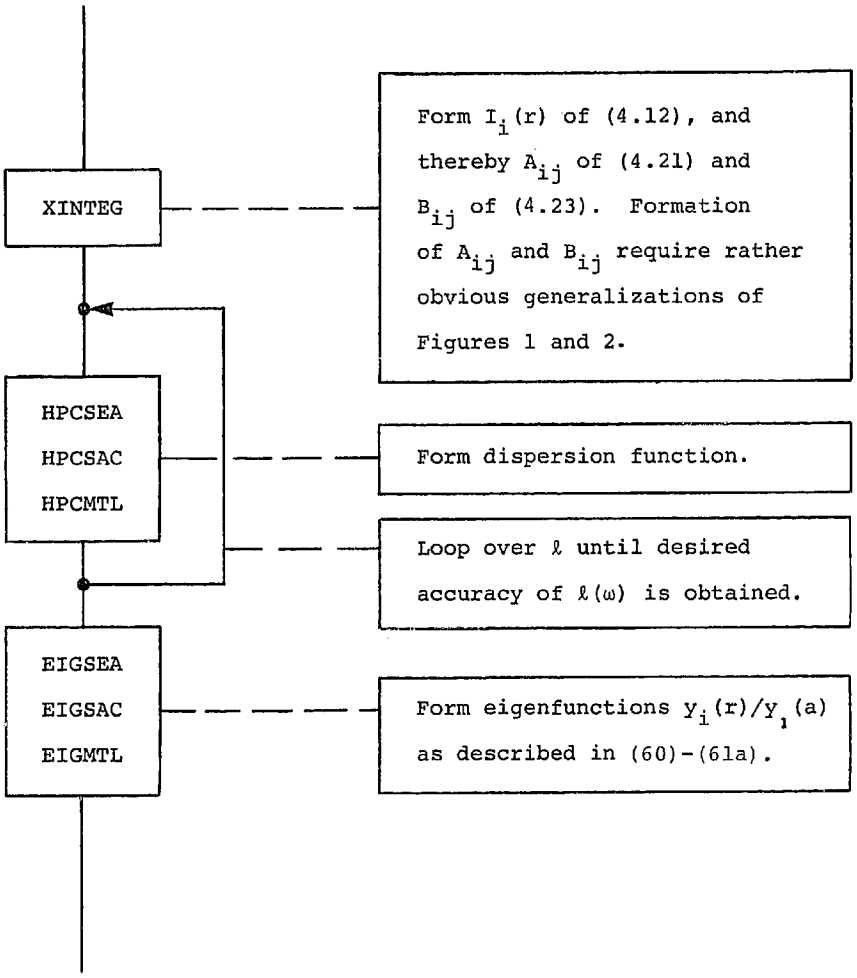
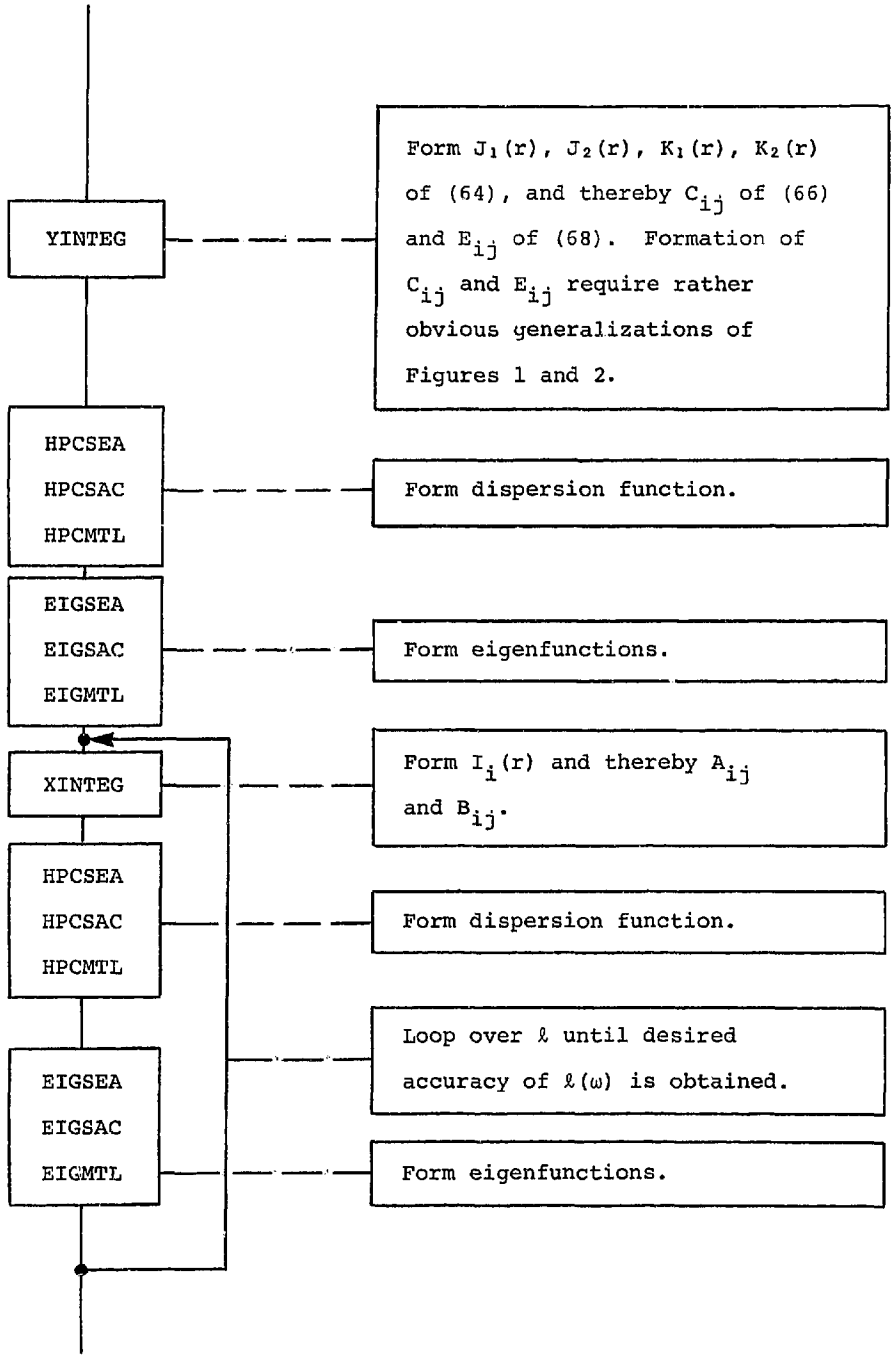


FIG. 27. Subroutine structure of high-speed procedure outlined in Figure 23. Subroutine HPCSEA performs integration of (4.22), taking Z(a) down to $Z(r_1)$; HPCSAC integrates (4.20), taking $X_i(a)$, or $X_i(r_1)$, down through the sediments and subsedimentary crustal layers; HPCMTL then carries the integration of X_i from the base of the crust, through the mantle, to r_0. Using the results $X_i(r_0)$, the corresponding eigenfunction determinations of (60)-(61a) are then made by integrating a single vector through the oceanic layer (EIGSEA), sediments and subsedimentary crustal layers (EIGSAC), and mantle (EIGMTL).

(Figure 28. First of two pages)



(Figure 28. Second of two pages)

FIG. 28. Somewhat expanded version of key part of Figure 21, showing more detailed outline of fourth-order, "starting" procedure. Note that the subroutines used to form the dispersion function--HPCSEA, HPCSAC, HPCMTL--and the subroutines used to form eigenfunctions--EIGSEA, EIGSAC, EIGMTL--are exactly the same routines that are used in the high-speed procedure of Figures 23 and 27 (see caption of Figure 27 for more details concerning the subroutines). This is possible because (1) the key optimizing technique is the evaluation of the coefficient-matrix elements prior to entering these subroutines, and (2) these matrix multiplications have the same form in both the "starting" and high-speed procedures. For example, both (4.20) and (65) use the same multiplication scheme for $[X_1(r)]_i$ and $[X_2(r)]_i$ in HPCSAC and HPCMTL: evaluation of XBAR(I) and YBAR(I) following statement number 130 in Figure 4c. The only new subroutine required by the "starting" procedure is YINTEG.

REFERENCES APPEARING ONLY IN ADDITIONAL NOTES

Backus, G. and F. Gilbert (1967). Numerical applications of a formalism to geophysical inverse problems, Geophys. J. 13, 247-276.

Calcagnile, G., G.F. Panza, F. Schwab and E.G. Kausel (1976). On the computation of theoretical seismograms for multimode surface waves, Geophys. J. 47, 73-81.

Chapman, C.H. and R.A. Phinney (1972). Diffracted seismic signals and their numerical solution, in Methods in Computational Physics, Volume 12, ed. B.A. Bolt, Academic Press, New York, 165-230.

Crossley, D.J. (1975). The free-oscillation equations at the centre of the Earth, Geophys. J. 41, 153-163.

Dunkin, J.W. (1965). Computation of modal solutions in layered, elastic media at high frequencies, Bull. Seism. Soc. Am. 55, 335-358.

Gilbert, F. and G. Backus (1969). A computational problem encountered in a study of the earth's normal modes, AFIPS Conf. Proc. 33, 1273-1277.

Kausel, E.G., F. Schwab and E. Mantovani (1977). Oceanic Sa, Geophys. J. 50, 407-440.

Knopoff, L., F. Schwab and E. Kausel (1973). Interpretation of Lg, Geophys. J. 33, 389-404.

Knopoff, L., F. Schwab, K. Nakanishi and F. Chang (1974). Evaluation of Lg as a discriminant among different continental crustal structures, Geophys. J. 39, 41-70.

Mantovani, E. (1978). Generation of complete theoretical seismograms for SH. Part III, Geophys. J. 53, 55-60.

Mantovani, E., F. Schwab, H. Liao and L. Knopoff (1976). Generation of complete theoretical seismograms for SH. Part II, Geophys. J. 48, 531-536.

Mantovani, E., F. Schwab, H. Liao and L. Knopoff (1977). Teleseismic Sn: A guided wave in the mantle, Geophys. J. 51, 709-726.

Nakanishi, K.K., F. Schwab and E. Kausel (1976). Interpretation of Sa on a continental structure, Geophys. J. 47, 211-223.

Nakanishi, K.K., F. Schwab and L. Knopoff (1976). Generation of complete theoretical seismograms for SH. Part I, Geophys. J. 48, 525-530.

North, R.G. and A.M. Dziewonski (1975). Seismic Discrimination. II. Surface wave studies. B. An empirical earth flattening correction for Rayleigh wave dispersion, in Semiannual Technical Summary Report to the Advanced Research Projects Agency, M.I.T. Lincoln Laboratory, Lexington, Massachusetts.

Panza, G.F., F. Schwab and L. Knopoff (1972). Channel and crustal Rayleigh waves, Geophys. J. 30, 273-280.

Pestel, E.C. and F.A. Leckie (1963). Matrix Methods in Elastomechanics, McGraw-Hill, New York.

Pitteway, M.L.V. (1965). The numerical calculation of wave-fields, reflexion coefficients and polarizations for long radio waves in the lower ionosphere. Part I, Phil. Trans. Roy. Soc. Lond., A 257, 219-241.

Schwab, F., K.K. Nakanishi and G. Liang (1980). Surface-wave computations and the synthesis of theoretical seismograms at high frequencies, in preparation.

Scott, M.R. (1973). Invariant Imbedding and its Applications to Ordinary Differential Equations, Addison-Wesley, Massachusetts.

Shanks, E.B. (1966). Solutions of differential equations by evaluations of functions, Math. Comp. 20, 21-38.

Thrower, E.N. (1965). The computation of dispersion of elastic waves in layered media, J. Sound. Vib. 2, 210-226.

Watson, T.H. (1970). Fast computation of Rayleigh wave dispersion in a layered half space, Bull. Seism. Soc. Am. 60, 161-166.