

# Numerical Methods for the Accurate Calculation of Spherical Bessel Functions and the Location of Mie Resonances

C. D. Cantrell  
Center for Applied Optics  
University of Texas at Dallas  
Richardson, TX 75083-0688

August 2, 1988; references updated 2006

## **Abstract**

This report reviews numerical methods for the accurate calculation of spherical Bessel functions and for the location of Mie resonances in a dielectric sphere.

Detailed numerical results are presented for a real index of refraction  $n = 1.47$ .

FORTRAN source listings are provided for computer programs that implement the algorithms described in the text.

# 1 Introduction

The method of partial-wave analysis of a field scattered by an obstacle appears to have been introduced into physics by Lord Rayleigh,<sup>1</sup> whose theory of the scattering of electromagnetic waves by a cylinder<sup>2</sup> employed the usual boundary conditions to determine the partial-wave amplitudes  $B_m$  via equations such as

$$B_m \left\{ ka \frac{d}{d(ka)} \psi_m(ka) J_m(k'a) - k'a \frac{d}{d(k'a)} J_m(k'a) \psi_m(ka) \right\} \\ = 2i^m \{ k'a J_m(ka) J'_m(k'a) - ka J_m(k'a) J'_m(ka) \}. \quad (1)$$

In (1),  $a$  is the radius of the cylinder,  $\psi_n$  is the solution of a radial equation inside the cylinder and  $J_m$  is the usual Bessel function of the first kind. Rayleigh also commented that slow convergence makes the infinite series for the Bessel and radial functions of large argument unsuitable for computation.

The numerical difficulties associated with the evaluation of Bessel functions of large (and possibly complex) argument are probably nowhere more evident than in the field of Mie scattering. The theory of the scattering of light by a sphere due to Mie<sup>3</sup> depends upon a partial-wave decomposition of the incident ( $\mathbf{E}^{(i)}$ ), scattered ( $\mathbf{E}^{(s)}$ ) and internal ( $\mathbf{E}^{(w)}$ ) waves:

$$\mathbf{E}^{(i)} = E_0 \hat{\mathbf{e}}_\kappa e^{ikz} \\ = E_0 \sum_l \sqrt{2\pi(2l+1)} \\ \times \left\{ \frac{i^{l-1}}{\sqrt{2l+1}} \left[ -\sqrt{l} j_{l+1}(kr) \mathbf{Y}_{l,l+1,\kappa}(\theta, \phi) + \sqrt{l+1} j_{l-1}(kr) \mathbf{Y}_{l,l-1,\kappa}(\theta, \phi) \right] \right. \\ \left. - \kappa i^l j_l(kr) \mathbf{Y}_{l\kappa}(\theta, \phi) \right\} \quad (2)$$

$$\mathbf{E}^{(s)} = E_0 \sqrt{2\pi} \sum_l i^{l-1} \\ \times \left\{ \sqrt{\epsilon} \left[ -\sqrt{l} h_{l+1}^{(1)}(kr) \mathbf{Y}_{l,l+1,m\kappa}(\theta, \phi) + \sqrt{l+1} h_{l-1}^{(1)}(kr) \mathbf{Y}_{l,l-1,\kappa}(\theta, \phi) \right] a_l \right. \\ \left. - i \kappa \sqrt{2l+1} h_l^{(1)}(kr) \mathbf{Y}_{l\kappa}(\theta, \phi) b_l \right\} \quad (3)$$

<sup>1</sup> Lord Rayleigh (J. W. Strutt), *Theory of Sound* (London, Macmillan, 1877; reprinted by Dover, New York, 1945), §§338-341.

<sup>2</sup> Lord Rayleigh (J. W. Strutt), "On the Electromagnetic Theory of Light", *Phil. Mag.* **12**, 81-101 (1881).

<sup>3</sup> G. Mie, "Beiträge zur Optik trüber Medien, speziell kolloider Metallösungen", *Ann. d. Physik* (4) **25**, 377-445 (1908). See also M. Born and E. Wolf, *Principles of Optics*, 6th edition (Oxford, Pergamon Press, 1986), Section 13.5.

$$\begin{aligned}
\mathbf{E}^{(w)} &= E_0 \sqrt{2\pi} \sum_l i^{l-1} \\
&\times \left\{ \sqrt{\epsilon} \left[ -\sqrt{l} j_{l+1}(k'r) \mathbf{Y}_{l,l+1,\kappa}(\theta, \phi) + \sqrt{l+1} j_{l-1}(k'r) \mathbf{Y}_{l,l-1,\kappa}(\theta, \phi) \right] c_l \right. \\
&\quad \left. - i\kappa \sqrt{2l+1} j_l(k'r) \mathbf{Y}_{l\kappa}(\theta, \phi) d_l \right\}
\end{aligned} \tag{4}$$

$$\begin{aligned}
\mathbf{H}^{(w)} &= E_0 \sqrt{2\pi} \sum_l i^{l-1} \\
&\times \left\{ -\kappa \sqrt{\epsilon} \left[ -\sqrt{l} j_{l+1}(k'r) \mathbf{Y}_{l,l+1,\kappa}(\theta, \phi) \right. \right. \\
&\quad \left. \left. + \sqrt{l+1} j_{l-1}(k'r) \mathbf{Y}_{l,l-1,\kappa}(\theta, \phi) \right] d_l \right. \\
&\quad \left. + i\epsilon \sqrt{2l+1} j_l(k'r) \mathbf{Y}_{l\kappa}(\theta, \phi) c_l \right\}
\end{aligned} \tag{5}$$

in which the partial-wave amplitudes  $a_l, \dots, d_l$  are<sup>4</sup>

$$a_l = \left[ D_l^{(e)} \right]^{-1} \left[ j_l(kr) \frac{d}{dr} r j_l(k'r) - \epsilon j_l(k'r) \frac{d}{dr} r j_l(kr) \right]_{r=a} \tag{6}$$

$$b_l = \left[ D_l^{(m)} \right]^{-1} \left[ j_l(kr) \frac{d}{dr} r j_l(k'r) - j_l(k'r) \frac{d}{dr} r j_l(kr) \right]_{r=a} \tag{7}$$

$$c_l = i \left[ ka D_l^{(e)} \right]^{-1} \tag{8}$$

$$d_l = i \left[ ka D_l^{(m)} \right]^{-1} . \tag{9}$$

In (2-9),  $\kappa = \pm 1$  specifies the circular polarization of the incident field;  $a$  is the radius of the sphere;  $k = 2\pi/\lambda$ , where  $\lambda$  is the wavelength of the light;  $k' = \sqrt{\epsilon}k$ , where  $\epsilon$  is the complex dielectric constant;  $a_l$  and  $b_l$  are the amplitudes of the electric (transverse magnetic) and magnetic (transverse electric) modes outside the sphere; and  $c_l$  and  $d_l$  are the amplitudes of the electric (transverse magnetic) and magnetic (transverse electric) modes inside the sphere, respectively. The functions  $\mathbf{Y}_{jlm}$  are vector spherical harmonics.<sup>5</sup> The fields in an infinite dielectric medium may be obtained from (4-5) by setting  $c_l = 1/\sqrt{\epsilon}$  and  $d_l = 1$ .

<sup>4</sup> C. D. Cantrell, "Theory of Nonlinear Optics in Dielectric Spheres. II. Coupled-Partial-Wave Theory of Resonant, Resonantly Pumped Stimulated Brillouin Scattering", *Journal of the Optical Society of America B* **8**, 2158-2180 (1991).

<sup>5</sup> A. R. Edmonds, *Angular Momentum in Quantum Mechanics*, second edition (Princeton University Press, 1960), §5.9.

The denominators in (6–9) are

$$D_l^{(e)} = - \left[ \epsilon j_l(k'r) \frac{d}{dr} r h_l^{(1)}(kr) - h_l^{(1)}(kr) \frac{d}{dr} r j_l(k'r) \right]_{r=a} \quad (10)$$

$$D_l^{(m)} = - \left[ j_l(k'r) \frac{d}{dr} r h_l^{(1)}(kr) - h_l^{(1)}(kr) \frac{d}{dr} r j_l(k'r) \right]_{r=a} . \quad (11)$$

In (2–9),

$$j_l(x) := \sqrt{\frac{\pi}{2x}} J_{l+\frac{1}{2}}(x) \quad (12)$$

is the spherical Bessel function,

$$h_l^{(1)} := \sqrt{\frac{\pi}{2x}} H_{l+\frac{1}{2}}^{(1)}(x) = j_l(x) + iy_l(x) \quad (13)$$

is the spherical Hankel function of the first kind and  $y_n(x)$  is the spherical Neumann function. With the convention that  $e^{ikr}$  represents an outgoing spherically symmetric wave, the function  $h_l^{(1)}$  represents an outgoing wave.

If a denominator (10–11) could vanish, the sphere would support internal and outgoing waves with no incident wave.<sup>6</sup> In fact, the denominators can vanish only for complex values of the size parameter

$$x = ka . \quad (14)$$

The absolute value of the imaginary part of a complex root is the half-width at half maximum of a resonance representing an eigenmode, the finite lifetime of which results from the loss of energy due to radiation. Van de Hulst<sup>7</sup> and Stratton<sup>8</sup> have given careful discussions of the behavior of the resonant values of  $x$  in various limits. While a knowledge of the qualitative behavior of the roots of (10–11) is always useful, it was especially so at a time when numerical computation of a general case for many values of  $x$  was impossible.

It might be thought that the asymptotic formulae for the Bessel functions of large argument would be adequate for calculations of the resonance positions. Stratton's formulae, for example, apply to resonant values of  $x$  such that  $|x| \gg \nu$ . Unfortunately the narrowest and most interesting resonances occur for the intermediate case  $|x| \sim \nu$ , for which the asymptotic formulae converge slowly.

Mie-scattering computations prior to 1957, when van de Hulst summarized the then known results, were far less detailed than the calculations that have

<sup>6</sup> P. Debye, "Der Lichtdruck auf Kugeln von beliebigem Material", *Ann. Physik* (4) **30**, 57–136 (1909).

<sup>7</sup> H. C. van de Hulst, *Light Scattering by Small Particles* (New York, Wiley, 1957), Chapter 10.

<sup>8</sup> J. A. Stratton, *Electromagnetic Theory* (New York, McGraw-Hill, 1940), p. 570ff.

been made possible by the great increases in the speed and availability of digital computers over the past three decades. The purpose of this report is to review the articles scattered through the literature since 1957 on the numerical evaluation of the roots of (10–11) and of  $j_l$  and  $h_l^{(1)}$  for complex arguments, and to provide a set of tested algorithms that permit rapid, accurate numerical computations.

## 2 The Calculation of Spherical Bessel Functions and Their Logarithmic Derivatives

The precision required in the calculation of the spherical Bessel functions depends on the application one has in mind. A relatively low precision of several significant decimal digits is required in calculations of the scattering cross section, while much higher precision is required for the accurate location of narrow resonances. Infinite series and the explicit formulae for the spherical Bessel functions in terms of trigonometric functions both fail for large arguments because of catastrophic cancellation when arithmetic can be performed only with a finite, fixed number of significant digits, as is the case for hand computation or any of the usual floating-point representations in use on digital computers. Series methods work for general arguments only when an arbitrary number of significant digits can be employed. Symbolic-manipulation programs such as MACSYMA offer arbitrary-precision arithmetic, but at a prohibitive penalty in speed.

### 2.1 Recurrence Relations for the Bessel and Spherical Bessel Functions

Modern methods for the evaluation of a solution of Bessel's equation  $Z_\nu(x)$  are based upon the recurrence relation

$$Z_{\nu-1}(x) + Z_{\nu+1}(x) = \frac{2\nu}{x} Z_\nu(x) . \quad (15)$$

An obvious application of Eq. (15) is to calculate a sequence of values proportional to  $J_\nu(x)$  for real order  $\nu$  by downward recurrence starting from an approximate value (even zero) for an order considerably higher than the order desired. Any of several normalization techniques then yields  $J_\nu(x)$  to an accuracy that can be pre-selected.<sup>9</sup>

It is essential to know the numerical stability of a recurrence relation before applying it to the calculation of a set of functions. If both the computed solution  $\bar{Z}_\nu(x)$  and the exact solution  $Z_\nu(x)$  satisfy Eq. (15) within roundoff error, then the error  $\epsilon_\nu(x) := \bar{Z}_\nu(x) - Z_\nu(x)$  satisfies the same recurrence relation and is

---

<sup>9</sup> J. C. P. Miller, *British Association for the Advancement of Science, Mathematical Tables*, Vol. X, *Bessel Functions*, Part II (Cambridge University Press, 1952), p. xvi.

therefore equal to a linear combination of two linearly independent solutions of Bessel's equation, say

$$\epsilon_\nu(x) = \alpha J_\nu(x) + \beta Y_\nu(x) . \quad (16)$$

From the asymptotic relations<sup>10</sup>

$$J_\nu(x) \sim \frac{1}{\sqrt{2\pi\nu}} \left(\frac{ex}{2\nu}\right)^\nu \quad (17)$$

$$Y_\nu(x) \sim -\sqrt{\frac{2}{\pi\nu}} \left(\frac{ex}{2\nu}\right)^{-\nu} \quad (18)$$

it follows that for large  $\nu$ ,

$$\frac{J_{\nu-1}(x)}{J_\nu(x)} \sim \frac{2\nu}{x} \quad (19)$$

while

$$\frac{Y_{\nu-1}(x)}{Y_\nu(x)} \sim \frac{x}{2\nu} . \quad (20)$$

Thus  $|Y_{\nu-1}(x)/Y_\nu(x)|$  becomes small without limit as  $\nu$  becomes much larger than  $|x|$  while  $|J_{\nu-1}(x)/J_\nu(x)|$  becomes large in the same limit. It follows that an attempt to calculate the Bessel function  $J_\nu(x)$  by upward recurrence (increasing  $\nu$ ) would be doomed to failure, since the unavoidable Neumann component  $Y_\nu(x)$  of the error  $\epsilon_\nu(x)$  would eventually become arbitrarily large. Upward recurrence is stable for the Neumann function, while downward recurrence is stable for the Bessel function.

Similar conclusions apply to the spherical Bessel functions

$$z_n(x) = \sqrt{\frac{\pi}{2x}} Z_{n+\frac{1}{2}}(x) , \quad (21)$$

which obey the recurrence relations

$$z_{n-1}(x) + z_{n+1}(x) = \frac{2n+1}{x} z_n(x) \quad (22)$$

$$\frac{n+1}{x} z_n(x) + \frac{d}{dx} z_n(x) = z_{n-1}(x) . \quad (23)$$

Corbató and Uretsky have given a careful discussion of algorithms for the calculation of  $j_n(x)$  for real  $x$  based on downward recurrence.<sup>11</sup> In view of the above

<sup>10</sup> M. Abramowitz and I. A. Stegun, "Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables," National Bureau of Standards Applied Mathematics Series No. 55 (U. S. Government Printing Office, 1965), p. 365.

<sup>11</sup> F. J. Corbató and J. L. Uretsky, "Generation of Spherical Bessel Functions in Digital Computers", J. Assoc. Comp. Mach. **6**, 366–375 (1959).

comments, backward-recursive algorithms are not appropriate for the spherical Neumann or Hankel functions.

A computer program for the generation of the spherical Bessel functions was created by Dusel et al.,<sup>12</sup> who used (22) in the upward direction for the calculation of the Hankel functions and in the downward direction for the calculation of the Bessel functions. Since this program uses function values rather than ratios or logarithmic derivatives, it suffers from loss of significant digits for large size parameters  $x$ .

Gillman and Fiebig have recently given forward recursive algorithms for the spherical Neumann functions and carefully designed backward recursive relations for the spherical Bessel functions.<sup>13</sup> Since the Gillman-Fiebig algorithms generate the actual values of the functions (with the leading term in the series for small  $x$  divided out) they may produce quantities so large as to produce loss of significant digits for large complex arguments, for which the magnitudes of the Bessel and Neumann functions grow exponentially. The computer programs presented by Gillman and Fiebig are written for a purely real argument  $x$ . Large complex arguments are, of course, important in the theory of the scattering of light by conducting or strongly absorbing spheres. Complex arguments are important even in the case of a purely real refractive index, because numerical methods for finding the zeroes of (10–11) require values of the spherical Bessel and Hankel functions in the complex plane.

Some widely available programs for Bessel and Neumann functions are written for integer order and cannot be applied to the calculation of the spherical functions without significant modifications.<sup>14</sup> Although the IMSL Bessel-function programs allow arbitrary real orders, they are restricted to real arguments.

## 2.2 Recurrence Relations for the Ratios and Logarithmic Derivatives of the Riccati-Bessel Functions

At this point it is convenient to introduce the Riccati-Bessel functions

$$\psi_n(x) = x j_n(x) \tag{24}$$

and

$$\zeta_n(x) = x h_n^{(1)}(x) \tag{25}$$

---

<sup>12</sup> P. W. Dusel, “A Numerical Study of Energy Absorption in Small Spheres”, M.Eng. Thesis, Clarkson College of Technology (1976); P. W. Dusel, M. Kerker and D. D. Cooke, “Distribution of Absorption Centers Within Irradiated Spheres”, *J. Opt. Soc. Am.* **69**, 55–59 (1979).

<sup>13</sup> E. Gillman and H. R. Fiebig, “Accurate recursive generation of spherical Bessel and Neumann functions for a large range of indices”, *Computers in Physics* **1**, 62–72 (1988).

<sup>14</sup> W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, *Numerical Recipes* (Cambridge University Press, 1986), p. 170ff.

since the Mie formulae (6–11) can be recast entirely in terms of  $\psi_n$ ,  $\zeta_n$  and their logarithmic derivatives, as follows:

$$a_n = \frac{\psi_n(ka)}{\zeta_n(ka)} \frac{A_n(k'a) - \sqrt{\epsilon} A_n(ka)}{A_n(k'a) - \sqrt{\epsilon} B_n(ka)} \quad (26)$$

$$b_n = \frac{\psi_n(ka)}{\zeta_n(ka)} \frac{\sqrt{\epsilon} A_n(k'a) - A_n(ka)}{\sqrt{\epsilon} A_n(k'a) - B_n(ka)} \quad (27)$$

$$c_n = \frac{i}{\zeta_n(ka) \psi_n(k'a) [A_n(k'a) - \sqrt{\epsilon} B_n(ka)]} \quad (28)$$

$$d_n = \frac{i\sqrt{\epsilon}}{\zeta_n(ka) \psi_n(k'a) [\sqrt{\epsilon} A_n(k'a) - B_n(ka)]} . \quad (29)$$

In (26–29),  $A_n$  and  $B_n$  are the logarithmic derivatives of  $\psi_n$  and  $\zeta_n$ , respectively:

$$A_n(x) := \frac{1}{x j_n(x)} \frac{d(x j_n(x))}{dx} \quad (30)$$

$$B_n(x) := \frac{1}{x h_n^{(1)}(x)} \frac{d(x h_n^{(1)}(x))}{dx} . \quad (31)$$

The ratios and logarithmic derivatives of the Riccati-Bessel and Riccati-Hankel functions obey useful recurrence relations. If we let  $\chi_n$  denote either the Riccati-Bessel or the Riccati-Hankel function, it follows from (23) that

$$\frac{d\chi_n}{dx} = \chi_{n-1} - \frac{n}{x} \chi_n , \quad (32)$$

so that the logarithmic derivative  $C_n$  is related to the ratio of functions of successive orders  $\rho_n$  as follows:

$$C_n := \frac{1}{\chi_n} \frac{d\chi_n}{dx} = \frac{\chi_{n-1}}{\chi_n} - \frac{n}{x} = \rho_{n-1} - \frac{n}{x} . \quad (33)$$

Then (22) implies that the ratio

$$\rho_n := \frac{\chi_n}{\chi_{n+1}} = \frac{z_n}{z_{n+1}} = \frac{Z_{n+\frac{1}{2}}}{Z_{n+\frac{3}{2}}} \quad (34)$$

obeys the two-term recurrence relation

$$\frac{1}{\rho_n} + \rho_{n-1} = \frac{2n+1}{x} . \quad (35)$$

From (33–35) one sees that the logarithmic derivative obeys the upward recurrence relation

$$C_n = -\frac{n}{x} + \frac{1}{\frac{n}{x} - C_{n-1}} . \quad (36)$$

and the mathematically but not numerically equivalent downward recurrence relation

$$C_n = \frac{n+1}{x} - \frac{1}{\frac{n+1}{x} + C_{n+1}} . \quad (37)$$

The stability of the recurrence relations (35–37) depends on the function whose ratio is to be calculated and on the direction in which the recurrence is applied, as has already been noted in conjunction with the equivalent relation (15) for the normal cylinder functions.

The computational advantage of the logarithmic derivatives and the ratios is that these quantities are all of comparable orders of magnitude and in fact are generally within a very few orders of magnitude of unity even for complex arguments, thus minimizing problems of overflow and arithmetic with quantities of very different orders of magnitude. Infeld<sup>15</sup> appears to have been the first to recognize the usefulness of the logarithmic derivatives of solutions of the spherical Bessel equation for numerical computation. Infeld’s calculation of the logarithmic derivative  $B_n$  of the spherical Riccati-Hankel function  $\zeta_n$  was based on a complicated recurrence relation that reduces to (36) after some algebra and that appears to offer no numerical advantage with respect to (36).

Deirmendjian *et al.*<sup>16</sup> appear to have been the first to calculate the logarithmic derivative  $A_n$  of the Riccati-Bessel function  $\psi_n$  to evaluate the Mie coefficients (6–11). Their calculations extended up to size parameter  $x = 70$  at intervals of 0.5. In retrospect their work suffered from several problems: They used the upward recurrence relation (36) to calculate  $A_n$ ; when they recast (6–11) in terms of logarithmic derivatives they introduced only  $A_n$  and not  $B_n$ , choosing instead to express the derivative of  $\zeta_n$  via the recurrence relation (32); and the interval chosen in  $x$  was too great to reveal narrow resonances.

In 1968 Dave<sup>17</sup> improved upon the approach of Deirmendjian *et al.* by correctly applying downward recursion to the calculation of  $A_n$ . Like Deirmendjian, he expressed the derivative of  $\zeta_n$  in terms of  $\zeta_{n-1}$  and  $\zeta_n$  rather than in terms of  $B_n$ . Dave’s Mie subroutines DBMIE (which uses downward recursion for  $A_n$ ) and DAMIE (which uses upward recursion) have been the basis of much subsequent work, such as that of Bennett and Rosasco.<sup>18</sup> The performance of

<sup>15</sup> L. Infeld, “The Influence of the Width of the Gap Upon the Theory of Antennas”, *Quart. Appl. Math.* **5**, 113–132 (1947).

<sup>16</sup> D. Deirmendjian, R. Clasen and W. Viezee, *J. Opt. Soc. Am.* **51**, 620–633 (1961).

<sup>17</sup> J. V. Dave, “Subroutines for Computing the Parameters of the Electromagnetic Radiation Scattered by a Sphere”, IBM Palo Alto Scientific Center Report No. 320.3237 (1968).

<sup>18</sup> H. S. Bennett and G. J. Rosasco, “Resonances in the Efficiency Factors for Absorption: Mie Scattering Theory”, *Appl. Opt.* **17**, 491–493 (1978).

DBMIE was impressive by comparison with all previous Mie-scattering calculations. Dave’s published test case used a size parameter  $x = 1570.796$ , far greater than any for which convergence had previously been obtained.

### 2.3 Continued-Fraction Methods

Lentz<sup>19</sup> pioneered a different direction in the calculation of the spherical Bessel functions by introducing a continued-fraction method that permits the convenient evaluation of a function to any desired precision. Lentz’s method is the basis for the Bessel-function subroutine CONFRA in the efficient Mie-scattering computer program MIEV0 written by Wiscombe<sup>20</sup> to take advantage of the vector computation capabilities of modern supercomputers. Since Lentz’s approach was considerably different than in other work, it deserves a detailed discussion.

Any linear three-term recurrence relation such as (15) or (22) can be converted to a continued fraction for the ratio of two successive terms. Thus, from (35), one has

$$\rho_n = \frac{2n+3}{x} - \frac{1}{\frac{2n+5}{x} - \frac{1}{\frac{2n+7}{x} - \dots}} . \quad (38)$$

Of course, (38) will be numerically stable only when (35) is stable in the downward direction, *i.e.* for the calculation of the ratio of Bessel rather than Neumann or Hankel functions. Since (38) is equivalent to the normal downward recurrence relation, it is useful only when it is more convenient or can be made more accurate than downward recurrence. Standard methods<sup>21</sup> for the evaluation of continued fractions are equivalent to the application of recurrence in the upward direction, and are therefore useless in the present context.

One of the main disadvantages of downward recurrence is the necessity to choose as good a starting value of the function, ratio or logarithmic derivative as possible in order to avoid unnecessary iterations of the recurrence. The main disadvantage of a naive approach to evaluating a continued fraction is that one must choose a point at which to terminate the fraction and then calculate downward, effectively going through the same steps as in downward recursion. Lentz discovered a way to avoid having to choose a starting point by turning the continued fraction “inside out”. Following the standard practice of denoting

<sup>19</sup> W. J. Lentz, “Generating Bessel functions in Mie scattering calculations using continued fractions”, *Appl. Opt.* **15**, 668–671 (1976).

<sup>20</sup> W. J. Wiscombe, “Improved Mie scattering algorithms”, *Appl. Opt.* **19**, 1505–1509 (1980); W. J. Wiscombe, “Mie Scattering Calculations: Advances in Technique and Fast, Vector-Speed Computer Codes”, National Center for Atmospheric Research Technical Note NCAR/TN-140+STR (1979).

<sup>21</sup> W. J. Press *et al.*, *op. cit. supra.*

the continued fraction

$$f = a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \cdots}} \quad (39)$$

as

$$f = a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \cdots}} \quad (40)$$

one defines the  $n^{\text{th}}$  convergent as the finite continued fraction

$$f_n := [a_1, \dots, a_n] \quad (41)$$

$$= a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \cdots + \frac{1}{a_n}}} . \quad (42)$$

Evidently

$$\lim_{n \rightarrow \infty} f_n = f. \quad (43)$$

The development of Chrystal<sup>22</sup> provides a convenient framework for the next step. If we express the convergent  $f_n$  as the ratio

$$f_n = \frac{p_n}{q_n}, \quad (44)$$

then the numerators and denominators obey the recurrence relations

$$p_n = a_n p_{n-1} + p_{n-2} \quad (45)$$

$$q_n = a_n q_{n-1} + q_{n-2} \quad (46)$$

provided that one adopts the starting values

$$p_{-1} = 0 ; \quad p_0 = a_1 ; \quad p_1 = a_1 \quad (47)$$

$$q_{-1} = 1 ; \quad q_0 = 0 ; \quad q_1 = 1 . \quad (48)$$

The proof is inductive: Obviously (45–46) are true for  $n = 1$ . Assuming that they hold for  $n$ , we note that one obtains  $f_{n+1}$  from  $f_n$  by replacing  $a_n$  by  $a_n + 1/a_{n+1}$ . If we do so in the expression

$$f_n = \frac{a_n p_{n-1} + p_{n-2}}{a_n q_{n-1} + q_{n-2}} \quad (49)$$

---

<sup>22</sup> G. Chrystal, *Algebra: An Elementary Text-Book*, Part Two (London, Adam and Charles Black, 1900; reprint: New York, Dover, 1961), pp. 431–433.

we obtain

$$f_{n+1} = \frac{\left(a_n + \frac{1}{a_{n+1}}\right) p_{n-1} + p_{n-2}}{\left(a_n + \frac{1}{a_{n+1}}\right) q_{n-1} + q_{n-2}} = \frac{a_{n+1} p_n + p_{n-1}}{a_{n+1} q_n + q_{n-1}} = \frac{p_{n+1}}{q_{n+1}} . \quad (50)$$

Now the recurrences (45–46) imply that

$$\frac{p_n}{p_{n-1}} = a_n + \frac{1}{\frac{p_{n-1}}{p_{n-2}}} \quad (51)$$

$$\frac{q_n}{q_{n-1}} = a_n + \frac{1}{\frac{q_{n-1}}{q_{n-2}}} . \quad (52)$$

Eqs. (51–52) lead to the *terminating* continued fractions

$$\frac{p_n}{p_{n-1}} = a_n + \frac{1}{a_{n-1} + \cdots + \frac{1}{a_1}} = [a_n, \dots, a_1] \quad (53)$$

$$\frac{q_n}{q_{n-1}} = a_n + \frac{1}{a_{n-1} + \cdots + \frac{1}{a_2}} = [a_n, \dots, a_2] . \quad (54)$$

Then we can express the convergent  $f_n$  as

$$\begin{aligned} f_n &= \frac{p_n}{p_{n-1}} \frac{p_{n-1}}{p_{n-2}} \cdots p_1 \left( \frac{q_n}{q_{n-1}} \frac{q_{n-1}}{q_{n-2}} \cdots q_2 \right)^{-1} \\ &= \frac{[a_n, \dots, a_1] [a_{n-1}, \dots, a_1] \cdots a_1}{[a_n, \dots, a_2] [a_{n-1}, \dots, a_2] \cdots a_2} . \end{aligned} \quad (55)$$

Eq. (55), which is the basis for Lentz’s method and Wiscombe’s subroutine CONFRA, has the advantage over Eq. (42) that it is trivial to calculate  $[a_n, \dots, a_1]$  from  $[a_{n-1}, \dots, a_1]$  for use in (55), while the passage from  $n$  to  $n+1$  in (42) can be accomplished only by recalculating the entire continued fraction. Thus (55) permits one to increase  $n$  at the expense of two additions and three divisions per step until the ratio  $[a_n, \dots, a_1]/[a_n, \dots, a_2]$  cannot be distinguished from unity. This provides the maximum precision available in a particular floating-point representation as simply as possible, and probably with a minimum of computational effort.

### 3 Location of Mie Resonances

While resonances in the Mie scattering amplitudes  $a_n$  and  $b_n$  for size parameters  $x$  of the order of 5 to 10 were known for many years, it is only recently that

Bennett and Rosasco<sup>23</sup> and Chylek et al.<sup>24</sup> brought the attention of workers in the field to the existence of very narrow resonances for size parameters of 20 to 30 or more. Partial-wave resonances have significant effects on the optical levitation of spherical particles<sup>25</sup> and on the spectrum of stimulated Raman light generated within a spherical aerosol.<sup>26</sup> Cantrell<sup>27</sup> and Chitanvis and Cantrell<sup>28</sup> have recently suggested that simultaneous resonance of the incident laser light and light produced by stimulated Brillouin scattering may generate sufficiently high acoustic pressures to shatter an aerosol. Order-of-magnitude estimates show that at least one wave must be resonant for an aerosol to display interesting nonlinear-optical effects due to internal fields. Accurate location of the partial-wave resonances is therefore a prerequisite to a quantitative study of nonlinear-optical effects within spherical beads or droplets.

### 3.1 Previous Work

The physical interpretation of the partial-wave resonances is that they correspond to standing waves trapped just inside the surface of the sphere, as conjectured by van de Hulst and demonstrated numerically by Conwell *et al.*<sup>29</sup> and Chylek et al.<sup>30</sup> The latter authors observed enhancements of up to  $10^6$  in the ratio of internal to incident electric-field intensity under resonant conditions.

The brute-force way to locate resonances is to calculate the Mie scattering efficiency or the internal electric-field intensity as a function of real size parameter  $x$  with a sufficiently fine step size to ensure that all resonances will be detected. This is the approach of Bennett and Rosasco and Chylek et al. However, this method requires unreasonable amounts of computer time for size parameters greater than roughly 40.

The most thorough published numerical study of the partial-wave resonances is that of Conwell *et al.*, who systematically located resonances as functions of the partial-wave order  $n$  by solving for the zeroes of the denominators of (10) and (11) in the complex  $x$ -plane. For this purpose they used the IMSL subroutine ZANLYT, which solves for the zeroes of an analytic function using Muller's method (which essentially consists of approximating the first three terms of a Taylor series for the function).<sup>31</sup> In some respects their numerical methods were

---

<sup>23</sup> H. S. Bennett and G. J. Rosasco, *op. cit. supra* (1978).

<sup>24</sup> P. Chylek, J. T. Kiehl and M. K. W. Ko, "Narrow resonance structure in the Mie scattering characteristics", *Appl. Opt.* **17**, 3019–3021 (1978).

<sup>25</sup> P. Chylek, J. T. Kiehl and M. K. W. Ko, "Optical levitation and partial-wave resonances", *Phys. Rev. A* **18**, 2229–2233 (1978).

<sup>26</sup> J. B. Snow, S.-X. Qian and R. K. Chang, "Stimulated Raman scattering from individual water and ethanol droplets at morphology-dependent resonances", *Opt. Lett.* **8**, 37–39 (1985).

<sup>27</sup> C. D. Cantrell, *op. cit. supra*.

<sup>28</sup> S. M. Chitanvis and C. D. Cantrell, "Simple approach to stimulated Brillouin scattering in glass aerosols", *Journal of the Optical Society of America B* **6**, 1326–1331 (1989).

<sup>29</sup> P. R. Conwell, P. W. Barber and C. K. Rushforth, "Resonant spectra of dielectric spheres", *J. Opt. Soc. Am. A* **1**, 62–67 (1984).

<sup>30</sup> P. Chylek, J. D. Pendleton and R. G. Pinnick, "Internal and near-surface scattered field of a spherical particle at resonant conditions", *Appl. Opt.* **24**, 3940–3942 (1985).

<sup>31</sup> D. E. Muller, "A Method for Solving Algebraic Equations using an Automatic Computer",

relatively unsophisticated. For example, they used (22) for downward recurrence of the spherical Bessel functions, and did not take advantage of the simplified forms (26–29) of the partial-wave amplitudes. Nevertheless their article was invaluable for the present investigation.

## 3.2 Methods and Results of This Work

### 3.2.1 Brute-Force Approach

The initial attempts at locating partial-wave resonances in this work relied on the brute-force approach.

The angle-averaged electromagnetic energy density in a dielectric is  $\langle U^{(w)} \rangle = \int U^{(w)} d\Omega / 4\pi$ , where  $U^{(w)} = (\mathbf{D} \cdot \mathbf{E} + \mathbf{B} \cdot \mathbf{H}) / 8\pi$  in terms of real fields and where angle brackets denote an average over  $\theta$  and  $\phi$ . The incident energy density is  $U^{(i)} = E_0^2 / 8\pi$ . From (4–5) and the orthogonality properties of the vector spherical harmonics it follows that

$$\begin{aligned} \langle U^{(w)} \rangle &= \frac{1}{32\pi} \int \left[ \mathbf{E}^{(w)*} \cdot \mathbf{D}^{(w)} + \mathbf{D}^{(w)*} \cdot \mathbf{E}^{(w)} + \mathbf{H}^{(w)*} \cdot \mathbf{B}^{(w)} + \mathbf{B}^{(w)*} \cdot \mathbf{H}^{(w)} \right] \frac{d\Omega}{4\pi} \\ &= \frac{1}{16\pi} \int \left[ \text{Re}(\epsilon) \mathbf{E}^{(w)*} \cdot \mathbf{E}^{(w)} + \mathbf{H}^{(w)*} \cdot \mathbf{H}^{(w)} \right] \frac{d\Omega}{4\pi} \\ &= \frac{E_0^2}{32\pi} \sum_l \left\{ |\epsilon| \left[ l |j_{l+1}(k'r)|^2 + (l+1) |j_{l-1}(k'r)|^2 \right] \left[ \text{Re}(\epsilon) |c_l|^2 + |d_l|^2 \right] \right. \\ &\quad \left. + (2l+1) |j_l(k'r)|^2 \left[ \text{Re}(\epsilon) |d_l|^2 + |\epsilon|^2 |c_l|^2 \right] \right\}. \end{aligned} \tag{56}$$

It is puzzling that both Conwell *et al.* and Chylek *et al.* defined intensities proportional to  $\mathbf{E} \cdot \mathbf{E}$ , without considering the role of either  $\mathbf{D}$  or  $\mathbf{H}$  in the total energy density or the fact that the intensity is properly defined as the amplitude of the Poynting vector  $\mathbf{S} = (c/4\pi) \mathbf{E} \times \mathbf{H}$ .

For stimulated Brillouin scattering in a spherical particle, however, the source term is proportional to  $\mathbf{E} \cdot \mathbf{E}$ .<sup>4,28</sup> From (4) one finds that the angle-averaged square of the real electric field is

$$\begin{aligned} \langle \mathbf{E}^2 \rangle &= \frac{E_0^2}{2} \sum_l \left\{ |\epsilon| \left[ l |j_{l+1}(k'r)|^2 + (l+1) |j_{l-1}(k'r)|^2 \right] |c_l|^2 \right. \\ &\quad \left. + (2l+1) |j_l(k'r)|^2 |d_l|^2 \right\}. \end{aligned} \tag{57}$$

Outside the sphere, the square of the real electric field, averaged over a few optical cycles, is  $E_0^2/2$ .

The program RES in Appendix A calculates either (56) (divided by  $E_0^2/8\pi$ ) or (57) (divided by  $E_0^2/2$ ) according to the user's choice.

---

Math. Tables & Other Aids to Comp. **10**, 208–215 (1956).

The Bessel-function subroutine in RES is called SPHBESS. In SPHBESS, the logarithmic derivative of the Riccati-Bessel function is calculated by downward recursion using Dave's method. Riccati-Bessel functions (or their logarithmic derivatives) are required both inside the sphere, where the refractive index may be complex, and outside, where I have assumed that the refractive index is 1. A real array is used for the logarithmic derivative  $A_n(ka)$  outside (ACAPR) while a complex array must be used for  $A_n(k'a)$  inside (ACAP). The spherical Bessel functions are then calculated by successive upward multiplication

$$j_n(x) = \frac{j_{n-1}(x)}{\rho_{n-1}^{(j)}(x)} \quad (58)$$

using the ratios  $\rho_n^{(j)}$ , which are equal to

$$\rho_n^{(j)} = A_{n+1} + \frac{n+1}{x} \quad (59)$$

according to (33). The starting value is  $j_0(x) = \sin x/x$ . This procedure produces values of the spherical Bessel functions that agree with tabulated values to the full precision of the tables for real arguments.<sup>32</sup>

The spherical Neumann function, which is required only outside the sphere (hence only for real  $x$ ), is calculated by upward recurrence since this is always stable for the Neumann function of real argument. The values so obtained agree with the values tabulated by Abramowitz and Stegun. The spherical Hankel function is then calculated from (13). This approach to calculating the spherical Hankel function was adopted after it became apparent that straightforward upward recurrence of the Hankel functions using (22) for real  $x$  produces a real part ( $j_n$ ) that agrees with tabulated values to only five or six significant figures even when the calculation is performed in double (64-bit) precision.

The exterior partial-wave amplitudes  $a_n$  and  $b_n$  are calculated in subroutine PWAVE using the formulae

$$a_n = \frac{j_n(ka) [A_n(k'a)/\sqrt{\epsilon} - A_n(ka)]}{[A_n(k'a)/\sqrt{\epsilon} + n/(ka)] h_n^{(1)}(ka) - h_{n-1}^{(1)}(ka)} \quad (60)$$

$$b_n = \frac{j_n(ka) [\sqrt{\epsilon}A_n(k'a) - A_n(ka)]}{[\sqrt{\epsilon}A_n(k'a) + n/(ka)] h_n^{(1)}(ka) - h_{n-1}^{(1)}(ka)}. \quad (61)$$

Note that  $k' = \sqrt{\epsilon}k$ . These formulae differ from those used by Dave in that downward recurrence is used here to calculate the Bessel functions outside the sphere, while Dave used upward recurrence. The numerical values calculated with (59-60) agree with those calculated by DBMIE.

The values of the interior partial-wave amplitudes  $c_n$  and  $d_n$  are needed only for the calculation of the energy densities at  $r = a$  using (56-57), and are

<sup>32</sup> M. Abramowitz and I. A. Stegun, *op. cit. supra*, pp. 465-466.

therefore calculated in the same subroutine, DENS. The formulae used are

$$c_n = \frac{i}{\epsilon(ka)^2 j_n(k'a) \left\{ [A_n(k'a)/\sqrt{\epsilon} + n/(ka)] h_n^{(1)}(ka) - h_{n-1}^{(1)}(ka) \right\}} \quad (62)$$

$$d_n = \frac{i}{(ka)^2 j_n(k'a) \left\{ [\sqrt{\epsilon}A_n(k'a) + n/(ka)] h_n^{(1)}(ka) - h_{n-1}^{(1)}(ka) \right\}} . \quad (63)$$

Program RES requests the following inputs from the user: The real and imaginary parts of the refractive index  $\sqrt{\epsilon}$ , the upper and lower limits of the range of real size parameters  $x$  for which the ratio of internal to external energy density is to be found, the step size in  $x$ , and a variable that selects either (56) or (57). The output, to FORTRAN device number 9, consists of a single line for each value of  $x$  giving  $x$  and the energy-density ratio selected by the user.

The numerical computations were performed on a Convex C-1 minisuper-computer. Although the Convex compiler automatically vectorizes program units up to subprogram boundaries, much of the calculation was inherently recursive and could not be vectorized. Double precision (REAL\*8 and COMPLEX\*16) was used throughout RES in order to forestall problems from loss of significance. While RES requires relatively little time for values of  $x$  less than 30 (0.095s per value of  $x$  for  $x \leq 135$ ), it requires roughly one CPU-day to calculate the interval  $376.9 \leq x \leq 377.1$ . The dramatic increase of computation time at high values of  $x$  is the direct result of the very small step sizes necessitated by the great narrowing of the primary resonances as  $x$  increases. In fact,  $\Delta x = 10^{-7}$  at  $x = 377$  was insufficient to resolve one resonance with a full-width at half maximum of  $1.4 \times 10^{-13}$ .

When RES is run with very small step sizes, the size of the output file may exceed the available disk storage. In order to minimize this problem, the output section of RES has been coded so that results are actually written to disk only when one of three conditions is met: (a) The separation between the present value of  $x$  and the value for which results were most recently written exceeds a certain value; (b) The value of the energy density most recently written to disk is below the present one by a certain tolerance; (c) The value of the energy density most recently written to disk is above the present one by a certain tolerance. The relevant parameters are hardwired in the present version of RES.

A slightly modified version of RES was prepared in order to permit the identification of the resonating partial wave when a resonant value of  $x$  is known.

The numerical results obtained with RES, which are presented in Figs. 1–7 and Tables 1 and 2, are discussed in greater detail below.

### 3.2.2 Root-Finding Approach

Conwell *et al.* recognized that finding the roots of the denominators of the partial-wave amplitudes (6–9), (26–29) or (60–63) permits a more systematic

and somewhat less expensive study of partial-wave resonances than the brute-force approach.

As has already been described, Conwell *et al.* used the IMSL subroutine ZANLYT to locate the roots of (6–9). In order to test ZANLYT I attempted to solve for the roots of the function

$$f(z) = \tan z + Ki , \quad (64)$$

since applying the asymptotic forms of the Bessel and Hankel functions to (6–9) leads to the condition

$$i\sqrt{\epsilon} + \tan \left( \sqrt{\epsilon}x - \frac{n+1}{2}\pi \right) = 0 \quad (65)$$

for the vanishing of the electric (TM) partial-wave denominators and the condition

$$1 - i\sqrt{\epsilon} \tan \left( \sqrt{\epsilon}x - \frac{n+1}{2}\pi \right) = 0 \quad (66)$$

for the vanishing of the magnetic (TE) partial-wave denominators.<sup>33</sup>

Unfortunately ZANLYT performed very poorly on  $f(z)$ , skipping many roots and finding spurious roots even when it was given input values very close to the analytically known roots of  $f(z)$ ,

$$z_l = l\pi + \frac{i}{2} \ln \left( \frac{K-1}{K+1} \right) . \quad (67)$$

The causes of the failure of ZANLYT in the case of (64) are unknown. However, since ZANLYT fails for a function  $f(z)$  that presents few of the numerical complexities of the functions whose roots are desired, it was necessary to look elsewhere for a root-finding routine.

If a complex function  $f(z)$  vanishes at  $z = z_0$ , its modulus  $|f(z)|$  and modulus squared  $|f(z)|^2$  achieve a minimum value of zero at  $z_0$ . From the point of view of locating roots accurately,  $|f(z)|$  is preferable since it has a finite gradient in the vicinity of a simple zero, while the gradient of  $|f(z)|^2$  approaches zero even if the zero is simple.

Surface plots of the moduli of (64) and the functions

$$f_{TM}^{(n)}(x) = A_n(\sqrt{\epsilon}x) - \sqrt{\epsilon} B_n(x) \quad (68)$$

$$f_{TE}^{(n)}(x) = \sqrt{\epsilon} A_n(\sqrt{\epsilon}x) - B_n(x) , \quad (69)$$

which vanish at the resonant values of  $x$  according to (26–29), are shown in Figs. 8–11. In (68–69),  $TE$  refers to the magnetic partial wave and  $TM$  refers to the electric partial wave.

---

<sup>33</sup> J. A. Stratton, *op. cit.*, p. 570.

The program FINDRES3, which is listed in Appendix B, is the third in a series of attempts to use a root-finding approach to the location of resonances. The first attempt, based on (60–63), made use of the same Bessel-function subroutines as in RES, as well as the IMSL root-finder ZANLYT. The second attempt employed improved routines based on (26–29) that require only the values of the logarithmic derivatives and not of the Riccati-Bessel functions themselves for the calculation of the partial-wave denominators, and ZANLYT. In the third and current attempt, ZANLYT has been replaced by the IMSL subroutine ZXMIN, which locates local minima by a variant of Newton’s method. ZXMIN performs well on the test problem (64). ZXMIN was applied to the moduli of (68,69), which are calculated in subroutine DENOM. FINDRES3 produces a reasonably satisfactory set of computed resonances the positions and widths of which are in excellent agreement with those found by the brute-force approach. Some of the numerical results obtained with FINDRES3 are presented in Appendix C, where one sees that FINDRES3 sometimes failed to find the narrowest (primary) resonance and sometimes converged to the same root twice. The use of a more sophisticated search procedure to determine better starting values for ZXMIN would probably alleviate these problems.

Wiscombe’s subroutine CONFRA is used in FINDRES3 to calculate the logarithmic derivative  $A_n$  of the Riccati-Bessel function. Since CONFRA was originally written for a Cray, in which normal precision is 64 bits, it had to be modified to specify REAL\*8 and COMPLEX\*16 variables for use on a Convex. Also, since the values of  $A_n$  should have the full accuracy available, the variable EPS2, which is used to set the criterion for terminating the calculation of the partial-fraction convergents, was changed from 1.D–8 to 4.44D–16, which is twice machine epsilon on the Convex and on all machines that use the IEEE standard for floating-point representations.

An in-line substitute for the FORTRAN library routine CABS has been inserted in the function subprogram FUNCT, which calculates the function to be minimized by ZXMIN, in order to increase the accuracy of calculations of the absolute value of complex arguments in which the real and imaginary parts have different orders of magnitude.

The method used for calculation of the logarithmic derivative  $B_n$  of the Riccati-Hankel functions is upward recurrence of Eq. (35). When this procedure is tested for real argument  $x$  by calculating the values of the spherical Hankel function by upward multiplication using a formula analogous to (58), the values of the real and imaginary part agree with tabulated values of  $j_n(x)$  and  $y_n(x)$  to the precision of the tables. The code used to test the function values has been preserved in the listing in Appendix B.

FINDRES3 is almost entirely scalar code. Vectorization produces no perceptible improvement in performance.

## 4 Discussion

RES was run to determine the ratio of the interior to exterior electric-field density for values of  $x$  ranging from 1 to 40 and from 376.9 to 377.1. The results are shown in Figs. 1–7 and Tables I and II. I know of no other detailed plot or tabulation of Mie resonances over a comparable range of size parameter. Several systematic trends, some of which are not yet fully understood, are evident in Figs. 1–6. The broad resonances that early investigators found at low values of  $x$  quickly become narrower and higher as  $x$  increases, first revealing the existence of two kinds of primary resonances and later revealing secondary, tertiary, etc., resonances. The primary electric resonance ( $TM_{n,1}$  in the notation of Chylek *et al.*<sup>30</sup>) for a given partial wave  $n$  always occurs at a higher value of  $x$  than the primary magnetic resonance  $TE_{n,1}$ , but the amplitude of the magnetic ( $TE$ ) resonance is higher. Other narrow resonances become visible near the primary resonances for  $x \gtrsim 25$ . These are the secondary resonances ( $TE_{n,2}$  or  $TM_{n,2}$ ), several of which are assigned in Figs. 3–6. The enhancement of the exterior electric-field density ranges from a factor of 5 at  $x = 15$  to more than a factor of  $10^{10}$  at  $x = 377.05$ .

FINDRES3 was run for values of  $x$  from 30 to 60 and for  $x = 100, 400$  and 418. The results are given in Appendix C. From the runs at lower values of the size parameter it is clear that the primitive method coded in FINDRES3 of choosing the starting value for the size parameter needs to be improved. In many cases the primary resonances were missed. It should be possible to code the driver to call ZXMIN with a few significant figures with different values of  $x$ , then analyze the results to choose a better starting  $x$  for a calculation with the full number of significant figures requested by the user. The success of such an attempt would depend to some extent on the development of quantitative formulae for the dependence of the locations and widths of the primary resonances on  $n$ .

The performance of Newton's method in finding the roots of (67–68) might be improved if the numerical calculation of the logarithmic derivative of the function whose roots are to be found (which is now performed in ZXMIN) were replaced by the analytical formulae

$$\frac{1}{f_{TM}^{(n)}(x)} \frac{df_{TM}^{(n)}(x)}{dx} = \frac{\sqrt{\epsilon} \left[ \frac{n(n+1)}{x^2} \left( \frac{1-\epsilon}{\epsilon} \right) + [B_n(x)]^2 - [A_n(\sqrt{\epsilon}x)]^2 \right]}{A_n(\sqrt{\epsilon}x) - \sqrt{\epsilon}B_n(x)} \quad (70)$$

$$\frac{1}{f_{TE}^{(n)}(x)} \frac{df_{TE}^{(n)}(x)}{dx} = \frac{1-\epsilon}{\sqrt{\epsilon}A_n(\sqrt{\epsilon}x) - B_n(x)} - [B_n(x) + \sqrt{\epsilon}A_n(\sqrt{\epsilon}x)] \quad (71)$$

However, this would mean abandoning ZXMIN in favor of an appropriately designed Newton's-method routine, which would have to be coded from scratch.

The fundamental difficulty of using a root-finding approach to locate resonances for values of the size parameter, or, equivalently, for partial-wave orders much greater than those for which most of the present calculations were performed, may be appreciated from the following considerations. One sees from

Figs. 1–6 that the peaks of the primary resonances of a particular type (TE or TM) define a nearly straight line on a semilogarithmic plot. Since the square of the field at resonance is proportional to the square of the resonating amplitude, which in turn can be written approximately as

$$\frac{C}{\gamma_n^2 + (\Delta x)^2},$$

it follows that the width of the primary TE resonance obeys the empirical equation

$$\gamma_n = \alpha e^{-\beta n} . \quad (72)$$

From the data available for  $25 \leq x \leq 35$  one concludes that  $\alpha = 43$  and that  $\beta = .33$ . If (72) can be extrapolated, then for the  $TE_{418}$  family of resonances, for example, the width of the primary resonance should be approximately  $5 \times 10^{-59}$ . Since  $\gamma_n$  is the imaginary part of the root of (68) or (69), it follows that to find both the position and width of such a resonance accurately using Newton's method would require at least 60 significant decimal digits. Such high precision is available only in symbolic-manipulation programs, and only at great cost in terms of computer time. FINDRES3 with REAL\*8 and COMPLEX\*16 arithmetic cannot calculate the imaginary parts of the primary resonances for  $n \gtrsim 100$ . Figs. 8–11 suggest that FINDRES3 may be capable of locating the real part of a resonant value of  $x$  correctly, even when the imaginary part is too small to be determined.

To follow the outline of a resonance using RES requires step sizes smaller than the width of the resonance. RES must fail, then, when the width of the resonance is less than the resonant size parameter times machine epsilon, that is, when  $x + \Delta x$  cannot be distinguished numerically from  $x$ . According to (72) this is the case for  $n \gtrsim 100$  in REAL\*8 or COMPLEX\*16 arithmetic. Undoubtedly the results from  $x = 376.9$  to  $x = 377.1$  shown in Fig. 7 omit several very narrow resonances that could not be detected using a step size  $\Delta x = 10^{-7}$ . To obtain Fig. 7, in fact, it was necessary to make supplementary calculations with step sizes much smaller than  $10^{-7}$  over the intervals in which three of the four resonances were located.

Runs of FINDRES3 for  $n = 400$  illustrate the foregoing comments. The imaginary parts, which should all be negative, fluctuate about zero and are actually unnormalized in the case of the run in which 14 significant figures were requested. Runs for  $n = 418$  show roots with increasingly negative imaginary parts as  $x$  increases from 376. Presumably the primary resonances for  $n = 418$  are at lower values of  $x$  and have widths much smaller than  $10^{-12}$ .

## 5 Acknowledgements

I am grateful to Professor Dale M. Byrne for several helpful discussions, and to Kimberly Heroy for assistance with graphics.

$x_{\text{res}}$	assignment	FWHM	$U_E^{(w)}/U_E^{(i)}$
25.0271	TM <sub>31,1</sub>	$2.3 \times 10^{-3}$	$5.4534 \times 10^1$
25.3040	TE <sub>32,1</sub>	$1.1 \times 10^{-3}$	$1.6721 \times 10^2$
25.7525	TM <sub>32,1</sub>	$1.7 \times 10^{-3}$	$7.1738 \times 10^1$
26.0263	TE <sub>33,1</sub>	$8 \times 10^{-4}$	$2.1900 \times 10^2$
26.4767	TM <sub>33,1</sub>	$1.2 \times 10^{-3}$	$9.4590 \times 10^1$
26.7477	TE <sub>34,1</sub>	$6 \times 10^{-4}$	$2.9140 \times 10^2$
27.1999	TM <sub>34,1</sub>	$9 \times 10^{-4}$	$1.2489 \times 10^2$
27.4682	TE <sub>35,1</sub>	$5 \times 10^{-4}$	$3.8150 \times 10^2$
27.9220	TM <sub>35,1</sub>	$7 \times 10^{-4}$	$1.6578 \times 10^2$
28.18780	TE <sub>36,1</sub>	$3 \times 10^{-4}$	$5.0882 \times 10^2$
28.64310	TM <sub>36,1</sub>	$5 \times 10^{-4}$	$2.2482 \times 10^2$
28.90665	TE <sub>37,1</sub>	$3 \times 10^{-4}$	$6.7618 \times 10^2$
29.36335	TM <sub>37,1</sub>	$4 \times 10^{-4}$	$3.0172 \times 10^2$
29.62475	TE <sub>38,1</sub>	$2 \times 10^{-4}$	$9.0218 \times 10^2$
30.08276	TM <sub>38,1</sub>	$2.4 \times 10^{-4}$	$4.0310 \times 10^2$
30.34214	TE <sub>39,1</sub>	$1.2 \times 10^{-4}$	$1.2174 \times 10^3$
30.80136	TM <sub>39,1</sub>	$1.8 \times 10^{-4}$	$5.4238 \times 10^2$
31.05886	TE <sub>40,1</sub>	$9 \times 10^{-5}$	$1.6180 \times 10^3$
31.51920	TM <sub>40,1</sub>	$1.6 \times 10^{-4}$	$7.1606 \times 10^2$
31.77492	TE <sub>41,1</sub>	$7 \times 10^{-5}$	$2.2132 \times 10^3$
32.23634	TM <sub>41,1</sub>	$9 \times 10^{-5}$	$9.8236 \times 10^2$
32.49037	TE <sub>42,1</sub>	$5 \times 10^{-5}$	$2.9724 \times 10^3$
32.95280	TM <sub>42,1</sub>	$7 \times 10^{-5}$	$1.3406 \times 10^3$
33.20523	TE <sub>43,1</sub>	$4 \times 10^{-5}$	$4.0458 \times 10^3$
33.66861	TM <sub>43,1</sub>	$5 \times 10^{-5}$	$1.8190 \times 10^3$
33.91953	TE <sub>44,1</sub>	$2.5 \times 10^{-5}$	$5.4110 \times 10^3$
34.38381	TM <sub>44,1</sub>	$4 \times 10^{-5}$	$2.3784 \times 10^3$
34.63329	TE <sub>45,1</sub>	$1.5 \times 10^{-5}$	$6.0292 \times 10^3$
35.098414	TM <sub>45,1</sub>	$2.6 \times 10^{-5}$	$3.3648 \times 10^3$
35.346540	TE <sub>46,1</sub>	$1.4 \times 10^{-5}$	$9.8984 \times 10^3$
35.812464	TM <sub>46,1</sub>	$1.8 \times 10^{-5}$	$4.5974 \times 10^3$
36.059292	TE <sub>47,1</sub>	$1.0 \times 10^{-5}$	$1.3784 \times 10^4$
36.525976	TM <sub>47,1</sub>	$1.4 \times 10^{-5}$	$6.2646 \times 10^3$
36.771567	TE <sub>48,1</sub>	$7 \times 10^{-6}$	$1.8293 \times 10^4$
37.238972	TM <sub>48,1</sub>	$1.0 \times 10^{-5}$	$8.2926 \times 10^3$
37.483382	TE <sub>49,1</sub>	$5 \times 10^{-6}$	$2.4772 \times 10^4$
37.951476	TM <sub>49,1</sub>	$7 \times 10^{-6}$	$1.1449 \times 10^4$
38.194757	TE <sub>50,1</sub>	$4 \times 10^{-6}$	$3.2575 \times 10^4$
38.663506	TM <sub>50,1</sub>	$4 \times 10^{-6}$	$1.6154 \times 10^4$
38.905705	TE <sub>51,1</sub>	$3 \times 10^{-6}$	$4.2101 \times 10^4$
39.375079	TM <sub>51,1</sub>	$4 \times 10^{-6}$	$2.2179 \times 10^4$
39.616241	TE <sub>52,1</sub>	$2 \times 10^{-6}$	$5.6913 \times 10^4$

Table 1: Primary electromagnetic resonances for dielectric spheres with size parameters ranging from 25 to 40 and index of refraction  $\sqrt{\epsilon} = 1.47 + 0i$ .

$x_{\text{res}}$	assignment	FWHM	$U_E^{(w)}/U_E^{(i)}$
376.9137048	TE <sub>397</sub> ,?	$7.2 \times 10^{-5}$	$1.6094 \times 10^2$
376.964433580738	TE <sub>418</sub> ,?	$5.1 \times 10^{-12}$	$2.0960 \times 10^9$
376.982613439	TM <sub>409</sub> ,?	$1.6 \times 10^{-8}$	$3.5960 \times 10^5$
377.05465079070317	TM <sub>422</sub> ,?	$1.4 \times 10^{-13}$	$4.2880 \times 10^{10}$

Table 2: Some electromagnetic resonances for dielectric spheres with size parameters ranging from 376.9 to 377.1 and index of refraction  $\sqrt{\epsilon} = 1.47 + 0i$ .

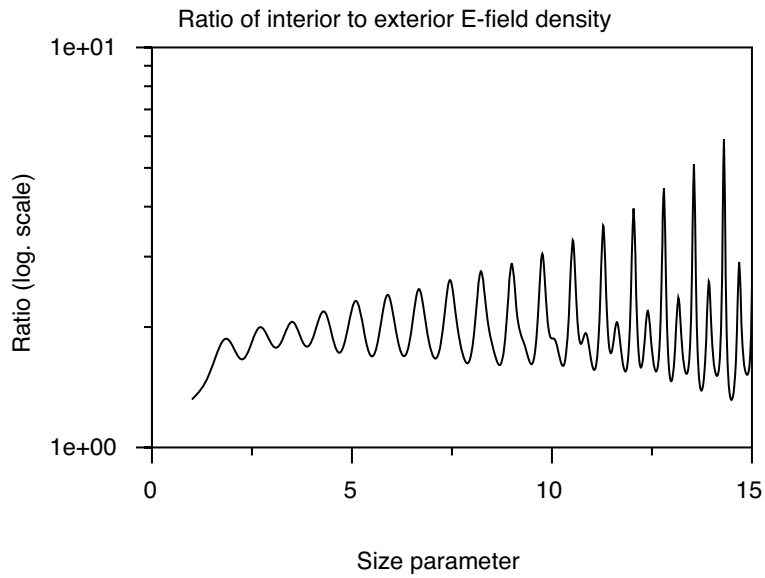


Figure 1: Semilogarithmic plot of the ratio of the interior electric-field density (Eq. (57)) to the exterior electric-field density  $E_0^2/2$  for real size parameters between 1 and 15. The calculation was performed by program RES. Index of refraction =  $1.47 + 0i$ ; step size  $\Delta x = 10^{-2}$ .

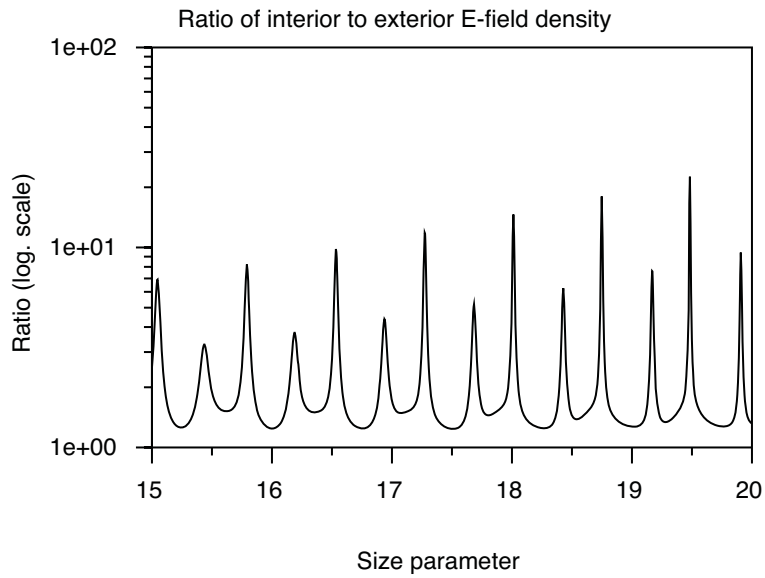


Figure 2: Semilogarithmic plot of the ratio of the interior electric-field density (Eq. (57)) to the exterior electric-field density  $E_0^2/2$  for real size parameters between 15 and 20. The calculation was performed by program RES. Index of refraction =  $1.47 + 0i$ ; step size  $\Delta x = 10^{-3}$ .

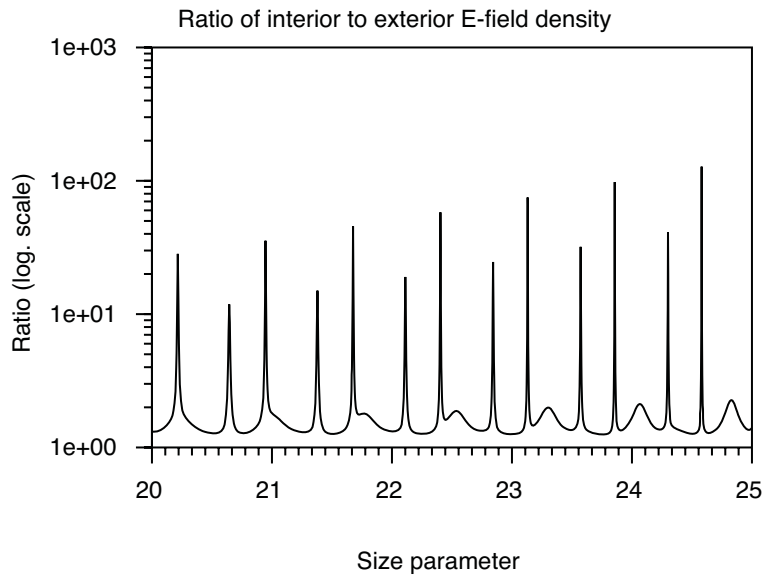


Figure 3: Semilogarithmic plot of the ratio of the interior electric-field density (Eq. (57)) to the exterior electric-field density  $E_0^2/2$  for real size parameters between 20 and 25. The calculation was performed by program RES. Index of refraction =  $1.47 + 0i$ ; step size  $\Delta x = 10^{-4}$ .

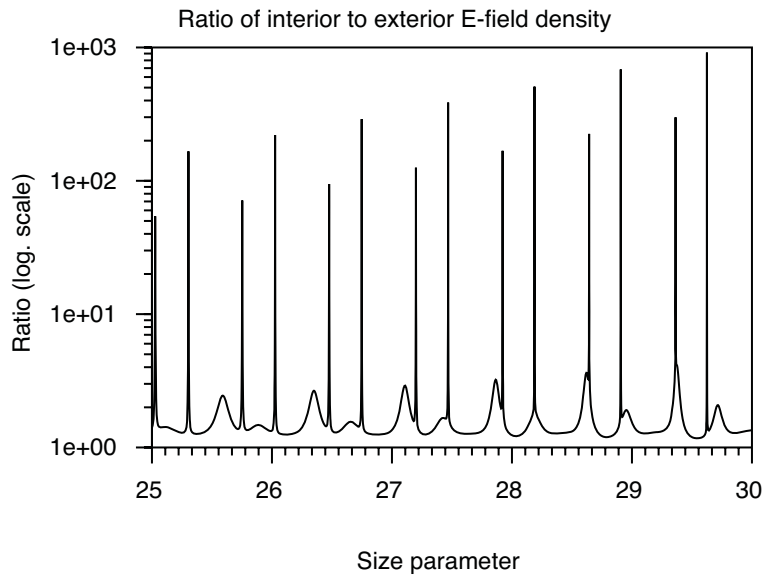


Figure 4: Semilogarithmic plot of the ratio of the interior electric-field density (Eq. (57)) to the exterior electric-field density  $E_0^2/2$  for real size parameters between 25 and 30. The calculation was performed by program RES. Index of refraction =  $1.47 + 0i$ ; step size  $\Delta x = 10^{-5}$ .

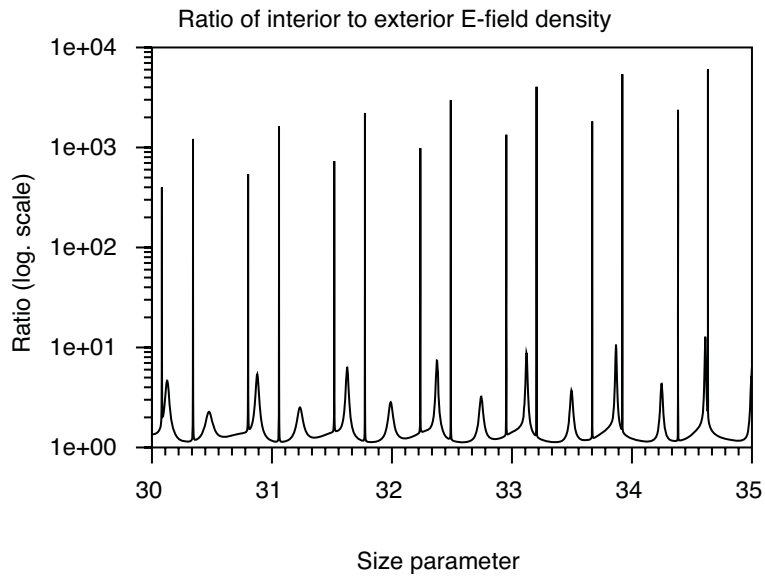


Figure 5: Semilogarithmic plot of the ratio of the interior electric-field density (Eq. (57)) to the exterior electric-field density  $E_0^2/2$  for real size parameters between 30 and 35. The calculation was performed by program RES. Index of refraction =  $1.47 + 0i$ ; step size  $\Delta x = 10^{-5}$ .

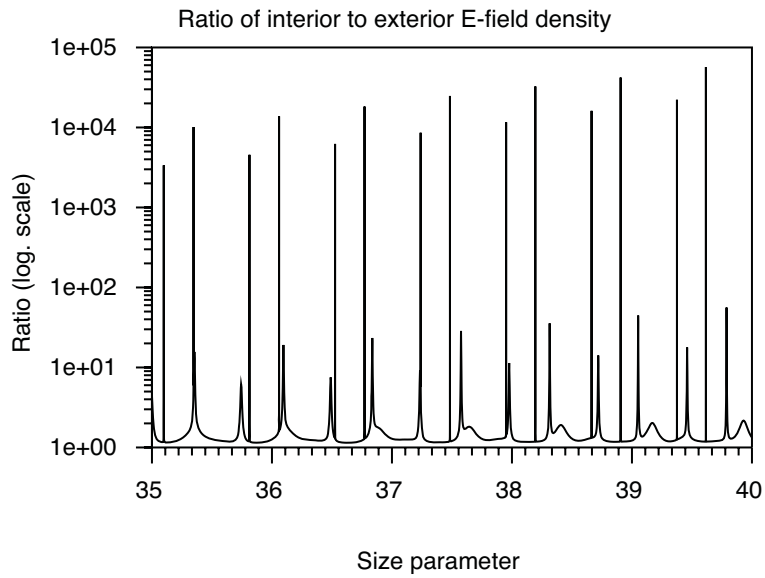


Figure 6: Semilogarithmic plot of the ratio of the interior electric-field density (Eq. (57)) to the exterior electric-field density  $E_0^2/2$  for real size parameters between 35 and 40. The calculation was performed by program RES. Index of refraction =  $1.47 + 0i$ ; step size  $\Delta x = 10^{-6}$ .

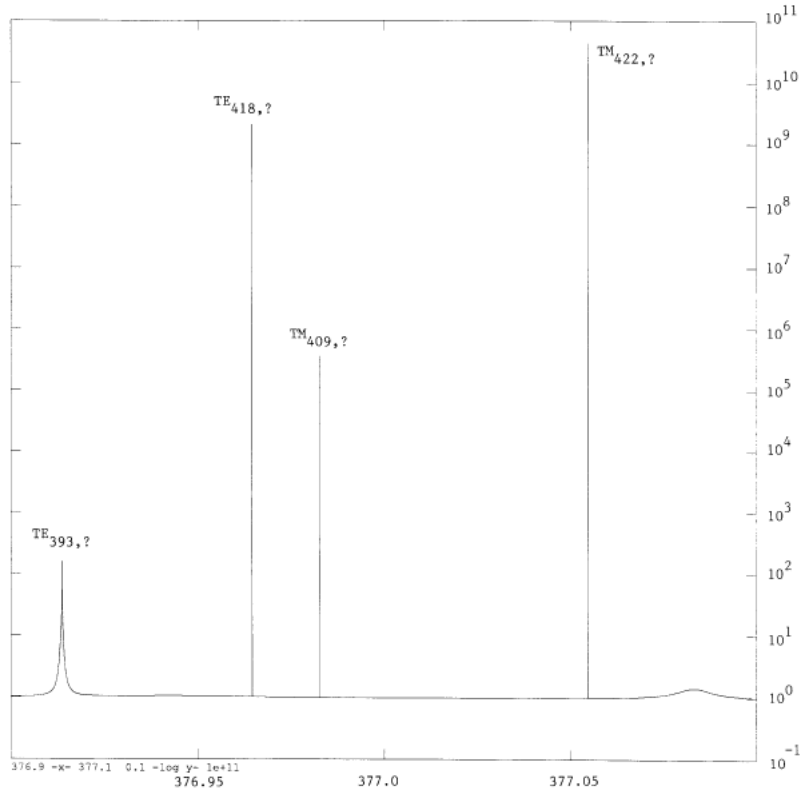


Figure 7: Semilogarithmic plot of the ratio of the interior electric-field density (Eq. (57)) to the exterior electric-field density  $E_0^2/2$  for real size parameters between 376.9 and 377.1. The calculation was performed by program RES. Index of refraction =  $1.47 + 0i$ ; step size  $\Delta x$  was varied between  $10^{-5}$  and  $10^{-13}$ , depending on the resolution required, as determined from Table 2.

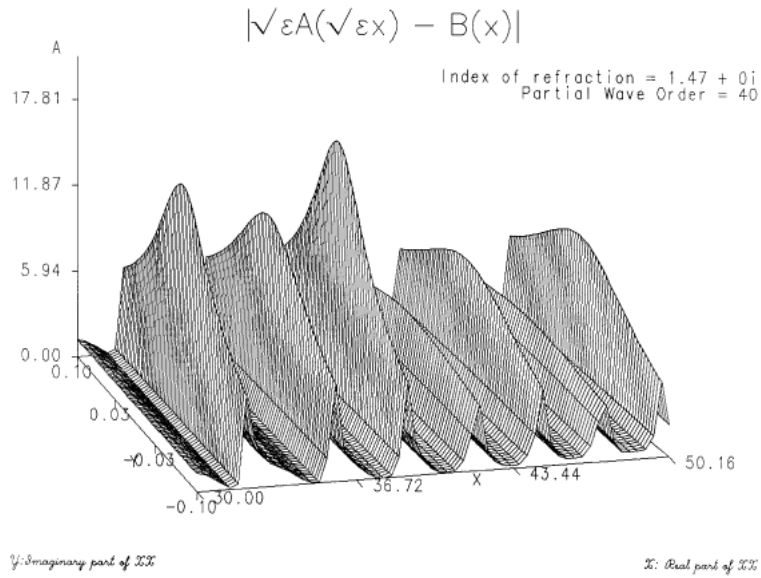


Figure 8: Surface plot of  $|f_{TE}^{(40)}(x)|$  (Eq. (69)) for complex values of  $x$  (shown as  $XX$  in the Figure) near the primary resonance at  $x = 31.05886$ . Index of refraction =  $1.47 + 0i$ .



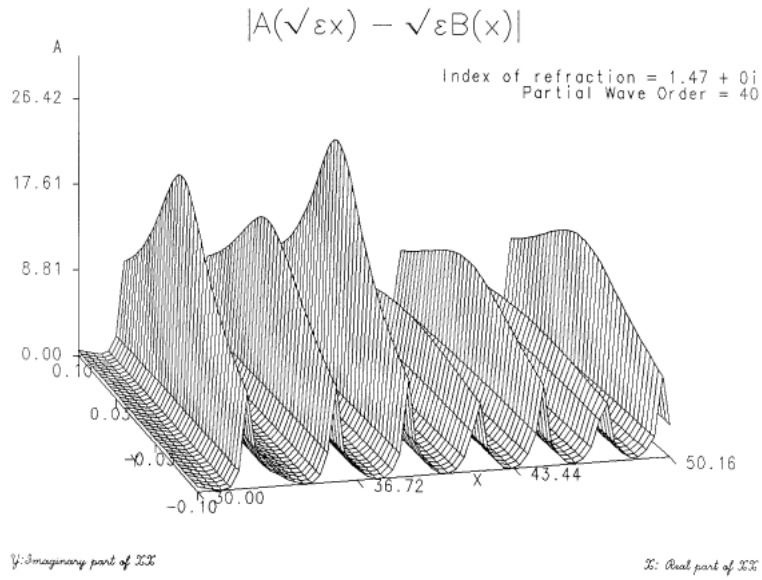


Figure 10: Surface plot of  $|f_{TM}^{(40)}(x)|$  (Eq. (68)) for complex values of  $x$  (shown as  $XX$  in the Figure) near the primary resonance at  $x = 31.51920$ . Index of refraction =  $1.47 + 0i$ .

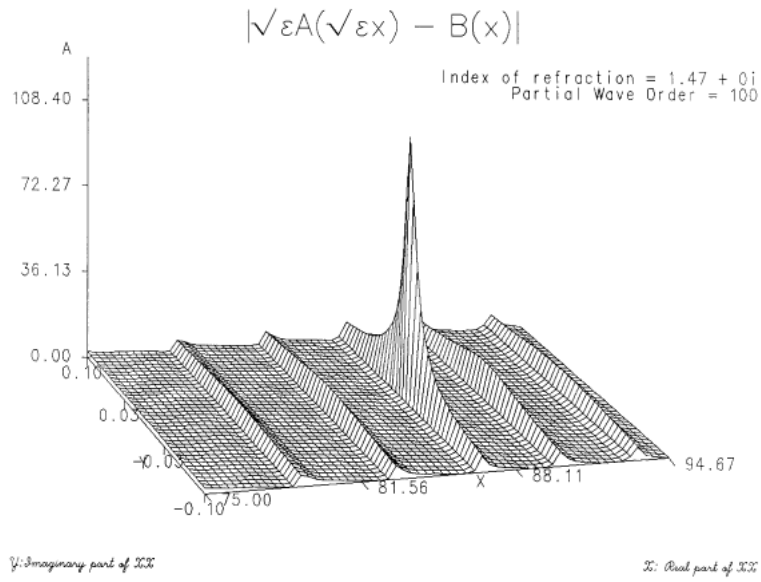


Figure 11: Surface plot of  $|f_{TE}^{(40)}(x)|$  (Eq. (69)) for complex values of  $x$  (shown as  $XX$  in the Figure) near the primary resonance at  $x = 78.0666114704$ . Index of refraction =  $1.47 + 0i$ .

# Appendix A

Program RES



```

C*****
PROGRAM RES
C   CALCULATION OF RESONANCES FOR A SPHERICAL DROPLET
C   PROGRAMMED IN MARCH AND JULY 1988
C   COPYRIGHT BY C. D. CANTRELL, 1988
C   ALL RIGHTS RESERVED
C   USES DOWNWARD RECURRENCE FOR CALCULATION OF THE SPHERICAL BESSEL
C   FUNCTIONS AND UPWARD RECURRENCE FOR CALCULATION OF THE SPHERICAL
C   HANKEL FUNCTIONS, FOLLOWING J. V. DAVE.
C   CALCULATES THE RATIO OF ANGLE-AVERAGED ENERGY DENSITY INSIDE A
C   DIELECTRIC SPHERE TO THE INCIDENT ENERGY DENSITY AS A FUNCTION
C   OF REAL SIZE PARAMETER.
C*****
C   LAST MODIFICATION 29 JULY 1988
C*****
C   DICTIONARY OF MAJOR VARIABLES
C*****
C   A      -- THE ELECTRIC (TRANSVERSE MAGNETIC) PARTIAL-WAVE AMPLITUDE
C           a sub n. CALCULATED IN SUBROUTINE PWAVE. COMPLEX*16
C           VECTOR, DIMENSIONS 0 TO MAXN. ARRAY INDEX IS EQUAL TO
C           PARTIAL-WAVE ORDER.
C   ACAP   -- LOGARITHMIC DERIVATIVE OF THE SPHERICAL BESSEL FUNCTION OF
C           THE FIRST KIND JUST INSIDE THE DROPLET; CALCULATED BY THE
C           SUBROUTINE SPHBESS. COMPLEX*16 VECTOR, DIMENSIONS
C           0 TO MAXN.
C           ARRAY INDEX IS EQUAL TO THE PARTIAL-WAVE ORDER.
C   ACAPR  -- LOGARITHMIC DERIVATIVE OF THE SPHERICAL BESSEL FUNCTION OF
C           THE FIRST KIND JUST OUTSIDE THE DROPLET; CALCULATED BY THE
C           SUBROUTINE SPHBESS. REAL*8 VECTOR, DIMENSIONS 0 TO MAXN.
C           ARRAY INDEX IS EQUAL TO THE PARTIAL-WAVE ORDER.
C   B      -- THE MAGNETIC (TRANSVERSE ELECTRIC) PARTIAL-WAVE AMPLITUDE
C           b sub n. CALCULATED IN SUBROUTINE PWAVE. COMPLEX*16
C           VECTOR, DIMENSIONS 0 TO MAXN. ARRAY INDEX IS EQUAL TO
C           PARTIAL-WAVE ORDER.
C   CHG    -- THE CHANGE IN ENDENS THAT MUST OCCUR BEFORE THE RESULTS
C           ARE WRITTEN TO DISK.
C   DE     -- DENOMINATOR OF THE ELECTRIC (TRANSVERSE MAGNETIC) PARTIAL-
C           WAVE AMPLITUDE a sub n. CALCULATED IN SUBROUTINE PWAVE.
C           COMPLEX*16 VECTOR, DIMENSIONS 0 TO MAXN.
C           ARRAY INDEX IS EQUAL TO THE PARTIAL-WAVE ORDER.
C   DLAST  -- THE VALUE OF ENDENS MOST RECENTLY WRITTEN TO DISK. REAL*8.
C   DM     -- DENOMINATOR OF THE MAGNETIC (TRANSVERSE ELECTRIC) PARTIAL-
C           WAVE AMPLITUDE b sub n. CALCULATED IN SUBROUTINE PWAVE.
C           COMPLEX*16 VECTOR, DIMENSIONS 0 TO MAXN.
C           ARRAY INDEX IS EQUAL TO THE PARTIAL-WAVE ORDER.
C   DX     -- INTERVAL BETWEEN SUCCESSIVE VALUES OF THE SIZE PARAMETER

```

```

C          AT WHICH ENDENS WILL BE CALCULATED (INPUT). REAL*8.
C  DXLAST  -- DIFFERENCE BETWEEN THE PRESENT VALUE OF X AND THE VALUE FOR
C          WHICH RESULTS WERE MOST RECENTLY WRITTEN TO DISK.
C  ENDENS  -- RATIO OF ANGLE-AVERAGED ENERGY DENSITY JUST INSIDE A
C          DIELECTRIC SPHERE TO THE INCIDENT ENERGY DENSITY.
C          CALCULATED IN SUBROUTINE DENS. TYPE REAL*8.
C  H       -- SPHERICAL HANKEL FUNCTION EVALUATED JUST OUTSIDE THE
C          SPHERE, WHERE IT IS ASSUMED THAT THE INDEX OF REFRACTION
C          IS EQUAL TO  $1 + 0i$ . CALCULATED AS  $JR + iy$  IN SUBROUTINE
C          SPHBESS. COMPLEX*16 VECTOR, DIMENSIONS 0 TO MAXN.
C          ARRAY INDEX IS EQUAL TO THE PARTIAL-WAVE ORDER.
C  J       -- SPHERICAL BESSEL FUNCTION EVALUATED JUST INSIDE THE
C          SPHERE. CALCULATED IN SUBROUTINE SPHBESS. COMPLEX*16
C          VECTOR, DIMENSIONS 0 TO MAXN.
C          ARRAY INDEX IS EQUAL TO THE PARTIAL-WAVE ORDER.
C  JR      -- SPHERICAL BESSEL FUNCTION EVALUATED JUST OUTSIDE THE
C          SPHERE, WHERE IT IS ASSUMED THAT THE INDEX OF REFRACTION
C          IS EQUAL TO  $1 + 0i$ . REAL*8 VECTOR, DIMENSIONS 0 TO MAXN.
C          CALCULATED IN SUBROUTINE SPHBESS.
C          ARRAY INDEX IS EQUAL TO THE PARTIAL-WAVE ORDER.
C  MAXN    -- MAXIMUM SIZE OF ARRAYS. SET VIA PARAMETER STATEMENT IN
C          MAIN PROGRAM TO 7000. THE MAXIMUM SIZE PARAMETER IS
C           $MAXN/(1.1|RF|)$ , WHERE RF IS THE COMPLEX INDEX OF
C          REFRACTION.
C  NMAX    -- HIGHEST ORDER AT WHICH ONE MAY USE ACAP, ACAPR, J OR JR.
C          CALCULATED IN MAIN PROGRAM AS  $|RF*XLO|$ . INTEGER.
C          USED IN SUBROUTINE DENS.
C  NMX1    -- STARTING ORDER FOR DOWNWARD RECURSION THAT DETERMINES ACAP
C          AND ACAPR IN SUBROUTINE SPHBESS. CALCULATED AS  $1.1*X*SQRT$ 
C           $(RFR**2 + RFI**2)$  OR 150, WHICHEVER IS LARGER. INTEGER.
C  NMX2    -- HIGHEST ORDER AT WHICH ONE MAY MAKE MEANINGFUL DIAGNOSTIC
C          WRITES OF ACAP, ACAPR, J OR JR. CALCULATED IN SUBROUTINE
C          SPHBESS AS  $|RF*X|$  OR 135, WHICHEVER IS LARGER. INTEGER.
C  RAT     -- RATIO OF THE PRESENT VALUE OF ENDENS TO THE VALUE MOST
C          RECENTLY WRITTEN TO DISK. REAL*8.
C  RFI     -- IMAGINARY PART OF THE INDEX OF REFRACTION (INPUT). MUST
C          BE  $\geq 0$  TO REPRESENT ABSORPTION. REAL*8.
C  RFR     -- REAL PART OF THE INDEX OF REFRACTION (INPUT). REAL*8.
C  SEP     -- THE GREATEST SEPARATION BETWEEN VALUES OF X FOR WHICH
C          RESULTS WILL BE WRITTEN. REAL*8.
C  TOL1    -- CALCULATED AS  $1 - CHG$ . USED AS THE LOWER LIMIT OF THE
C          VALUES RAT WILL BE ALLOWED TO HAVE BEFORE RESULTS
C          ARE WRITTEN TO DISK. REAL*8.
C  TOL2    -- CALCULATED AS  $1 + CHG$ . USED AS THE UPPER LIMIT OF THE
C          VALUES RAT WILL BE ALLOWED TO HAVE BEFORE RESULTS
C          ARE WRITTEN TO DISK. REAL*8.

```

```

C   XHI      -- UPPER LIMIT OF REAL SIZE PARAMETER FOR WHICH ENDENS WILL
C             BE CALCULATED (INPUT).  REAL*8.
C   XLAST    -- THE VALUE OF X FOR WHICH RESULTS WERE MOST RECENTLY
C             WRITTEN TO DISK.  REAL*8.
C   XLO      -- LOWER LIMIT OF REAL SIZE PARAMETER FOR WHICH ENDENS WILL
C             BE CALCULATED (INPUT).  REAL*8.
C   Y        -- SPHERICAL NEUMANN FUNCTION EVALUATED JUST OUTSIDE THE
C             SPHERE, WHERE IT IS ASSUMED THAT THE INDEX OF REFRACTION
C             IS EQUAL TO 1 + Oi.  REAL*8 VECTOR, DIMENSIONS 0 TO MAXN.
C             CALCULATED IN SUBROUTINE SPHBESS.
C             ARRAY INDEX IS EQUAL TO THE PARTIAL-WAVE ORDER.
C*****
C*****SET DIMENSION OF ARRAYS
C             PARAMETER (MAXN = 7000)
C             REAL*8 RFR,RFI,X,Y(0:MAXN),JR(0:MAXN),TEST,TEST1
C             REAL*8 XLO,XHI,DX,ACAPR(0:MAXN),ENDENS,TOL1,TOL2
C             REAL*8 RAT,SEP,XLAST,DLAST,DXLAST
C             COMPLEX*16 J(0:MAXN),ACAP(0:MAXN),H(0:MAXN),DE(0:MAXN),
C             1 DM(0:MAXN),A(0:MAXN),B(0:MAXN)
C*****HARDWIRED PARAMETERS: INITIAL LAST X AND DENSITY; DEFAULT SEPARATION
C             BETWEEN VALUES OF X FOR WHICH RESULTS ARE WRITTEN; THE CHANGE IN
C             ENDENS THAT MUST OCCUR BEFORE RESULTS ARE WRITTEN:
C             XLAST = 1.D-2
C             DLAST = 1.D-2
C             SEP = 2.D-3
C             CHG = 1.D-2
C             TOL1 = 1.DO - CHG
C             TOL2 = 1.DO + CHG
C*****INPUT PARAMETERS: REAL & IMAGINARY PARTS OF REFRACTIVE INDEX;
C             LOWER AND UPPER LIMITS OF SIZE PARAMETER; INCREMENT OF SIZE
C             PARAMETER
C             1 WRITE (*,515)
C             READ (*,*) RFR
C             WRITE (*,516)
C             READ (*,*) RFI
C             IF ( RFR.LE.0.DO .OR. RFI.LT.0.DO ) THEN
C             WRITE (9,514)
C             GO TO 1
C             END IF
C             WRITE (*,517)
C             READ (*,*) XLO
C             WRITE (*,518)
C             READ (*,*) XHI
C             WRITE (*,519)
C             READ (*,*) DX
C             TEST = (RFR*RFR + RFI*RFI) * XLO*XLO

```

```

TEST = DSQRT (TEST)
NTEST = TEST
IF (NTEST.LT.135) THEN
NMAX = 135
ELSE
NMAX = NTEST
END IF
NMAX = NMAX - 1
TEST1 = (RFR*RFR + RFI*RFI) * XHI * XHI
TEST1 = DSQRT (TEST1)
NTEST1 = 1.1D0 * TEST1
IF (NTEST1.GT.MAXN) THEN
WRITE (*,521)
GO TO 1000
END IF
2 WRITE (*,522)
READ (*,*) IDENS
IF (IDENS.NE.1.AND.IDENS.NE.2) GO TO 2
WRITE (*,520)
WRITE(9,*) RFR,RFI,XLO,XHI,DX,NMAX,IDENS
C*****STEP SIZE PARAMETER IN INCREMENTS DX
NX=0
IF (DX.EQ.0.D0) GO TO 15
NX = (XHI - XLO) / DX
15 CONTINUE
DO 100 IX=0,NX
X = XLO + IX*DX
C*****CALCULATION OF SPHERICAL BESSEL FUNCTIONS
CALL SPHBESS(X,RFR,RFI,ACAP,ACAPR,J,JR,Y,H,MAXN)
C*****EVALUATE MIE PARTIAL-WAVE AMPLITUDES (A,B) AND RESONANT DENOMINATORS
C (DE,DM)
CALL PWAVE(X,RFR,RFI,ACAP,ACAPR,J,JR,Y,H,DE,DM,A,B,MAXN)
C*****EVALUATE ENERGY DENSITY JUST INSIDE SURFACE OF SPHERE
CALL DENS(J,DE,DM,X,RFR,RFI,ENDENS,NMAX,MAXN,IDENS)
C*****WRITE RESULTS
DXLAST = X - XLAST
RAT = ENDENS / DLAST
IF (RAT.LE.TOL1.OR.RAT.GE.TOL2.OR.DXLAST.GE.SEP) THEN
XLAST = X
DLAST = ENDENS
WRITE (9,530) X,ENDENS
END IF
C WRITE (9,513) (A(NN),NN,NN=1,NMAX)
C WRITE (9,513) (B(NN),NN,NN=1,NMAX)
100 CONTINUE
513 FORMAT(2(1PD30.15),I10)

```

```

514 FORMAT('ILLEGAL VALUE FOR REFRACTIVE INDEX')
515 FORMAT('Please enter the real part of the refractive index:')
516 FORMAT('Please enter the imaginary part of the refractive index:'
1 )
517 FORMAT('Please enter the lower limit of the size parameter:')
518 FORMAT('Please enter the upper limit of the size parameter:')
519 FORMAT('Please enter the step size:')
520 FORMAT('Calculating...')
521 FORMAT('SIZE PARAMETER TOO LARGE FOR ARRAY DIMENSIONS!'/
1 'Please see documentation under MAXN')
522 FORMAT('Please enter 1 for total EM energy density '/
1 'or 2 for E-field density:')
530 FORMAT(2F30.15)
1000 STOP
END
C*****
SUBROUTINE SPHBESS (X,RFR,RFI,ACAP,ACAPR,J,JR,Y,H,MAXN)
C*****CALCULATION OF SPHERICAL BESSEL FUNCTIONS
C JN IS CALCULATED FOR COMPLEX ARGUMENT USING THE LOGARITHMIC DERIVATIVE
C AS FIRST DONE BY INFELD (CODING BY J. V. DAVE)
C YN IS CALCULATED BY UPWARD RECURRENCE
C HN(1) IS THEN CALCULATED AS JN + SQRT(-1) * YN
C ADAPTED (WITH EXTENSIONS) IN MARCH 1988 FROM ROUTINE DBMIE OF
C J. V. DAVE (1968)
C IN PARTICULAR, DAVE USED EXP(-IKR)/R FOR OUTGOING WAVES, SO HE
C USED THE HANKEL FUNCTION OF THE SECOND KIND AND A COMPLEX
C REFRACTIVE INDEX WITH NEGATIVE IMAGINARY PART. THIS ROUTINE
C USES EXP(+IKR)/R FOR OUTGOING WAVES, HENCE HANKEL FUNCTIONS
C OF THE FIRST KIND AND A COMPLEX REFRACTIVE INDEX WITH
C POSITIVE REAL PART.
C*****COPYRIGHT C. D. CANTRELL, 1988
REAL*8 X,RX,RFR,RFI,T,JRM1,ACAPR(0:MAXN)
REAL*8 RFRX,RFIX,YM1,YM2,Y(0:MAXN),JR(0:MAXN)
COMPLEX*16 RF,RRF,RRFX,ACAP(0:MAXN)
COMPLEX*16 J(0:MAXN),JNM1,H(0:MAXN)
C*****COMPLEX INDEX OF REFRACTION & ITS RECIPROCAL
RF = DCMPLX(RFR,RFI)
RRF = 1.DO/RF
RX = 1.DO/X
RRFX = RRF*RX
RFRX=RFR*X
RFIX=RFI*X
C*****CALCULATE UPPER LIMIT OF ORDER FOR DOWNWARD RECURRENCE
T = (X**2)*(RFR**2 + RFI**2)
T = DSQRT(T)
MAXNM1 = MAXN - 1

```

```

        NMX2 = T
        NMX1 = 1.1D0*T
        IF( NMX1.LE.MAXNM1 ) GO TO 21
        WRITE (9,515)
        RETURN
21 CONTINUE
        IF( NMX1.GT.150 ) GO TO 22
        NMX1 = 150
        NMX2 = 135
C*****BEGIN DOWNWARD RECURRENCE FOR THE LOGARITHMIC DERIVATIVE OF JN(RF*X)
22 ACAP(NMX1 + 1) = ( 0.DO, 0.DO )
        NMX1P1 = NMX1 + 1
        DO 23 N = 1, NMX1P1
        NN = NMX1 - N + 1
        ACAP(NN) = ( NN + 1 )*RRFX - 1.DO / ((NN+1)*RRFX + ACAP(NN+1))
23 CONTINUE
C        WRITE (9,513) (ACAP(NN), NN, NN=0, NMX2)
C*****CALCULATION OF JN(RX*X) BY UPWARD RECURRENCE USING THE LOG. DERIVATIVE
        J(0)=DCMLPX (DSIN(RFRX)*DCOSH(RFIX) , DCOS(RFRX)*DSINH(RFIX))
1 *RRFX
        JNM1=J(0)
        DO 24 NN=1, NMX1
        J(NN)=JNM1/(ACAP(NN)+NN*RRFX)
        JNM1=J(NN)
24 CONTINUE
C        WRITE (9,513) (J(NN), NN, NN=0, NMX2)
C*****CALCULATION OF NEUMANN FUNCTIONS FOR REAL ARGUMENT BY UPWARD RECURRENCE
        YM2 = DSIN(X) * RX
        YM1 = -DCOS(X) * RX
        Y(0) = YM1
        DO 25 NN=1, NMX2
        Y(NN) = (2*NN - 1) * RX * YM1 - YM2
        YM2=YM1
        YM1=Y(NN)
25 CONTINUE
C        WRITE (9,514) (Y(NN), NN, NN=0, NMX2)
C*****CALCULATION OF BESSEL FUNCTIONS OF REAL ARGUMENT FOR USE IN H(1)
        ACAPR(NMX1 + 1) = 0.DO
        DO 26 N = 1, NMX1P1
        NN = NMX1 - N + 1
        ACAPR(NN) = ( NN + 1 )*RX - 1.DO / ((NN+1)*RX + ACAPR(NN+1))
26 CONTINUE
C        WRITE (9,514) (ACAPR(NN), NN, NN=0, NMX2)
C*****CALCULATION OF REAL JN(X) BY UPWARD RECURRENCE USING THE LOG. DERIVATIVE
        JR(0)= DSIN(X)*RX
        JRM1=JR(0)

```

```

DO 27 NN=1,NMX1
JR(NN)=JRM1/(ACAPR(NN)+NN*RX)
JRM1=JR(NN)
27 CONTINUE
C WRITE (9,514) (JR(NN),NN,NN=0,NMX2)
C*****CALCULATION OF HANKEL FUNCTION (1) USING NEUMANN & BESSEL FUNCTIONS
DO 30 NN=0,NMX2
H(NN) = DCMPLX( JR(NN) , Y(NN) )
30 CONTINUE
C WRITE (9,513) (H(NN),NN,NN=0,NMX2)
513 FORMAT(2(1PD30.15),I10)
514 FORMAT(1PD30.15,I10)
515 FORMAT('DIMENSION TOO SMALL FOR SIZE PARAMETER')
RETURN
END
C*****
SUBROUTINE PWAVE(X,RFR,RFI,ACAP,ACAPR,J,JR,Y,H,DE,DM,A,B,
1 MAXN)
REAL*8 RFR,RFI,X,Y(O:MAXN),JR(O:MAXN)
REAL*8 T,REB,ACAPR(O:MAXN)
COMPLEX*16 J(O:MAXN),ACAP(O:MAXN),H(O:MAXN),DE(O:MAXN),
1 DM(O:MAXN),RF,RRF,UE,UM,A(O:MAXN),B(O:MAXN),HNM1
C*****CALCULATION OF PARTIAL-WAVE AMPLITUDES AN AND BN FOR MIE
C SCATTERING
C*****EMPLOYS FORMULAS DERIVED BY C. D. CANTRELL AND LOGARITHMIC
C DERIVATIVE OF SPHERICAL BESSEL FUNCTIONS OF THE FIRST KIND
C (ARRAYS ACAP AND ACAPR) CALCULATED BY METHOD OF J. V. DAVE
C*****COPYRIGHT 1988 BY C. D. CANTRELL
C*****COMPLEX REFRACTIVE INDEX
RF = DCMPLX(RFR,RFI)
RRF = 1.DO/RF
C*****CALCULATION OF RESONANT DENOMINATORS
C*****FIRST CALCULATE UPPER LIMIT OF ORDER FOR LONGEST WAVELENGTH
T = (X**2)*(RFR**2 + RFI**2)
T = DSQRT(T)
NMX1 = 1.1DO*T
NMX2 = T
IF( NMX1.GT.150 ) GO TO 10
NMX1 = 150
NMX2 = 135
10 CONTINUE
C*****HANKEL (1) FUNCTION OF ORDER -1
HNM1 = DCMPLX(DCOS(X),DSIN(X)) / X
DO 90 NN=0,NMX2
C*****ELECTRIC HERTZ VECTOR PARTIAL-WAVE AMPLITUDE
C DENOMINATOR

```

```

          DE(NN) = ( RRF*ACAP(NN) + NN / X ) * H(NN) - HNM1
C          NUMERATOR
          UE = JR(NN) * ( RRF*ACAP(NN) - ACAPR(NN) )
C          AMPLITUDE
          A(NN) = UE / DE(NN)
C*****MAGNETIC HERTZ VECTOR PARTIAL-WAVE AMPLITUDE
C          DENOMINATOR
          DM(NN) = ( RF*ACAP(NN) + NN / X ) * H(NN) - HNM1
C          NUMERATOR
          UM = JR(NN) * ( RF*ACAP(NN) - ACAPR(NN) )
C          AMPLITUDE
          B(NN) = UM / DM(NN)
          HNM1 = H(NN)
    90 CONTINUE
C          WRITE (9,516) X,NMX2
C          WRITE (9,513) (DE(NN),NN,NN=0,NMX2)
C          WRITE (9,513) (DM(NN),NN,NN=0,NMX2)
C          WRITE (9,513) (A(NN),NN,NN=0,NMX2)
C          WRITE (9,513) (B(NN),NN,NN=0,NMX2)
513 FORMAT (2(1PD30.15),I10)
514 FORMAT (2(F30.15),I10)
515 FORMAT (3(F25.15))
516 FORMAT (1PD30.15,I10)
          RETURN
          END
C*****
          SUBROUTINE DENS (J,DE,DM,X,RFR,RFI,ENDENS,NMAX,MAXN,IDENS)
C*****CALCULATION OF THE RATIO OF THE ANGLE-AVERAGED ENERGY DENSITY JUST
C          INSIDE THE SURFACE OF THE SPHERE TO THE INCIDENT ENERGY DENSITY.
C          EMPLOYS EQUATIONS DERIVED BY C. D. CANTRELL
          COMPLEX*16 J(0:MAXN),DE(0:MAXN),DM(0:MAXN),CI,RF
          COMPLEX*16 CN,DN
          REAL*8 ENDENS,X,TEMP1,TEMP2,RFR,RFI
          DATA CI / ( 0.DO, 1.DO ) /
          PI = DACOS (-1.DO)
C*****COPYRIGHT 1988 BY C. D. CANTRELL
C          ALL RIGHTS RESERVED
          RF = DCMPLX ( RFR, RFI )
          ENDENS = 0.DO
C*****RATIO OF INTERIOR TO EXTERIOR TOTAL EM ENERGY DENSITY
C          NOTE THAT Re (epsilon) = (Re(n))**2 - (Im(n))**2
C          AND THAT | epsilon | = (Re(n))**2 + (Im(n))**2
C*****CN AND DN ARE THE INTERIOR PARTIAL-WAVE AMPLITUDES
          IF (IDENS.EQ.1) THEN
          DO 100 NN=1,NMAX
          CN = CI / (RF*RF*X*X*J(NN)*DE(NN))

```

```

DN = CI / (X*X*J(NN)*DM(NN))
TEMP1 = DCONJG (J(NN)) * J(NN) * (2*NN + 1)
TEMP2 = ((NN + 1) * DCONJG(J(NN-1)) * J(NN-1)
1 + NN * DCONJG(J(NN+1)) * J(NN+1)) * (RFR*RFR + RFI*RFI)
TEMP1 = TEMP1 * ( DCONJG(RF * RF * CN) * RF * RF * CN
1 + (RFR * RFR - RFI * RFI) * DCONJG(DN) * DN )
TEMP2 = TEMP2 * ( (RFR * RFR - RFI * RFI) * DCONJG(CN) * CN
1 + DCONJG(DN) * DN )
ENDENS = ENDENS + TEMP1 + TEMP2
100 CONTINUE
ENDENS = ENDENS / 4.DO
GO TO 300
C*****RATIO OF INTERIOR TO EXTERIOR ELECTRIC-FIELD DENSITY
ELSE IF (IDENS.EQ.2) THEN
DO 200 NN=1,NMAX
CN = CI / (RF*RF*X*X*J(NN)*DE(NN))
DN = CI / (X*X*J(NN)*DM(NN))
TEMP1 = DCONJG (J(NN)) * J(NN) * (2*NN + 1)
TEMP2 = ((NN + 1) * DCONJG(J(NN-1)) * J(NN-1)
1 + NN * DCONJG(J(NN+1)) * J(NN+1)) * (RFR*RFR + RFI*RFI)
TEMP1 = TEMP1 * DCONJG(DN) * DN
TEMP2 = TEMP2 * DCONJG(CN) * CN
ENDENS = ENDENS + TEMP1 + TEMP2
200 CONTINUE
END IF
300 CONTINUE
RETURN
END
C*****
C END OF PROGRAM RES
C*****

```

# Appendix B

Program FINDRES3

```

C*****
C      PROGRAM FINDRES3
C      LOCATION OF RESONANCES FOR A SPHERICAL DROPLET
C      PROGRAMMED IN MARCH-APRIL AND JULY 1988
C      COPYRIGHT BY C. D. CANTRELL, 1988
C      ALL RIGHTS RESERVED
C      THIRD ATTEMPT: USES THE LENTZ-WISCOMBE CONTINUED FRACTION FOR
C      THE LOG. DERIVATIVE OF THE SPHERICAL BESSEL FUNCTIONS AND
C      UPWARD RECURRENCE FOR THE LOG. DERIVATIVE OF THE SPHERICAL
C      HANKEL FUNCTIONS.  CALCULATES THE PARTIAL-WAVE-AMPLITUDE
C      DENOMINATORS USING ONLY THE LOGARITHMIC DERIVATIVES OF THE BESSEL
C      FUNCTIONS.  ZEROES FOUND USING ZXMIN APPLIED TO THE ABSOLUTE
C      VALUE OF THE DENOMINATOR.
C*****
C      DICTIONARY OF MAJOR VARIABLES
C*****
C      ACAP      -- LOGARITHMIC DERIVATIVE OF THE SPHERICAL BESSEL FUNCTION OF
C                THE FIRST KIND JUST INSIDE THE DROPLET; CALCULATED BY THE
C                CONTINUED-FRACTION ROUTINE CONFRA.  TYPE COMPLEX*16.
C      BCAP      -- LOGARITHMIC DERIVATIVE OF THE SPHERICAL HANKEL FUNCTION OF
C                THE FIRST KIND JUST OUTSIDE THE DROPLET; CALCULATED IN THE
C                UPWARD-RECURRENCE SUBROUTINE HANK.  TYPE COMPLEX*16.
C                ARRAY INDEX IS EQUAL TO THE PARTIAL-WAVE ORDER.
C      DE        -- DENOMINATOR OF THE ELECTRIC (TRANSVERSE MAGNETIC) PARTIAL-
C                WAVE AMPLITUDE  $a_{sub n}$ .  CALCULATED IN SUBROUTINE DENOM.
C                ZEROS FOUND BY SUBROUTINE ZANLYT.  TYPE COMPLEX*16.
C      DM        -- DENOMINATOR OF THE MAGNETIC (TRANSVERSE ELECTRIC) PARTIAL-
C                WAVE AMPLITUDE  $b_{sub n}$ .  CALCULATED IN SUBROUTINE DENOM.
C                ZEROS FOUND BY SUBROUTINE ZANLYT.  TYPE COMPLEX*16.
C      FUNCT     -- FUNCTION WHOSE ZEROS ARE TO BE LOCATED.  TYPE COMPLEX*16.
C                CALLED BY SUBROUTINE ZANLYT.  EQUATED TO THE ABSOLUTE
C                VALUE OF DE OR DM FOR PARTIAL-WAVE ORDER NORD IN
C                SUBROUTINE FUNCT.
C      IAB       -- INTEGER VARIABLE USED TO CHOOSE BETWEEN DE AND DM.  NAME
C                REFERS TO  $a_{sub n}$  OR  $b_{sub n}$  MIE-SCATTERING AMPLITUDES.
C      IAORB     -- INTEGER LOOP INDEX ON DE AND DM.  SEE IAB FOR NAME.
C      IER       -- INTEGER ERROR CODE UPON RETURN FROM SUBROUTINE ZXMIN.
C                IER = 129 IMPLIES THAT THE INITIAL HESSIAN USED BY ZXMIN
C                IS NOT POSITIVE DEFINITE; IER = 130 IMPLIES THAT THE
C                ITERATION WAS TERMINATED BECAUSE ROUNDING ERRORS BECAME
C                DOMINANT, SO THAT THE COMPONENTS OF X HAVE NOT BEEN
C                DETERMINED TO NSIG DIGITS; IER = 131 IMPLIES THAT THE
C                ITERATION WAS TERMINATED BECAUSE MAXFN WAS EXCEEDED.
C      MAXFN     -- MAXIMUM NUMBER OF ITERATIONS TO BE PERFORMED BY SUBROUTINE
C                ZXMIN IN ATTEMPTING TO FIND ANY ONE ROOT (RESONANCE)
C                (INPUT).  TYPE INTEGER.

```

```

C      NRES      -- NUMBER OF ROOTS (RESONANCES) TO BE FOUND FOR EACH PARTIAL-
C                WAVE ORDER AND EACH TYPE OF AMPLITUDE (A--ELECTRIC OR B--
C                MAGNETIC) (INPUT).  TYPE INTEGER.
C      NHI       -- UPPER LIMIT OF THE ORDER OF PARTIAL WAVES FOR WHICH
C                RESONANCES ARE TO BE FOUND (INTEGER, INPUT).
C      NLO       -- LOWER LIMIT OF THE ORDER OF PARTIAL WAVES FOR WHICH
C                RESONANCES ARE TO BE FOUND (INTEGER, INPUT).
C      NORD      -- PARTIAL-WAVE ORDER (INTEGER).  EQUATED TO LOOP INDEX IORD
C                IN MAIN PROGRAM.
C      NSIG      -- A ROOT IS ACCEPTED BY SUBROUTINE ZXMIN IF TWO SUCCESSIVE
C                APPROXIMATIONS AGREE TO THE FIRST NSIG DIGITS.
C                TYPE INTEGER.
C                >>>>NOTE!!!! FOR REASONS UNKNOWN TO ME, NSIG=3 PRODUCES
C                ANSWERS WITH UP TO 9 SIGNIFICANT DIGITS IN THE RESONANCE
C                POSITION'S REAL PART, XX(1).  IN THIS EXAMPLE THERE ARE
C                ONLY 3 SIG. DIGITS IN THE HALFWIDTH (IMAGINARY PART OF
C                THE RESONANCE POSITION), XX(2).  SIGNIFICANT DIGITS
C                >>INCLUDE<< ZEROES TO THE RIGHT OF THE DECIMAL POINT!!!
C      RF        -- COMPLEX*16 REFRACTIVE INDEX.  SINCE THE SPHERICAL HANKEL
C                FUNCTION OF THE FIRST KIND IS USED TO REPRESENT OUTGOING
C                WAVES, THE IMAGINARY PART OF THE COMPLEX REFRACTIVE INDEX
C                MUST BE NON-NEGATIVE TO REPRESENT ABSORPTION.  SET
C                IN MAIN PROGRAM TO DCMPLX (RFR,RFI).
C      RFR       -- REAL PART OF COMPLEX REFRACTIVE INDEX (INPUT).
C      RFI       -- IMAGINARY PART OF COMPLEX REFRACTIVE INDEX (INPUT).
C      RHO       -- RATIO OF H(1) SUB N+1 TO H(1) SUB N WITH ARGUMENT X.
C                CALCULATED BY UPWARD RECURRENCE IN SUBROUTINE HANK.
C                USED IN CALCULATION OF BCAP.  TYPE COMPLEX*16.
C                ARRAY INDEX IS EQUAL TO THE PARTIAL-WAVE ORDER.
C      RRF       -- COMPLEX*16 RECIPROCAL OF RF.  CALCULATED IN MAIN PROGRAM.
C      RRFX      -- COMPLEX*16 RECIPROCAL OF (RF*X).  CALCULATED IN FUNCTION F.
C      RX        -- COMPLEX*16 RECIPROCAL OF X.  CALCULATED IN FUNCTION F.
C      X         -- COMPLEX*16 SIZE PARAMETER (GENERALIZATION OF DROPLET
C                CIRCUMFERENCE DIVIDED BY WAVELENGTH).  NOW USED ONLY IN
C                DIAGNOSTIC ROUTINES, WHICH ARE BYPASSED WITH A GO TO
C                STATEMENT.
C      XIM       -- IMAGINARY PART OF THE GUESSTIMATE FOR A RESONANT VALUE
C                OF X.  TYPE REAL*8.  SET TO 0.DO
C      XINC      -- INCREMENT BETWEEN SUCCESSIVE STARTING VALUES OF THE
C                COMPLEX SIZE PARAMETER (INPUT).  TYPE REAL*8.
C      XORD      -- REAL*8 VALUE OF THE PARTIAL-WAVE ORDER.
C      XRE       -- INITIAL GUESS FOR THE REAL PART OF THE RESONANT X FOR
C                THE FIRST RESONANCE.  THE DEFAULT IS THE FOLLOWING
C                EMPIRICAL FORMULA:
C                FOR THE a sub n COEFFICIENT:
C                PI*XORD

```

```

C             XRE = -----
C             E*RFR
C             FOR THE b sub n COEFFICIENT:
C             PI*(XORD-.5)
C             XRE = -----
C             E*RFR
C   XREO      -- STARTING VALUE OF THE REAL PART OF THE SIZE PARAMETER
C             FOR ONE RESONANCE (INPUT). TYPE REAL*8.
C             IT IS STRONGLY SUGGESTED TO SET XREO TO 0.DO UNLESS THE
C             USER CANNOT OBTAIN SATISFACTORY RESULTS WITH THE
C             EMPIRICAL STARTING VALUES (SEE ABOVE, UNDER XRE).
C   XX        -- REAL*8 VECTOR OF LENGTH 2. INPUT BY THE USER, THEN SET
C             BY ZXMIN TO THE VALUE THAT MINIMIZES A RESONANT DENOMINATOR.
C             ON OUTPUT, XX(1) IS THE REAL PART OF THE RESONANT
C             VALUE OF THE SIZE PARAMETER, AND XX(2) IS THE IMAGINARY
C             PART. THE LATTER IS EQUAL TO MINUS THE HALF WIDTH AT
C             HALF MAXIMUM.
C*****
C   SOFTWARE REQUIRED
C*****
C   THIS PROGRAM MAKES USE OF THE DOUBLE-PRECISION VERSIONS OF THE
C   ROUTINES ZXMIN, ZXMJN, UERTST, UGETIO AND USPKD MARKETED AND
C   COPYRIGHTED BY IMSL, INC.
C*****
C   BEGINNING OF PROGRAM FINDRES3
C*****
REAL*8 RFR,RFI,XORD,E,PI,XREO,XIM,XRE,XX,F,G,H,W,XINC
COMPLEX*16 ACAP,BCAP(-1:7000),RHO(-1:7000),H1N,H1NM1,Z,CONFRA,
1 DE,DM,X,RF,RX,RRF,RRFX,CI,XXX,BESSJ,BESSJM1,RAT,ZINV
CHARACTER AB(2)
DIMENSION XX(2),H(3),G(2),W(6)
EXTERNAL FUNCT
COMMON/INT/NORD,IAB,ITER
COMMON/BESS/ACAP,BCAP,RHO,DE,DM,RX,ZINV
COMMON/INDEX/RFR,RFI,RF,RRF,RRFX
DATA AB/'a','b'/
DATA CI/(0.DO,1.DO)/
C*****INPUT PARAMETERS: REAL & IMAGINARY PARTS OF REFRACTIVE INDEX;
C   LOWER AND UPPER LIMITS OF PARTIAL-WAVE ORDER; NUMBER OF RESONANCES
C   TO FIND; MAXIMUM NUMBER OF ITERATIONS PER RESONANCE; NO. OF
C   SIGNIFICANT DIGITS DESIRED IN THE ANSWER; STARTING VALUE OF THE SIZE
C   PARAMETER; INCREMENT BETWEEN SUCCESSIVE STARTING VALUES.
WRITE (*,501)
WRITE (9,501)
1 WRITE (*,515)
READ (*,*) RFR

```

```

WRITE (*,516)
READ (*,*) RFI
IF ( RFR.LE.0.DO .OR. RFI.LT.0.DO ) THEN
WRITE (*,514)
GO TO 1
END IF
WRITE (*,517)
READ (*,*) NLO
WRITE (*,518)
READ (*,*) NHI
WRITE (*,519)
READ (*,*) NRES
WRITE (*,520)
READ (*,*) MAXFN
WRITE (*,521)
READ (*,*) XREO
WRITE (*,522)
READ (*,*) XINC
WRITE (*,523)
READ (*,*) NSIG
WRITE (*,524)
C*****WRITE HEADER FOR OUTPUT
WRITE(9,530) RFR,RFI,NLO,NHI,MAXFN,NSIG
C*****COMPLEX INDEX OF REFRACTION & ITS RECIPROCAL
RF = DCMPLX(RFR,RFI)
RRF = 1.DO/RF
C*****DEFAULT PARAMETERS FOR ROOTFINDER
C                               OPTIMIZATION LEVEL (IMPLIES CALCULATION
C                               OF HESSIAN MATRIX BY ZXMIN)
      IOPT = 3
C                               LENGTH OF VECTOR XX (ARGUMENT TO ZXMIN)
      N = 2
      PI = DACOS(-1.DO)
      E = DEXP (1.DO)
C*****LOOP ON ORDER OF PARTIAL WAVE
DO 100 IORD = NLO,NHI
C*****ORDER OF BESSEL FUNCTIONS; NORD TO BE PASSED TO FUNCTION F IN COMMON
      NORD = IORD
      XORD = IORD
C*****LOOP ON IAORB, WHICH CHOOSES THE A COEFFICIENT IF IAORB=1 AND THE
C      B COEFFICIENT IF IAORB=2
DO 90 IAORB = 1,2
C*****VALUE TO BE PASSED IN COMMON
      IAB = IAORB
C*****WRITE HEADER
WRITE (9,512) AB(IAB), NORD

```

```

C*****LOOP ON RESONANCE NUMBER
      DO 80 IRES = 1,NRES
        ITER = 0
C*****INITIALIZE GUESSES FOR RESONANCES
        XIM = 0.DO
        IF (XREO.EQ.0.DO) THEN
          IF (IAB.EQ.1) XRE = PI*XORD/(E*RFR)
          IF (IAB.EQ.2) XRE = PI*(XORD-.5DO)/(E*RFR)
        ELSE
          XRE = XREO
        END IF
        XX(1) = XRE + (IRES-1)*XINC
        XX(2) = XIM
C*****CALL MINIMIZATION ROUTINE
        CALL ZXMIN(FUNCT,N,NSIG,MAXFN,IOPT,XX,H,G,F,W,IER)
C      BYPASS DIAGNOSTICS
C      GO TO 400
C*****WRITE RESULTS
        WRITE (9,509) IRES,XX,ITER
C*****WRITE DIAGNOSTICS
C      WRITE (9,511) H
C      WRITE (9,511) G
C      WRITE (9,511) F
C*****END OF LOOP ON IRES
      80 CONTINUE
C*****END OF LOOP ON IAORB
      90 CONTINUE
C*****END OF LOOP ON IORD
      100 CONTINUE
C*****DIAGNOSTICS
      GO TO 500
      X = DCPLX (XX(1),XX(2))
      RX = 1.DO/X
      ZINV = RX
      Z = X
      H1NM1 = - CI * CDEXP(CI*Z) * ZINV
      WRITE (9,513) H1NM1
      CALL HANK
      DO 200 I = 1,NORD
        H1N = H1NM1 / RHO(I-1)
        WRITE (9,509) I, H1N
        H1NM1 = H1N
      200 CONTINUE
      DO 225 I = 1,NORD
        WRITE (9,509) I, RHO(I)
      225 CONTINUE

```

```

DO 250 I = 1,NORD
WRITE (9,509) I, BCAP(I)
250 CONTINUE
RRFX = RRF * RX
XXX = 1.DO/RRFX
ZINV = RRFX
WRITE (9,511) XXX
WRITE (9,509) NORD, ACAP
BESSJM1 = CDSIN (XXX) / XXX
DO 300 I = 1,NORD
ACAP = CONFRA ( I, ZINV )
RAT = ACAP + I*ZINV
BESSJ = BESSJM1 / RAT
WRITE (9,509) I, BESSJ
BESSJM1 = BESSJ
300 CONTINUE
DO 350 I = 1,NORD
ACAP = CONFRA ( I, ZINV )
WRITE (9,509) I, ACAP
350 CONTINUE
400 CONTINUE
CALL FUNCT(N,XX,F)
WRITE (9,511) XX
WRITE (9,511) F
WRITE (9,511) DE
WRITE (9,511) DM
500 CONTINUE
501 FORMAT('LOCATION OF RESONANCES FOR A SPHERICAL DROPLET',/
1 'Copyright 1988 by C. D. Cantrell'/
2 'All rights reserved'/
3 'Last revision 25 July 1988'/
4 'Input parameters:')
508 FORMAT('Numbers of iterations:')
509 FORMAT(I5,2(1PD30.15),I5)
510 FORMAT(8I10)
511 FORMAT(2(1PD30.15))
512 FORMAT('Resonances in ',A1,'(',I3,'):')
513 FORMAT(2(1PD30.15),I10)
514 FORMAT('ILLEGAL VALUE FOR REFRACTIVE INDEX')
515 FORMAT('Please enter the real part of the refractive index:')
516 FORMAT('Please enter the imaginary part of the refractive index:')
517 FORMAT('Please enter the lower limit of the partial-wave order:')
518 FORMAT('Please enter the upper limit of the partial-wave order:')
519 FORMAT('Please enter the number of resonances to be found:')
520 FORMAT('Please enter the maximum number of iterations/resonance:')
521 FORMAT('Please enter the starting value of the size parameter:')

```

```

522 FORMAT('Please enter the increment between successive starting ',
1 'values:')
523 FORMAT('Please enter the desired number of significant figures:')
524 FORMAT('Calculating...')
530 FORMAT('n.re =',F10.7,' n.im =',F10.7,/'lowest order =',I5,
1 ' highest order =',I5,' max. iterations =',I5/'no. of sig.'
2 ' figs. =',I5)
1000 STOP
END
C*****
SUBROUTINE FUNCT (N,XX,F)
REAL*8 RFR,RFI,DZERO,XX,ZABS,F
COMPLEX*16 ACAP,DE,DM,X,RF,RRF,RRFX,RX,ZERO,CONFRA,ZINV,
1 BCAP(-1:7000),RHO(-1:7000),CDON
DIMENSION XX(2)
COMMON/INT/NORD,IAB,ITER
COMMON/BESS/ACAP,BCAP,RHO,DE,DM,RX,ZINV
COMMON/INDEX/RFR,RFI,RF,RRF,RRFX
DATA ZERO/(0.DO,0.DO)/, DZERO/0.DO/
C*****CODE CABS FUNCTION INLINE FOR GREATER ACCURACY
ZABS(CDON)=(DMAX1(DABS(DBLE(CDON)),DABS(DIMAG(CDON))))*
1 DSQRT(((DMAX1(DABS(DBLE(CDON)),DABS(DIMAG(CDON))))/
2 (DMAX1(DABS(DBLE(CDON)),DABS(DIMAG(CDON)),1.ODO)))**2+
3 ((DMIN1(DABS(DBLE(CDON)),DABS(DIMAG(CDON))))/
4 (DMAX1(DABS(DBLE(CDON)),DABS(DIMAG(CDON)),1.ODO)))**2)
C*****SIZE PARAMETER (COMPLEX), ITS RECIPROCAL, AND 1/(RF*X)
X = DCMLX (XX(1),XX(2))
RX = 1.DO/X
RRFX = RRF*RX
C*****CALCULATION OF SPHERICAL BESSEL FUNCTIONS
C*****HANKEL FUNCTIONS OF FIRST KIND
ZINV = RX
CALL HANK
C*****BESSEL FUNCTION OF FIRST KIND
ZINV = RRFX
ACAP = CONFRA ( NORD , ZINV )
C*****EVALUATE MIE PARTIAL-WAVE RESONANT DENOMINATORS (DE,DM)
CALL DENOM
C*****CHOOSE A OR B ACCORDING TO VALUE OF IAB
IF (IAB.EQ.1) THEN
F = ZABS (DE)
ELSE IF (IAB.EQ.2) THEN
F = ZABS (DM)
ELSE
F = DZERO
END IF

```

```

ITER = ITER + 1
RETURN
END
C*****
SUBROUTINE HANK
C*****CALCULATION OF THE LOG. DERIVATIVE OF THE SPHERICAL HANKEL
C FUNCTIONS (RESULT IN ARRAY BCAP) AND THE RATIO OF SUCCESSIVE
C SPHERICAL HANKEL FUNCTIONS (RESULT IN ARRAY RHO). FORMULAS
C DERIVED BY C. D. CANTRELL FOLLOWING L. INFELD, QUARTERLY OF
C APPLIED MATH., VOL. 5, PP. 113-132 (1947).
C*****COPYRIGHT 1988 BY C. D. CANTRELL
C ALL RIGHTS RESERVED
C IMPLICIT REAL*8 (A-H,O-Z)
C COMPLEX*16 ACAP,DE,DM,RF,RRF,RRFX,RX,
C 1 BCAP(-1:7000),RHO(-1:7000),ZINV,CI
C COMMON/INT/NORD,IAB,ITER
C COMMON/BESS/ACAP,BCAP,RHO,DE,DM,RX,ZINV
C COMMON/INDEX/RFR,RFI,RF,RRF,RRFX
C DATA CI/(0.DO,1.DO)/
C RHO(-1) = CI
C DO 100 NN = 0,NORD
C RHO(NN) = 1.DO / ( ZINV * (2*NN + 1) - RHO(NN-1) )
100 CONTINUE
C BCAP(-1) = CI
C DO 200 NN = 0,NORD
C BCAP(NN) = -NN * ZINV + RHO(NN-1)
200 CONTINUE
RETURN
END
C*****
COMPLEX*16 FUNCTION CONFRA( N, ZINV )
C
C ROUTINE CODED BY W. J. WISCOMBE
C
C COMPUTE BESSEL FUNCTION RATIO CAPITAL-A-SUB-N FROM ITS
C CONTINUED FRACTION USING LENTZ METHOD ( REF. 1, PP. 17-20 )
C
C ZINV = RECIPROCAL OF ARGUMENT OF CAPITAL-A
C
C I N T E R N A L V A R I A B L E S
C -----
C
C CAK TERM IN CONTINUED FRACTION EXPANSION OF CAPITAL-A
C ( REF. 1, EQ. 25 )
C CAPT FACTOR USED IN LENTZ ITERATION FOR CAPITAL-A
C ( REF. 1, EQ. 27 )

```

```

C   CDENOM  DENOMINATOR IN -CAPT- ( REF. 1, EQ. 28B )
C   CNUMER  NUMERATOR   IN -CAPT- ( REF. 1, EQ. 28A )
C   CDTD    PRODUCT OF TWO SUCCESSIVE DENOMINATORS OF -CAPT-
C           FACTORS ( REF. 1, EQ. 34C )
C   CNTN    PRODUCT OF TWO SUCCESSIVE NUMERATORS OF -CAPT-
C           FACTORS ( REF. 1, EQ. 34B )
C   EPS1    ILL-CONDITIONING CRITERION
C   EPS2    CONVERGENCE CRITERION
C   KK      SUBSCRIPT K OF -CAK- ( REF. 1, EQ. 25B )
C   KOUNT   ITERATION COUNTER ( USED ONLY TO PREVENT RUNAWAY )
C   MAXIT   MAX. ALLOWED NO. OF ITERATIONS
C   MM      + 1 AND - 1, ALTERNATELY
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER      N
      COMPLEX*16 ZINV
      COMPLEX*16 CAK, CAPT, CDENOM, CDTD, CNUMER, CNTN, CDON
      DATA EPS1 / 1.D - 2 /, EPS2 / 4.4408920985006262D-16 /
      DATA MAXIT / 10000 /
C****DEFINE CABS INLINE FOR HIGHER ACCURACY
      ZABS(CDON)=(DMAX1(DABS(DREAL(CDON)),DABS(DIMAG(CDON))))*
1 DSQRT(((DMAX1(DABS(DREAL(CDON)),DABS(DIMAG(CDON))))/
2 (DMAX1(DABS(DREAL(CDON)),DABS(DIMAG(CDON)),1.0D0)))**2+
3 ((DMIN1(DABS(DREAL(CDON)),DABS(DIMAG(CDON))))/
4 (DMAX1(DABS(DREAL(CDON)),DABS(DIMAG(CDON)),1.0D0)))**2)
C
C                                     *** REF. 1, EQS. 25A, 27
      CONFRA = ( N + 1 ) * ZINV
      MM      = - 1
      KK      = 2 * N + 3
      CAK     = ( MM * KK ) * ZINV
      CDENOM = CAK
      CNUMER = CDENOM + 1.0 / CONFRA
      KOUNT  = 1
C
20 KOUNT = KOUNT + 1
C   IF ( KOUNT.GT.MAXIT )
C   $   CALL ERRMSG( 'CONFRA--ITERATION FAILED TO CONVERGE$', .TRUE.)
C
C                                     *** REF. 2, EQ. 25B
      MM = - MM
      KK = KK + 2
      CAK = ( MM * KK ) * ZINV
C
C                                     *** REF. 2, EQ. 32
      IF (          ZABS( CNUMER/CAK ).LE.EPS1
$   .OR. ZABS( CDENOM/CAK ).LE.EPS1 ) THEN

```



```

C      DERIVATIVE OF THE SPHERICAL BESSEL FUNCTION OF THE FIRST KIND
C      (ACAP) CALCULATED BY THE CONTINUED-FRACTION METHOD OF W. J.
C      LENTZ; AND THE LOGARITHMIC DERIVATIVE OF THE SPHERICAL HANKEL
C      FUNCTION OF THE FIRST KIND (BCAP), CALCULATED BY UPWARD
C      RECURRENCE
C*****COPYRIGHT 1988 BY C. D. CANTRELL
C      ALL RIGHTS RESERVED
C      REAL*8 RFR,RFI
C      COMPLEX*16 ACAP,BCAP(-1:7000),RHO(-1:7000),DE,DM,RF,RRF,RX,
C      1 RRFX,ZINV
C      COMMON/INT/NORD,IAB,ITER
C      COMMON/BESS/ACAP,BCAP,RHO,DE,DM,RX,ZINV
C      COMMON/INDEX/RFR,RFI,RF,RRF,RRFX
C*****ELECTRIC HERTZ VECTOR PARTIAL-WAVE DENOMINATOR
C      DE = ACAP - RF*BCAP(NORD)
C*****MAGNETIC HERTZ VECTOR PARTIAL-WAVE DENOMINATOR
C      DM = RF*ACAP - BCAP(NORD)
C      RETURN
C      END
C*****
C      END OF PROGRAM FINDRES3
C*****
C      .DE
C      .bp
C      .LP
C      .DS L
C*****
C      BEGINNING OF IMSL ROUTINES
C*****
C      IMSL ROUTINE NAME   - ZXMIN
C
C-----
C
C      COMPUTER           - CONVEX
C
C      LATEST REVISION    - JUNE 1, 1981
C
C      PURPOSE            - MINIMUM OF A FUNCTION OF N VARIABLES USING
C                          A QUASI-NEWTON METHOD
C
C      USAGE              - CALL ZXMIN (FUNCT,N,NSIG,MAXFN,IOPT,X,H,G,F,
C                          W,IER)
C
C      ARGUMENTS          FUNCT - A USER SUPPLIED SUBROUTINE WHICH CALCULATES
C                          THE FUNCTION F FOR GIVEN PARAMETER VALUES
C                          X(1),X(2),...,X(N).

```

C THE CALLING SEQUENCE HAS THE FOLLOWING FORM  
 C CALL FUNCT(N,X,F)  
 C WHERE X IS A VECTOR OF LENGTH N.  
 C FUNCT MUST APPEAR IN AN EXTERNAL STATEMENT  
 C IN THE CALLING PROGRAM. FUNCT MUST NOT  
 C ALTER THE VALUES OF X(I), I=1,...,N OR N.  
 C N - THE NUMBER OF PARAMETERS (I.E., THE LENGTH  
 C OF X) (INPUT)  
 C NSIG - CONVERGENCE CRITERION. (INPUT). THE NUMBER  
 C OF DIGITS OF ACCURACY REQUIRED IN THE  
 C PARAMETER ESTIMATES.  
 C THIS CONVERGENCE CONDITION IS SATISFIED IF  
 C ON TWO SUCCESSIVE ITERATIONS, THE PARAMETER  
 C ESTIMATES (I.E., X(I), I=1,...,N) AGREE,  
 C COMPONENT BY COMPONENT, TO NSIG DIGITS.  
 C MAXFN - MAXIMUM NUMBER OF FUNCTION EVALUATIONS (I.E.,  
 C CALLS TO SUBROUTINE FUNCT) ALLOWED. (INPUT)  
 C IOPT - OPTIONS SELECTOR. (INPUT)  
 C IOPT = 0 CAUSES ZXMIN TO INITIALIZE THE  
 C HESSIAN MATRIX H TO THE IDENTITY MATRIX.  
 C IOPT = 1 INDICATES THAT H HAS BEEN INITIALIZED  
 C BY THE USER TO A POSITIVE DEFINITE MATRIX.  
 C IOPT = 2 CAUSES ZXMIN TO COMPUTE THE DIAGONAL  
 C VALUES OF THE HESSIAN MATRIX AND SET H TO  
 C A DIAGONAL MATRIX CONTAINING THESE VALUES.  
 C IOPT = 3 CAUSES ZXMIN TO COMPUTE AN ESTIMATE  
 C OF THE HESSIAN IN H.  
 C X - VECTOR OF LENGTH N CONTAINING PARAMETER  
 C VALUES.  
 C ON INPUT, X MUST CONTAIN THE INITIAL  
 C PARAMETER ESTIMATES.  
 C ON OUTPUT, X CONTAINS THE FINAL PARAMETER  
 C ESTIMATES AS DETERMINED BY ZXMIN.  
 C H - VECTOR OF LENGTH  $N*(N+1)/2$  CONTAINING AN  
 C ESTIMATE OF THE HESSIAN MATRIX  
 C  $D**2F/(DX(I)DX(J))$ , I,J=1,...,N.  
 C H IS STORED IN SYMMETRIC STORAGE MODE.  
 C ON INPUT, IF IOPT = 0, 2, OR 3 ZXMIN INITIA-  
 C LIZES H. AN INITIAL SETTING OF H BY THE  
 C USER IS INDICATED BY IOPT=1.  
 C H MUST BE POSITIVE DEFINITE. IF IT IS NOT,  
 C A TERMINAL ERROR OCCURS.  
 C ON OUTPUT, H CONTAINS AN ESTIMATE OF THE  
 C HESSIAN AT THE FINAL PARAMETER ESTIMATES  
 C (I.E., AT X(1),X(2),...,X(N))  
 C G - A VECTOR OF LENGTH N CONTAINING AN ESTIMATE

```

C           OF THE GRADIENT DF/DX(I),I=1,...,N AT THE
C           FINAL PARAMETER ESTIMATES. (OUTPUT)
C           F   - A SCALAR CONTAINING THE VALUE OF THE FUNCTION
C                 AT THE FINAL PARAMETER ESTIMATES. (OUTPUT)
C           W   - A VECTOR OF LENGTH 3*N USED AS WORKING SPACE.
C                 ON OUTPUT, WORK(I), CONTAINS FOR
C                 I = 1, THE NORM OF THE GRADIENT (I.E.,
C                   Sqrt(G(1)**2+G(2)**2+...+G(N)**2))
C                 I = 2, THE NUMBER OF FUNCTION EVALUATIONS
C                   PERFORMED.
C                 I = 3, AN ESTIMATE OF THE NUMBER OF
C                   SIGNIFICANT DIGITS IN THE FINAL
C                   PARAMETER ESTIMATES.
C           IER  - ERROR PARAMETER (OUTPUT)
C                 TERMINAL ERROR
C                 IER = 129 IMPLIES THAT THE INITIAL HESSIAN
C                   USED BY ZXMIN IS NOT POSITIVE DEFINITE,
C                   EVEN AFTER ADDING A MULTIPLE OF THE
C                   IDENTITY TO MAKE ALL DIAGONAL ELEMENTS
C                   POSITIVE.
C                 IER = 130 IMPLIES THAT THE ITERATION WAS
C                   TERMINATED DUE TO ROUNDING ERRORS
C                   BECOMING DOMINANT. THE PARAMETER
C                   ESTIMATES HAVE NOT BEEN DETERMINED TO
C                   NSIG DIGITS.
C                 IER = 131 IMPLIES THAT THE ITERATION WAS
C                   TERMINATED BECAUSE MAXFN WAS EXCEEDED.
C
C   PRECISION/HARDWARE - REAL*8
C                       - CONVEX
C
C   REQD. IMSL ROUTINES - UERTST,UGETIO,ZXMJN
C
C   NOTATION           - INFORMATION ON SPECIAL NOTATION AND
C                       CONVENTIONS IS AVAILABLE IN THE MANUAL
C                       INTRODUCTION OR THROUGH IMSL ROUTINE UHELP
C
C   COPYRIGHT          - 1978 BY IMSL, INC. ALL RIGHTS RESERVED.
C
C   WARRANTY           - IMSL WARRANTS ONLY THAT IMSL TESTING HAS BEEN
C                       APPLIED TO THIS CODE. NO OTHER WARRANTY,
C                       EXPRESSED OR IMPLIED, IS APPLICABLE.
C
C-----
C
C   SUBROUTINE ZXMIN (FUNCT,N,NSIG,MAXFN,IOPT,X,H,G,F,W,IER)

```

```

C                                     SPECIFICATIONS FOR ARGUMENTS
INTEGER                               N, NSIG, MAXFN, IOPT, IER
DOUBLE PRECISION                       X(N), G(N), H(1), F, W(1)
C                                     SPECIFICATIONS FOR LOCAL VARIABLES
INTEGER                               IG, IGG, IS, IDIFF, IR, IJ, I, J, NM1, JJ, JP1, L, KJ, K,
* IFN, LINK, ITN, II, IM1, JNT, NP1, JB, NJ
DOUBLE PRECISION                       REPS, AX, ZERO, ONE, HALF, SEVEN, FIVE, TWELVE, TEN, HH,
* EPS, HJJ, V, DF, RELX, GSO, DIFF, AEPS, ALPHA, FF, TOT,
* F1, F2, Z, GYS, DGS, SIG, ZZ, GNRM, P1, HHH, GHH, H2, F11,
* F12, F21, F22, HMAX, HMIN
DATA                                   REPS/.2220446050D-15/, AX/0.1D0/
DATA                                   ZERO/0.0D0/, ONE/1.0D0/, HALF/0.5D0/,
* SEVEN/7.0D0/, FIVE/5.0D0/, TWELVE/12.0D0/,
* TEN/10.0D0/, P1/0.1D0/
C                                     INITIALIZATION
C                                     FIRST EXECUTABLE STATEMENT
IER = 0
HH = DSQRT(REPS)
H2 = DSQRT(HH)
EPS = TEN**(-NSIG)
IG = N
IGG = N+N
IS = IGG
IDIFF = 1
IR = N
W(1) = -ONE
W(2) = ZERO
W(3) = ZERO
C                                     EVALUATE FUNCTION AT STARTING POINT
DO 5 I=1, N
    G(I) = X(I)
5 CONTINUE
CALL FUNCT(N, G, F)
IFN = 1
IF (IOPT.EQ.1) GO TO 50
C                                     SET OFF-DIAGONAL ELEMENTS OF H TO 0.0
IF (N.EQ.1) GO TO 20
IJ = 2
DO 15 I=2, N
    DO 10 J=2, I
        H(IJ) = ZERO
        IJ = IJ+1
10 CONTINUE
    IJ = IJ+1
15 CONTINUE
20 IF (IOPT.NE.0) GO TO 30

```

```

C                               SET DIAGONAL ELEMENTS OF H TO ONE
    IJ = 0
    DO 25 I=1,N
        IJ = IJ+I
        H(IJ) = ONE
25  CONTINUE
    GO TO 95

C                               GET DIAGONAL ELEMENTS OF HESSIAN
30  IM1 = 1
    NM1 = 1
    NP1 = N+1
    DO 35 I=2,NP1
        HHH = H2*DMAX1(DABS(X(IM1)),AX)
        G(IM1) = X(IM1)+HHH
        CALL FUNCT(N,G,F2)
        G(IM1) = X(IM1)-HHH
        CALL FUNCT(N,G,FF)
        H(NM1) = (FF-F+F2-F)/(HHH*HHH)
        G(IM1) = X(IM1)
        IM1 = I
        NM1 = I+NM1
35  CONTINUE
    IFN = IFN+N+N
    IF (IOPT.NE.3 .OR. N.EQ.1) GO TO 50

C                               GET THE REST OF THE HESSIAN
    JJ = 1
    II = 2
    DO 45 I=2,N
        GHH = H2*DMAX1(DABS(X(I)),AX)
        DO 40 J=1,JJ
            HHH = H2*DMAX1(DABS(X(J)),AX)
            G(I) = X(I)+GHH
            G(J) = X(J)+HHH
            CALL FUNCT(N,G,F22)
            G(I) = X(I)-GHH
            CALL FUNCT(N,G,F12)
            G(J) = X(J)-HHH
            CALL FUNCT(N,G,F11)
            G(I) = X(I)+GHH
            CALL FUNCT(N,G,F21)
            H(II) = (F22-F21-F12+F11)/(4.DO*HHH*GHH)
            G(J) = X(J)
            II = II+1
40  CONTINUE
    G(I) = X(I)
    JJ = JJ+1

```



```

          KJ = KJ+K
75      CONTINUE
          H(IJ) = V
80      CONTINUE
85      CONTINUE
90      IF (IR.EQ.N) GO TO 95
          IER = 129
          GO TO 9000
95      ITN = 0
          DF = -ONE
C
C          EVALUATE GRADIENT W(IG+I), I=1,...,N
100     LINK = 1
          GO TO 280
105     CONTINUE
C
C          BEGIN ITERATION LOOP
          IF (IFN.GE.MAXFN) GO TO 240
          ITN = ITN+1
          DO 110 I=1,N
              W(I) = -W(IG+I)
110     CONTINUE
C
C          DETERMINE SEARCH DIRECTION W
C          BY SOLVING H*W = -G WHERE
C          H = L*D*L-TRANSPOSE
          IF (IR.LT.N) GO TO 140
C
C          N .EQ. 1
          G(1) = W(1)
          IF (N.GT.1) GO TO 115
          W(1) = W(1)/H(1)
          GO TO 140
C
C          N .GT. 1
115     II = 1
C
C          SOLVE L*W = -G
          DO 125 I=2,N
              IJ = II
              II = II+I
              V = W(I)
              IM1 = I-1
              DO 120 J=1,IM1
                  IJ = IJ+1
                  V = V-H(IJ)*W(J)
120     CONTINUE
              G(I) = V
              W(I) = V
125     CONTINUE
C
C          SOLVE (D*LT)*Z = W WHERE
C          LT = L-TRANSPOSE

```

```

W(N) = W(N)/H(II)
JJ = II
NM1 = N-1
DO 135 NJ=1,NM1
C
J = N-1,N-2,...,1
J = N-NJ
JP1 = J+1
JJ = JJ-JP1
V = W(J)/H(JJ)
IJ = JJ
DO 130 I=JP1,N
IJ = IJ+I-1
V = V-H(IJ)*W(I)
130 CONTINUE
W(J) = V
135 CONTINUE
C
DETERMINE STEP LENGTH ALPHA
140 RELX = ZERO
GSO = ZERO
DO 145 I=1,N
W(IS+I) = W(I)
DIFF = DABS(W(I))/DMAX1(DABS(X(I)),AX)
RELX = DMAX1(RELX,DIFF)
GSO = GSO+W(IG+I)*W(I)
145 CONTINUE
IF (RELX.EQ.ZERO) GO TO 245
AEPS = EPS/RELX
IER = 130
IF (GSO.GE.ZERO) GO TO 245
IF (DF.EQ.ZERO) GO TO 245
IER = 0
ALPHA = (-DF-DF)/GSO
IF (ALPHA.LE.ZERO) ALPHA = ONE
ALPHA = DMIN1(ALPHA,ONE)
IF (IDIFF.EQ.2) ALPHA = DMAX1(P1,ALPHA)
FF = F
TOT = ZERO
JNT = 0
C
SEARCH ALONG X+ALPHA*W
150 IF (IFN.GE.MAXFN) GO TO 240
DO 155 I=1,N
W(I) = X(I)+ALPHA*W(IS+I)
155 CONTINUE
CALL FUNCT(N,W,F1)
IFN = IFN+1
IF (F1.GE.F) GO TO 180

```

```

      F2 = F
      TOT = TOT+ALPHA
160  IER = 0
      F = F1
      DO 165 I=1,N
          X(I) = W(I)
165  CONTINUE
      IF (JNT-1) 170, 200, 205
170  IF (IFN.GE.MAXFN) GO TO 240
      DO 175 I=1,N
          W(I) = X(I)+ALPHA*W(IS+I)
175  CONTINUE
      CALL FUNCT(N,W,F1)
      IFN = IFN+1
      IF (F1.GE.F) GO TO 205
      IF (F1+F2.GE.F+F .AND. SEVEN*F1+FIVE*F2.GT.TWELVE*F) JNT = 2
      TOT = TOT+ALPHA
      ALPHA = ALPHA+ALPHA
      GO TO 160
180  CONTINUE
      IF (F.EQ.FF .AND. IDIFF.EQ.2 .AND. RELX.GT.EPS) IER = 130
      IF (ALPHA.LT.AEPS) GO TO 245
      IF (IFN.GE.MAXFN) GO TO 240
      ALPHA = HALF*ALPHA
      DO 185 I=1,N
          W(I) = X(I)+ALPHA*W(IS+I)
185  CONTINUE
      CALL FUNCT(N,W,F2)
      IFN = IFN+1
      IF (F2.GE.F) GO TO 195
      TOT = TOT+ALPHA
      IER = 0
      F = F2
      DO 190 I=1,N
          X(I) = W(I)
190  CONTINUE
      GO TO 200
195  Z = P1
      IF (F1+F.GT.F2+F2) Z = ONE+HALF*(F-F1)/(F+F1-F2-F2)
      Z = DMAX1(P1,Z)
      ALPHA = Z*ALPHA
      JNT = 1
      GO TO 150
200  IF (TOT.LT.AEPS) GO TO 245
205  ALPHA = TOT
C

```

SAVE OLD GRADIENT

```

        DO 210 I=1,N
            W(I) = W(IG+I)
210 CONTINUE
C                                     EVALUATE GRADIENT W(IG+I), I=1,...,N
        LINK = 2
        GO TO 280
215 IF (IFN.GE.MAXFN) GO TO 240
        GYS = ZERO
        DO 220 I=1,N
            GYS = GYS+W(IG+I)*W(IS+I)
            W(IGG+I) = W(I)
220 CONTINUE
        DF = FF-F
        DGS = GYS-GSO
        IF (DGS.LE.ZERO) GO TO 105
        IF (DGS+ALPHA*GSO.GT.ZERO) GO TO 230
C                                     UPDATE HESSIAN H USING
C                                     COMPLEMENTARY DFP FORMULA
        SIG = ONE/GSO
        IR = -IR
        CALL ZXMJN(H,N,W,SIG,G,IR,0,ZERO)
        DO 225 I=1,N
            G(I) = W(IG+I)-W(IGG+I)
225 CONTINUE
        SIG = ONE/(ALPHA*DGS)
        IR = -IR
        CALL ZXMJN(H,N,G,SIG,W,IR,0,ZERO)
        GO TO 105
C                                     UPDATE HESSIAN USING
C                                     DFP FORMULA
230 ZZ = ALPHA/(DGS-ALPHA*GSO)
        SIG = -ZZ
        CALL ZXMJN(H,N,W,SIG,G,IR,0,REPS)
        Z = DGS*ZZ-ONE
        DO 235 I=1,N
            G(I) = W(IG+I)+Z*W(IGG+I)
235 CONTINUE
        SIG = ONE/(ZZ*DGS*DGS)
        CALL ZXMJN(H,N,G,SIG,W,IR,0,ZERO)
        GO TO 105
240 IER = 131
C                                     MAXFN FUNCTION EVALUATIONS
        GO TO 250
245 IF (IDIFF.EQ.2) GO TO 250
C                                     CHANGE TO CENTRAL DIFFERENCES
        IDIFF = 2

```

```

        GO TO 100
250 IF (IER.NE.0) GO TO 255
    IF (RELX.LE.EPS) GO TO 255
        GO TO 100
C
                                MOVE GRADIENT TO G AND RETURN
255 GNRM = ZERO
    DO 260 I=1,N
        G(I) = W(IG+I)
        GNRM = GNRM+G(I)*G(I)
260 CONTINUE
    GNRM = DSQRT(GNRM)
    W(1) = GNRM
    W(2) = IFN
    W(3) = -DLOG10(DMAX1(REPS,RELX))
C
                                COMPUTE H = L*D*L-TRANSPPOSE
    IF (N.EQ.1) GO TO 9000
    NP1 = N+1
    NM1 = N-1
    JJ = (N*(NP1))/2
    DO 275 JB=1,NM1
        JP1 = NP1-JB
        JJ = JJ-JP1
        HJJ = H(JJ)
        IJ = JJ
        L = 0
        DO 270 I=JP1,N
            L = L+1
            IJ = IJ+I-1
            V = H(IJ)*HJJ
            KJ = IJ
            DO 265 K=I,N
                H(KJ+L) = H(KJ+L)+H(KJ)*V
                KJ = KJ+K
265     CONTINUE
            H(IJ) = V
270     CONTINUE
        HJJ = H(JJ)
275 CONTINUE
        GO TO 9000
C
                                EVALUATE GRADIENT
280 IF (IDIFF.EQ.2) GO TO 290
C
                                FORWARD DIFFERENCES
C
                                GRADIENT = W(IG+I), I=1,...,N
DO 285 I=1,N
    Z = HH*DMAX1(DABS(X(I)),AX)
    ZZ = X(I)

```

```

        X(I) = ZZ+Z
        CALL FUNCT(N,X,F1)
        W(IG+I) = (F1-F)/Z
        X(I) = ZZ
285 CONTINUE
        IFN = IFN+N
        GO TO (105, 215), LINK
C
C                                     CENTRAL DIFFERENCES
C                                     GRADIENT = W(IG+I), I=1,...,N
290 DO 295 I=1,N
        Z = HH*DMAX1(DABS(X(I)),AX)
        ZZ = X(I)
        X(I) = ZZ+Z
        CALL FUNCT(N,X,F1)
        X(I) = ZZ-Z
        CALL FUNCT(N,X,F2)
        W(IG+I) = (F1-F2)/(Z+Z)
        X(I) = ZZ
295 CONTINUE
        IFN = IFN+N+N
        GO TO (105, 215), LINK
9000 CONTINUE
        IF (IER.NE.0) CALL UERTST(IER,'ZXMIN ')
9005 RETURN
        END
C   IMSL ROUTINE NAME   - ZXMJN
C
C-----
C
C   COMPUTER           - CONVEX
C
C   LATEST REVISION    - NOVEMBER 1, 1984
C
C   PURPOSE            - NUCLEUS CALLED BY IMSL ROUTINES ZXMIN AND
C                       ZXMWD
C
C   PRECISION/HARDWARE - REAL*8
C                       - CONVEX
C
C   REQD. IMSL ROUTINES - NONE REQUIRED
C
C   NOTATION           - INFORMATION ON SPECIAL NOTATION AND
C                       CONVENTIONS IS AVAILABLE IN THE MANUAL
C                       INTRODUCTION OR THROUGH IMSL ROUTINE UHELP
C
C   COPYRIGHT          - 1978 BY IMSL, INC. ALL RIGHTS RESERVED.

```

```

C
C   WARRANTY          - IMSL WARRANTS ONLY THAT IMSL TESTING HAS BEEN
C                     APPLIED TO THIS CODE. NO OTHER WARRANTY,
C                     EXPRESSED OR IMPLIED, IS APPLICABLE.
C
C-----
C
C   SUBROUTINE ZXMJN(A,N,Z,SIG,W,IR,MK,EPS)
C                     SPECIFICATIONS FOR ARGUMENTS
C   INTEGER           N,IR,MK
C   DOUBLE PRECISION A(1),Z(N),SIG,W(N),EPS
C                     SPECIFICATIONS FOR LOCAL VARIABLES
C   INTEGER           J,JJ,IJ,JP1,I,II,MM
C   DOUBLE PRECISION ZERO,ONE,FOUR,TI,V,TIM,AL,R,B,GM,Y,RINF,SQRINF
C   DATA             ZERO/0.0D0/,ONE/1.0D0/,FOUR/4.0D0/
C   DATA             RINF/.4494232D+308/
C
C                     UPDATE FACTORS GIVEN IN A
C                     SIG*Z*Z-TRANSPOSE IS ADDED
C                     FIRST EXECUTABLE STATEMENT
C
C   SQRINF = DSQRT(RINF)
C   IF (N.GT.1) GO TO 5
C
C
C                     N .EQ. 1
C   A(1) = A(1) + SIG*Z(1)*Z(1)
C   IR = 1
C   IF (A(1).GT.ZERO) GO TO 9005
C   A(1) = ZERO
C   IR = 0
C   GO TO 9005
C
C                     N .GT. 1
C   5 IF (SIG.GT.ZERO) GO TO 65
C   IF (SIG.EQ.ZERO .OR. IR.EQ.0) GO TO 9005
C   TI = ONE/SIG
C   JJ = 0
C   IF (MK.EQ.0) GO TO 15
C
C                     L*W = Z ON INPUT
C
C   DO 10 J = 1, N
C     JJ = JJ + J
C     IF (A(JJ).NE.ZERO) TI = TI + (W(J)*W(J))/A(JJ)
C 10 CONTINUE
C   GO TO 40
C
C                     SOLVE L*W = Z
C
C 15 DO 20 J = 1, N
C     W(J) = Z(J)
C 20 CONTINUE
C   DO 35 J = 1, N
C     JJ = JJ + J

```

```

V = W(J)
IF (A(JJ).GT.ZERO) GO TO 25
W(J) = ZERO
GO TO 35
25  TI = TI + (V*V)/A(JJ)
    IF (J.EQ.N) GO TO 35
    IJ = JJ
    JP1 = J + 1
    DO 30 I = JP1, N
        IJ = IJ + I - 1
        W(I) = W(I) - V*A(IJ)
30  CONTINUE
35  CONTINUE
C                                     SET TI, TIM AND W
40  IF (IR.LE.0) GO TO 45
    IF (TI.GT.ZERO) GO TO 50
    IF (MK-1) 65, 65, 55
45  TI = ZERO
    IR = -IR - 1
    GO TO 55
50  TI = EPS/SIG
    IF (EPS.EQ.ZERO) IR = IR - 1
55  TIM = TI
    II = JJ
    I = N
    DO 60 J = 1, N
        IF (A(II).NE.ZERO) TIM = TI - (W(I)*W(I))/A(II)
        W(I) = TI
        TI = TIM
        II = II - I
        I = I - 1
60  CONTINUE
    MM = 1
    GO TO 70
65  MM = 0
    TIM = ONE/SIG
70  JJ = 0
C                                     UPDATE A
    DO 120 J = 1, N
        JJ = JJ + J
        IJ = JJ
        JP1 = J + 1
C                                     UPDATE A(J,J)
    V = Z(J)
    IF (A(JJ).GT.ZERO) GO TO 95
C                                     A(J,J) .EQ. ZERO

```

```

      IF (IR.GT.0 .OR. SIG.LT.ZERO .OR. V.EQ.ZERO) GO TO 90
      IR = 1 - IR
      IF (V.GE.SQRINF) GO TO 75
      A(JJ) = (V*V)/TIM
      GO TO 80
75    A(JJ) = RINF/TIM
80    IF (J.EQ.N) GO TO 9005
      DO 85 I = JP1, N
          IJ = IJ + I - 1
          A(IJ) = Z(I)/V
85    CONTINUE
      GO TO 9005
90    TI = TIM
      GO TO 120
C
      A(J,J) .GT. ZERO
95    AL = V/A(JJ)
      TI = W(J)
      IF (MM.EQ.0) TI = TIM + V*AL
      R = TI/TIM
      A(JJ) = R*A(JJ)
      IF (R.EQ.ZERO) GO TO 125
      IF (J.EQ.N) GO TO 125
C
      UPDATE REMAINDER OF COLUMN J
      B = AL/TI
      IF (R.GT.FOUR) GO TO 105
      DO 100 I = JP1, N
          IJ = IJ + I - 1
          Z(I) = Z(I) - V*A(IJ)
          A(IJ) = A(IJ) + B*Z(I)
100   CONTINUE
      GO TO 115
105   GM = TIM/TI
      DO 110 I = JP1, N
          IJ = IJ + I - 1
          Y = A(IJ)
          A(IJ) = B*Z(I) + Y*GM
          Z(I) = Z(I) - V*Y
110   CONTINUE
115   TIM = TI
120   CONTINUE
125   IF (IR.LT.0) IR = -IR
9005  CONTINUE
      RETURN
      END
C    IMSL ROUTINE NAME    - UERTST
C

```

```

C-----
C
C  COMPUTER          - CONVEX
C
C  LATEST REVISION  - JUNE 1, 1982
C
C  PURPOSE          - PRINT A MESSAGE REFLECTING AN ERROR CONDITION
C
C  USAGE           - CALL UERTST (IER,NAME)
C
C  ARGUMENTS      IER - ERROR PARAMETER. (INPUT)
C                   IER = I+J WHERE
C                   I = 128 IMPLIES TERMINAL ERROR MESSAGE,
C                   I = 64 IMPLIES WARNING WITH FIX MESSAGE,
C                   I = 32 IMPLIES WARNING MESSAGE.
C                   J = ERROR CODE RELEVANT TO CALLING
C                   ROUTINE.
C                   NAME - A CHARACTER STRING OF LENGTH SIX PROVIDING
C                   THE NAME OF THE CALLING ROUTINE. (INPUT)
C
C  PRECISION/HARDWARE - SINGLE/ALL
C
C  REQD. IMSL ROUTINES - UGETIO,USPKD
C
C  NOTATION        - INFORMATION ON SPECIAL NOTATION AND
C                   CONVENTIONS IS AVAILABLE IN THE MANUAL
C                   INTRODUCTION OR THROUGH IMSL ROUTINE UHELP
C
C  REMARKS         THE ERROR MESSAGE PRODUCED BY UERTST IS WRITTEN
C                   TO THE STANDARD OUTPUT UNIT. THE OUTPUT UNIT
C                   NUMBER CAN BE DETERMINED BY CALLING UGETIO AS
C                   FOLLOWS.. CALL UGETIO(1,NIN,NOUT).
C                   THE OUTPUT UNIT NUMBER CAN BE CHANGED BY CALLING
C                   UGETIO AS FOLLOWS..
C                   NIN = 0
C                   NOUT = NEW OUTPUT UNIT NUMBER
C                   CALL UGETIO(3,NIN,NOUT)
C                   SEE THE UGETIO DOCUMENT FOR MORE DETAILS.
C
C  COPYRIGHT       - 1982 BY IMSL, INC. ALL RIGHTS RESERVED.
C
C  WARRANTY        - IMSL WARRANTS ONLY THAT IMSL TESTING HAS BEEN
C                   APPLIED TO THIS CODE. NO OTHER WARRANTY,
C                   EXPRESSED OR IMPLIED, IS APPLICABLE.
C-----

```

```

C
SUBROUTINE UERTST (IER,NAME)
C
C                                SPECIFICATIONS FOR ARGUMENTS
INTEGER          IER
CHARACTER        NAME*(*)
C
C                                SPECIFICATIONS FOR LOCAL VARIABLES
INTEGER          I,IEQDF,IOUNIT,LEVEL,LEVOLD,NIN,NMTB
CHARACTER        IEQ,NAMEQ(6),NAMSET(6),NAMUPK(6)
DATA             NAMSET/'U','E','R','S','E','T'/
DATA             NAMEQ/6*' '/
DATA             LEVEL/4/,IEQDF/0/,IEQ/'='/'
C
C                                UNPACK NAME INTO NAMUPK
C                                FIRST EXECUTABLE STATEMENT
CALL USPDK (NAME,6,NAMUPK,NMTB)
C
C                                GET OUTPUT UNIT NUMBER
CALL UGETIO(1,NIN,IOUNIT)
C
C                                CHECK IER
IF (IER.GT.999) GO TO 25
IF (IER.LT.-32) GO TO 55
IF (IER.LE.128) GO TO 5
IF (LEVEL.LT.1) GO TO 30
C
C                                PRINT TERMINAL MESSAGE
IF (IEQDF.EQ.1) WRITE(IOUNIT,35) IER,NAMEQ,IEQ,NAMUPK
IF (IEQDF.EQ.0) WRITE(IOUNIT,35) IER,NAMUPK
GO TO 30
5 IF (IER.LE.64) GO TO 10
IF (LEVEL.LT.2) GO TO 30
C
C                                PRINT WARNING WITH FIX MESSAGE
IF (IEQDF.EQ.1) WRITE(IOUNIT,40) IER,NAMEQ,IEQ,NAMUPK
IF (IEQDF.EQ.0) WRITE(IOUNIT,40) IER,NAMUPK
GO TO 30
10 IF (IER.LE.32) GO TO 15
C
C                                PRINT WARNING MESSAGE
IF (LEVEL.LT.3) GO TO 30
IF (IEQDF.EQ.1) WRITE(IOUNIT,45) IER,NAMEQ,IEQ,NAMUPK
IF (IEQDF.EQ.0) WRITE(IOUNIT,45) IER,NAMUPK
GO TO 30
15 CONTINUE
C
C                                CHECK FOR UERSET CALL
DO 20 I=1,6
    IF (NAMUPK(I).NE.NAMSET(I)) GO TO 25
20 CONTINUE
LEVOLD = LEVEL
LEVEL = IER
IER = LEVOLD
IF (LEVEL.LT.0) LEVEL = 4

```

```

        IF (LEVEL.GT.4) LEVEL = 4
        GO TO 30
25 CONTINUE
        IF (LEVEL.LT.4) GO TO 30
C
                                PRINT NON-DEFINED MESSAGE
        IF (IEQDF.EQ.1) WRITE(IOUNIT,50) IER,NAMEQ,IEQ,NAMUPK
        IF (IEQDF.EQ.0) WRITE(IOUNIT,50) IER,NAMUPK
30 IEQDF = 0
        RETURN
35 FORMAT(19H *** TERMINAL ERROR,10X,7H( IER = ,I3,
1         20H) FROM IMSL ROUTINE ,6A1,A1,6A1)
40 FORMAT(27H *** WARNING WITH FIX ERROR,2X,7H( IER = ,I3,
1         20H) FROM IMSL ROUTINE ,6A1,A1,6A1)
45 FORMAT(18H *** WARNING ERROR,11X,7H( IER = ,I3,
1         20H) FROM IMSL ROUTINE ,6A1,A1,6A1)
50 FORMAT(20H *** UNDEFINED ERROR,9X,7H( IER = ,I5,
1         20H) FROM IMSL ROUTINE ,6A1,A1,6A1)
C
C
                                SAVE P FOR P = R CASE
C
                                P IS THE PAGE NAMUPK
C
                                R IS THE ROUTINE NAMUPK
55 IEQDF = 1
        DO 60 I=1,6
60 NAMEQ(I) = NAMUPK(I)
65 RETURN
        END
C  IMSL ROUTINE NAME   - UGETIO
C
C-----
C
C  COMPUTER           - CONVEX
C
C  LATEST REVISION    - JUNE 1, 1981
C
C  PURPOSE            - TO RETRIEVE CURRENT VALUES AND TO SET NEW
C                      VALUES FOR INPUT AND OUTPUT UNIT
C                      IDENTIFIERS.
C
C  USAGE              - CALL UGETIO(IOPT,NIN,NOUT)
C
C  ARGUMENTS          IOPT - OPTION PARAMETER. (INPUT)
C                      IF IOPT=1, THE CURRENT INPUT AND OUTPUT
C                      UNIT IDENTIFIER VALUES ARE RETURNED IN NIN
C                      AND NOUT, RESPECTIVELY.
C                      IF IOPT=2, THE INTERNAL VALUE OF NIN IS
C                      RESET FOR SUBSEQUENT USE.

```

```

C          IF IOPT=3, THE INTERNAL VALUE OF NOUT IS
C          RESET FOR SUBSEQUENT USE.
C          NIN   - INPUT UNIT IDENTIFIER.
C                  OUTPUT IF IOPT=1, INPUT IF IOPT=2.
C          NOUT  - OUTPUT UNIT IDENTIFIER.
C                  OUTPUT IF IOPT=1, INPUT IF IOPT=3.
C
C PRECISION/HARDWARE - SINGLE/ALL
C
C REQD. IMSL ROUTINES - NONE REQUIRED
C
C NOTATION          - INFORMATION ON SPECIAL NOTATION AND
C                  CONVENTIONS IS AVAILABLE IN THE MANUAL
C                  INTRODUCTION OR THROUGH IMSL ROUTINE UHELP
C
C REMARKS          EACH IMSL ROUTINE THAT PERFORMS INPUT AND/OR OUTPUT
C                  OPERATIONS CALLS UGETIO TO OBTAIN THE CURRENT UNIT
C                  IDENTIFIER VALUES. IF UGETIO IS CALLED WITH IOPT=2 OR
C                  IOPT=3, NEW UNIT IDENTIFIER VALUES ARE ESTABLISHED.
C                  SUBSEQUENT INPUT/OUTPUT IS PERFORMED ON THE NEW UNITS.
C
C COPYRIGHT        - 1978 BY IMSL, INC. ALL RIGHTS RESERVED.
C
C WARRANTY        - IMSL WARRANTS ONLY THAT IMSL TESTING HAS BEEN
C                  APPLIED TO THIS CODE. NO OTHER WARRANTY,
C                  EXPRESSED OR IMPLIED, IS APPLICABLE.
C
C-----
C
C          SUBROUTINE UGETIO(IOPT,NIN,NOUT)
C                  SPECIFICATIONS FOR ARGUMENTS
C          INTEGER          IOPT,NIN,NOUT
C                  SPECIFICATIONS FOR LOCAL VARIABLES
C          INTEGER          NIND,NOUDD
C          DATA            NIND/5/,NOUDD/6/
C
C                  FIRST EXECUTABLE STATEMENT
C          IF (IOPT.EQ.3) GO TO 10
C          IF (IOPT.EQ.2) GO TO 5
C          IF (IOPT.NE.1) GO TO 9005
C          NIN = NIND
C          NOUT = NOUDD
C          GO TO 9005
C          5 NIND = NIN
C            GO TO 9005
C          10 NOUDD = NOUT
C          9005 RETURN

```

```

      END
C   IMSL ROUTINE NAME   -  USPKD
C
C-----
C
C   COMPUTER           -  CONVEX
C
C   LATEST REVISION    -  NOVEMBER 1, 1984
C
C   PURPOSE            -  NUCLEUS CALLED BY IMSL ROUTINES THAT HAVE
C                       CHARACTER STRING ARGUMENTS
C
C   USAGE              -  CALL USPKD  (PACKED,NCHARS,UNPAKD,NCHMTB)
C
C   ARGUMENTS          PACKED - CHARACTER STRING TO BE UNPACKED.(INPUT)
C                       NCHARS - LENGTH OF PACKED. (INPUT)  SEE REMARKS.
C                       UNPAKD - CHARACTER ARRAY TO RECEIVE THE UNPACKED
C                               REPRESENTATION OF THE STRING. (OUTPUT)
C                       NCHMTB - NCHARS MINUS TRAILING BLANKS. (OUTPUT)
C
C   PRECISION/HARDWARE -  SINGLE/ALL
C
C   REQD. IMSL ROUTINES -  NONE
C
C   REMARKS  1.  USPKD UNPACKS A CHARACTER STRING INTO A CHARACTER ARRAY
C               IN (A1) FORMAT.
C               2.  UP TO 129 CHARACTERS MAY BE USED.  ANY IN EXCESS OF
C                   THAT ARE IGNORED.
C
C   COPYRIGHT          -  1984 BY IMSL, INC.  ALL RIGHTS RESERVED.
C
C   WARRANTY           -  IMSL WARRANTS ONLY THAT IMSL TESTING HAS BEEN
C                       APPLIED TO THIS CODE.  NO OTHER WARRANTY,
C                       EXPRESSED OR IMPLIED, IS APPLICABLE.
C-----
C   SUBROUTINE USPKD  (PACKED,NCHARS,UNPAKD,NCHMTB)
C                       SPECIFICATIONS FOR ARGUMENTS
C   INTEGER          NC,NCHARS,NCHMTB
C
C   CHARACTER        UNPAKD(1),IBLANK
C   CHARACTER*(1)    PACKED(1)
C   DATA            IBLANK /' '/
C                       INITIALIZE NCHMTB
C   NCHMTB = 0
C                       RETURN IF NCHARS IS LE ZERO

```

```

        IF(NCHARS.LE.0) RETURN
C
        NC = MINO (129,NCHARS)          SET NC=NUMBER OF CHARS TO BE DECODED
        DO 5 I=1,NC
            UNPAKD(I) = PACKED(I)
        5 CONTINUE
150 FORMAT (129A1)
C
C                                     CHECK UNPAKD ARRAY AND SET NCHMTB
C                                     BASED ON TRAILING BLANKS FOUND
        DO 200 N = 1,NC
            NN = NC - N + 1
            IF(UNPAKD(NN) .NE. IBLANK) GO TO 210
200 CONTINUE
        NN = 0
210 NCHMTB = NN
        RETURN
        END
C*****
C                                     END OF IMSL ROUTINES
C*****

```

# Appendix C

Output from FINDRES3

LOCATION OF RESONANCES FOR A SPHERICAL DROPLET

Copyright 1988 by C. D. Cantrell

All rights reserved

Last revision 25 July 1988

Input parameters:

n.re = 1.4700000 n.im = 0.0000000

lowest order = 30 highest order = 50 max. iterations =10000

no. of sig. figs. = 5

Resonances in a( 30):

1	2.742231929990528d+01	-7.096899011552049d-02	96
2	2.742231929593419d+01	-7.096898367984598d-02	104
3	3.240112097333694d+01	-1.843538679278955d+01	76
4	3.423737074023355d+01	-3.067655352659345d+00	180

Resonances in b( 30):

1	2.385632349099544d+01	-9.554286172147178d-04	88
2	2.710829266931894d+01	-3.949674321353046d-02	87
3	2.997667143644297d+01	-1.827002417138662d-01	110
4	3.538514978871977d+01	-5.304673719469976d+01	97

Resonances in a( 31):

1	2.502711540947323d+01	-1.104306446252708d-03	116
2	2.818847730351198d+01	-5.894459838849030d-02	103
3	3.810665003344501d+01	-1.710117477851368d+00	177
4	3.503037281134981d+01	-3.295071303504743d+00	151

Resonances in b( 31):

1	2.458070894956941d+01	-7.098510418781318d-04	97
2	2.786509819930301d+01	-3.302355409931680d-02	112
3	3.074695296485995d+01	-1.682725269109250d-01	121
4	3.611955948452390d+01	-5.438125146582615d+01	60

Resonances in a( 32):

1	2.575251130597253d+01	-8.157502741285476d-04	100
2	2.895262241465906d+01	-4.867636475040166d-02	83
3	4.212986134775409d+01	-5.718900280684982d+00	164
4	4.097480244665132d+01	-1.385131996026184d+00	210

Resonances in b( 32):

1	2.530402310092884d+01	-5.258959717539943d-04	95
2	2.862041872797096d+01	-2.744201713427551d-02	83
3	3.151645727749318d+01	-1.542501508319492d-01	102
4	3.677942189270622d+01	-5.290160507587177d+01	64

Resonances in a( 33):

1	3.246216330524961d+01	-2.862677788081956d-01	113
2	2.971474978568401d+01	-3.997048636614697d-02	89
3	3.246216330524961d+01	-2.862677788081916d-01	199
4	3.999889632252794d+01	-1.674657957570727d+00	209

Resonances in b( 33):

1	2.602633272292346d+01	-3.885791423099799d-04	97
2	2.937425133957858d+01	-2.266776953496812d-02	93

3	3.228521687097681d+01	-1.407031550487118d-01	117
4	3.763918714384834d+01	-5.558930811515197d+01	81
Resonances in a( 34):			
1	2.719985900483804d+01	-4.418496685024565d-04	96
2	3.047487217805662d+01	-3.264142637008277d-02	97
3	3.324030489265568d+01	-2.569446438277360d-01	165
4	3.739571087632519d+01	-3.897447447997224d+00	78
Resonances in b( 34):			
1	2.674769860713221d+01	-2.863884883868432d-04	51
2	3.012660372367553d+01	-1.861577648924068d-02	64
3	3.305324926672554d+01	-1.276974038349329d-01	114
4	3.855594211913341d+01	-5.677527727105681d+01	90
Resonances in a( 35):			
1	2.792196291115934d+01	-3.240834654230972d-04	102
2	3.123301848331985d+01	-2.651424270354142d-02	83
3	3.401781082438410d+01	-2.296005368910599d-01	136
4	4.181559739177845d+01	-1.528864334789008d+00	270
Resonances in b( 35):			
1	2.746817608060685d+01	-2.105997015383804d-04	58
2	3.087749349040710d+01	-1.520270538941921d-02	83
3	3.382055819725996d+01	-1.152926887786277d-01	111
4	3.943622152036358d+01	-5.350449847384734d+01	46
Resonances in a( 36):			
1	2.864310918940238d+01	-2.372101119145721d-04	70
2	3.198923171513989d+01	-2.142645711754557d-02	73
3	3.479462838684616d+01	-2.042177260427872d-01	109
4	3.896478633359824d+01	-4.232800295929599d+00	158
Resonances in b( 36):			
1	2.818781552235597d+01	-1.545197739840964d-04	65
2	3.162694691403236d+01	-1.234868484446504d-02	52
3	3.458713514939812d+01	-1.035410119510873d-01	77
4	3.890286881597515d+01	-5.017077196373927d+01	67
Resonances in a( 37):			
1	2.936335646917350d+01	-1.732852234249758d-04	47
2	3.274356666392934d+01	-1.722937840849071d-02	85
3	3.610932104618045d+01	-1.548430753194039d+01	32
4	3.974946363416557d+01	-4.381433059944744d+00	83
Resonances in b( 37):			
1	2.890666289733702d+01	-1.129728644895489d-04	37
2	3.237499793128277d+01	-9.978660942550423d-03	54
3	3.535296121657450d+01	-9.248514575452135d-02	74
4	3.985629480272978d+01	-5.804139564222065d+01	63
Resonances in a( 38):			
1	3.008275774860530d+01	-1.263529326909724d-04	48
2	3.349608740025882d+01	-1.378885804113019d-02	70
3	3.634595418446542d+01	-1.592647407953851d-01	133

4	4.639998157713171d+01	-1.761360899883029d+00	235
Resonances in b( 38):			
1	2.962475997162863d+01	-8.262902319989159d-05	54
2	3.312168620100530d+01	-8.023780057040892d-03	56
3	3.611800918608012d+01	-8.215755126323894d-02	113
4	3.965607890755555d+01	-5.953704685452601d+01	167
Resonances in a( 39):			
1	3.080136103911896d+01	-9.196488834794372d-05	63
2	3.424686479103556d+01	-1.098550582001744d-02	89
3	3.712023096860428d+01	-1.395582253369963d-01	234
4	5.045204355082647d+01	-4.764055657593230d+00	182
Resonances in b( 39):			
1	3.034214532952340d+01	-6.030729958242614d-05	57
2	3.386705609034361d+01	-6.421534531094076d-03	39
3	3.688224577073938d+01	-7.257969850005311d-02	117
4	3.991005429199778d+01	-5.597772123119986d+01	93
Resonances in a( 40):			
1	3.151920994176338d+01	-6.682650931447407d-05	68
2	3.499597416848636d+01	-8.714404058902480d-03	53
3	3.789353967076319d+01	-1.216905598626417d-01	136
4	5.009485642714105d+01	-2.078958780514281d+00	143
Resonances in b( 40):			
1	3.105885398344160d+01	-4.391044493131360d-05	35
2	3.461115506286738d+01	-5.116227713434645d-03	39
3	3.764563383642836d+01	-6.376178028804322d-02	88
4	4.109981068383356d+01	-2.189933777935628d+01	62
Resonances in a( 41):			
1	3.223634415901594d+01	-4.848270567369986d-05	101
2	3.574349319169698d+01	-6.884591680249842d-03	83
3	3.866576038933815d+01	-1.055641378609171d-01	144
4	5.111305253670977d+01	-2.150553839446371d+00	152
Resonances in b( 41):			
1	3.177491827646887d+01	-3.191926182304968d-05	64
2	3.535403245504970d+01	-4.058912825816567d-03	65
3	3.840813484428817d+01	-5.570313598473346d-02	126
4	4.179124144156270d+01	-6.114703135498844d+01	82
Resonances in a( 42):			
1	3.295279994493578d+01	-3.512119611169191d-05	74
2	3.648950012587835d+01	-5.417868684404795d-03	85
3	3.943676454677311d+01	-9.105767051311782d-02	135
4	5.317366683901601d+01	-3.118006209470592d+00	144
Resonances in b( 42):			
1	3.249036796899142d+01	-2.316157258786925d-05	43
2	3.609573836709970d+01	-3.207173905826309d-03	29
3	3.916971045911436d+01	-4.839235341645686d-02	102
4	4.436612320603114d+01	-5.729879531495242d+01	77

Resonances in a( 43):			
1	3.366861049977351d+01	-2.540697641220481d-05	90
2	3.723407228666139d+01	-4.247931249852386d-03	81
3	4.020662353368397d+01	-7.821105626652256d-02	124
4	4.567818395648440d+01	-6.601435448962149d-01	193
Resonances in b( 43):			
1	3.320523054247425d+01	-1.678757143763006d-05	75
2	3.683632277033672d+01	-2.524343257968402d-03	39
3	3.993032483399339d+01	-4.180867644305545d-02	107
4	4.576009343870788d+01	-4.024138142876898d+01	104
Resonances in a( 44):			
1	3.438380631461879d+01	-1.835213753744590d-05	135
2	3.797728492840731d+01	-3.319017181825948d-03	84
3	4.097512896281400d+01	-6.679845605124698d-02	151
4	5.171855861610187d+01	-1.367170569596717d+00	192
Resonances in b( 44):			
1	3.391953143636194d+01	-1.214787244947390d-05	97
2	3.757583471177765d+01	-1.979722689614176d-03	55
3	4.068994592070543d+01	-3.592296586747921d-02	129
4	4.800431136879445d+01	-2.457227665254376d+01	73
Resonances in a( 45):			
1	3.509841547179054d+01	-1.323950570693571d-05	78
2	3.871921036907052d+01	-2.584653161869625d-03	101
3	4.174227071530927d+01	-5.676476196607998d-02	129
4	4.723816500260146d+01	-5.951228076589490d-01	172
Resonances in b( 45):			
1	3.463329425704200d+01	-8.778361494564048d-06	98
2	3.831432180905045d+01	-1.547256819851820d-03	76
3	4.144854671946829d+01	-3.069914438164516d-02	138
4	4.873438688107876d+01	-2.171355762707154d+01	83
Resonances in a( 46):			
1	3.581246390624107d+01	-9.539323314573669d-06	103
2	3.945991743095881d+01	-2.006468236664665d-03	101
3	4.250802436354461d+01	-4.798437019189157d-02	149
4	5.334692148073847d+01	-1.235730451475994d+00	175
Resonances in b( 46):			
1	3.534654096122345d+01	-6.335097146724996d-06	91
2	3.905182981597471d+01	-1.205313137958769d-03	106
3	4.220610613213081d+01	-2.609570558359954d-02	127
4	4.511395861571361d+01	-1.247135904211044d-01	143
Resonances in a( 47):			
1	3.652597563393078d+01	-6.864910508361380d-06	142
2	4.019947105270811d+01	-1.552991729767696d-03	116
3	4.325861748679995d+01	-5.520395548874255d-02	171
4	4.879512524767002d+01	-5.340333828139809d-01	153
Resonances in b( 47):			

1	3.605929201734545d+01	-4.566080557497400d-06	121
2	3.978840235862788d+01	-9.360293557217260d-04	117
3	4.296260946729466d+01	-2.206725396742651d-02	101
4	4.588100921693566d+01	-1.142414392451814d-01	183
Resonances in a( 48):			
1	4.093791776999657d+01	-1.212936616094234d-03	237
2	4.093793212670174d+01	-1.198611252214838d-03	102
3	4.403500119067697d+01	-3.402188580094685d-02	194
4	4.957268753759639d+01	-5.047617089693385d-01	128
Resonances in b( 48):			
1	3.677156654760271d+01	-3.287052230335829d-06	97
2	4.052408077851538d+01	-7.247681490479128d-04	94
3	4.371804861918737d+01	-1.856600193412273d-02	103
4	4.664748155593576d+01	-1.042361336291061d-01	299
Resonances in a( 49):			
1	4.167535746395745d+01	-9.226158794241183d-04	176
2	4.167535748416825d+01	-9.225877149250982d-04	75
3	4.479665188852214d+01	-2.815097639614053d-02	196
4	6.433316662542035d+01	-9.032184459423117d+00	151
Resonances in b( 49):			
1	3.748338245310743d+01	-2.363541039321743d-06	76
2	4.125890406599493d+01	-5.596186403235905d-04	108
3	4.447242195736428d+01	-1.554316126063190d-02	97
4	4.741352207097353d+01	-9.458063718165075d-02	191
Resonances in a( 50):			
1	4.241179986950553d+01	-7.083592584786770d-04	128
2	4.241179986950576d+01	-7.083592580462872d-04	116
3	4.555664062397398d+01	-2.334514805029950d-02	209
4	6.109251973687182d+01	-1.838592550226023d+00	225
Resonances in b( 50):			
1	3.819475652439325d+01	-1.697589727421056d-06	83
2	4.199290886320236d+01	-4.309508124109688d-04	101
3	4.522573397527089d+01	-1.295018065273002d-02	139
4	4.817896056380602d+01	-8.546358892950855d-02	202

LOCATION OF RESONANCES FOR A SPHERICAL DROPLET

Copyright 1988 by C. D. Cantrell

All rights reserved

Last revision 25 July 1988

Input parameters:

n.re = 1.4700000 n.im = 0.0000000

lowest order = 50 highest order = 60 max. iterations =10000

no. of sig. figs. = 6

Resonances in a( 50):

1	4.241179986950553d+01	-7.083592584786770d-04	128
2	4.241179986950576d+01	-7.083592580462872d-04	116
3	4.555664062397398d+01	-2.334514805029950d-02	211
4	6.109251973687182d+01	-1.838592550226023d+00	227

Resonances in b( 50):

1	3.819475652439325d+01	-1.697589727421056d-06	83
2	4.199290886320236d+01	-4.309508124109688d-04	101
3	4.522573397527086d+01	-1.295018065268326d-02	126
4	4.817896056380602d+01	-8.546358892950855d-02	202

Resonances in a( 51):

1	4.314730829437765d+01	-5.425364729685307d-04	141
2	4.314730829437752d+01	-5.425364732291920d-04	122
3	4.915210652735821d+01	-1.517497594209962d-01	174
4	6.667389416878429d+01	-8.353888495269054d+00	142

Resonances in b( 51):

1	3.890570453919567d+01	-1.217961573407214d-06	94
2	4.272612951827932d+01	-3.310248291134592d-04	126
3	4.597799475198455d+01	-1.073979657953637d-02	137
4	5.539681305587212d+01	-2.524780387434901d+01	123

Resonances in a( 52):

1	4.388192803849040d+01	-4.145680802413111d-04	137
2	4.388192803849091d+01	-4.145680801036101d-04	124
3	6.072687464193101d+01	-1.312850128258043d+00	183
4	6.483321494270967d+01	-1.624870063425922d+00	266

Resonances in b( 52):

1	3.961624134918569d+01	-8.729367814801706d-07	121
2	4.345859817556100d+01	-2.536540268145518d-04	93
3	4.672921928317792d+01	-8.866880110064441d-03	116
4	4.970809986395409d+01	-6.879299670471177d-02	203

Resonances in a( 53):

1	4.461570098390948d+01	-3.160842889482016d-04	124
2	4.461570098394216d+01	-3.160819134709477d-04	196
3	6.386665112349775d+01	-1.743654826724760d+00	173
4	5.248764381499537d+01	-9.871200015339280d+00	42

Resonances in b( 53):

1	4.032638095711344d+01	-6.250236190905599d-07	149
2	4.419034488924042d+01	-1.939182363433944d-04	89

3	4.747942673502058d+01	-7.289076700052619d-03	136
4	7.551611202624420d+01	-6.646784383253701d+01	175
Resonances in a( 54):			
1	4.534866584297490d+01	-2.404818760269900d-04	128
2	4.858263809545855d+01	-1.057340762753464d-02	259
3	6.475410482972953d+01	-1.670965505922525d+00	207
4	6.233659710421428d+01	-1.205038332851496d+00	174
Resonances in b( 54):			
1	4.492139775068382d+01	-1.479228513945380d-04	166
2	4.492139775068344d+01	-1.479228526226483d-04	80
3	4.822863966919349d+01	-5.967247860858101d-03	122
4	5.410833485251781d+01	-1.610040552348584d-01	157
Resonances in a( 55):			
1	4.608085840892944d+01	-1.825925448133545d-04	106
2	4.933581984657980d+01	-8.587466776147636d-03	159
3	5.224037184244141d+01	-9.228536260585733d-02	154
4	5.411639571499556d+01	-1.050777927362142d+01	39
Resonances in b( 55):			
1	4.565178302189615d+01	-1.125984884795283d-04	134
2	4.565178302189629d+01	-1.125984883136825d-04	96
3	4.897688327944737d+01	-4.865732787682963d-03	105
4	2.073286625437000d+03	-2.100143781512885d+03	198
Resonances in a( 56):			
1	4.681231180189289d+01	-1.383687587296730d-04	130
2	5.008775899585795d+01	-6.948846456732532d-03	166
3	5.301040042900269d+01	-8.055972485963903d-02	132
4	5.496698268515389d+01	-9.747370639952511d+00	43
Resonances in b( 56):			
1	4.638152526963608d+01	-8.553594053134503d-05	139
2	4.638152526963692d+01	-8.553599326409238d-05	83
3	4.972418467124836d+01	-3.952454140574446d-03	115
4	5.564498530926253d+01	-1.412239754132795d-01	186
Resonances in a( 57):			
1	4.754305670345419d+01	-1.046598764642987d-04	102
2	5.083850206735940d+01	-5.603081490362127d-03	132
3	5.377948763900948d+01	-6.999852624727936d-02	169
4	-8.782691329876816d+03	-6.108727620436458d+04	134
Resonances in b( 57):			
1	4.711064749627473d+01	-6.485134930557307d-05	157
2	4.711064749627465d+01	-6.485134921547959d-05	102
3	5.047057220698734d+01	-3.198896411898793d-03	105
4	5.641265139899401d+01	-1.315925652850174d-01	327
Resonances in a( 58):			
1	4.827312158340559d+01	-7.903171285083493d-05	110
2	5.158809629932925d+01	-4.502682675688895d-03	127
3	5.454758007285108d+01	-6.054180329025489d-02	139

4	5.690188011067879d+01	-3.535644349217422d+01	52
Resonances in b( 58):			
1	4.783917126480235d+01	-4.907686945702070d-05	128
2	4.783917126480243d+01	-4.907686943805829d-05	97
3	5.121607493059142d+01	-2.579977842542763d-03	144
4	1.787656054129822d+05	-4.603628693562564d+04	170
Resonances in a( 59):			
1	4.900253287196950d+01	-5.956018273142969d-05	115
2	5.233658887813507d+01	-3.606720206207627d-03	147
3	5.531463189007033d+01	-5.212361857500288d-02	222
4	5.811924948282893d+01	-1.946617450866279d+01	37
Resonances in b( 59):			
1	4.856711681636529d+01	-3.707260677554292d-05	99
2	4.856711681636507d+01	-3.707260671746313d-05	149
3	5.196072207845761d+01	-2.073864186392532d-03	190
4	5.794672738861047d+01	-1.130221270251524d-01	157
Resonances in a( 60):			
1	4.973131521053553d+01	-4.481720274722525d-05	155
2	5.308402620788951d+01	-2.880120242987207d-03	153
3	5.608060555225899d+01	-4.467355715108566d-02	267
4	5.887654635694136d+01	-2.132335278175509d-01	135
Resonances in b( 60):			
1	4.929450317946388d+01	-2.795611613618293d-05	137
2	4.929450317946398d+01	-2.795611629541625d-05	154
3	5.270454267668360d+01	-1.661712768072651d-03	200
4	5.871314620352274d+01	-1.041516136665461d-01	162

LOCATION OF RESONANCES FOR A SPHERICAL DROPLET

Copyright 1988 by C. D. Cantrell

All rights reserved

Last revision 25 July 1988

Input parameters:

n.re = 1.4700000 n.im = 0.0000000

lowest order = 100 highest order = 100 max. iterations =10000

no. of sig. figs. = 10

Resonances in a(100):

1	7.853166267292634d+01	-2.027730261995531d-10	98
2	8.244419243006529d+01	-7.316179858847182d-08	80
3	8.598163159976563d+01	-7.456265657986963d-06	84
4	8.927270112001059d+01	-2.973144808480295d-04	134
5	9.237340379964152d+01	-5.281849980076242d-03	113
6	9.530926392311352d+01	-4.257160348329855d-02	126
7	9.813877611714970d+01	-1.599183268657998d-01	133
8	1.039360747470430d+02	-2.752283143281307d+01	45
9	1.217541266653523d+02	-2.028727221461355d+00	421
10	1.172561521879046d+02	-1.404305669689679d+00	357

Resonances in b(100):

1	7.806661147042260d+01	-1.335964914135102d-10	65
2	8.200193103405789d+01	-4.583724595442737d-08	93
3	8.556815728573166d+01	-4.452537121248629d-06	108
4	8.889703539390320d+01	-1.690872297862671d-04	91
5	9.205130479278758d+01	-2.855802086025470d-03	129
6	9.506586454221042d+01	-2.184158868967949d-02	113
7	9.798601254568958d+01	-7.765320182812076d-02	118
8	1.008619354832529d+02	-1.546325039475331d-01	120
9	1.037030409297346d+02	-2.227231741893465d-01	160
10	1.367630811844319d+03	-7.023356285666945d+04	45

LOCATION OF RESONANCES FOR A SPHERICAL DROPLET

Copyright 1988 by C. D. Cantrell

All rights reserved

Last revision 25 July 1988

Input parameters:

n.re = 1.4700000 n.im = 0.0000000

lowest order = 400 highest order = 400 max. iterations =10000

no. of sig. figs. = 14

Resonances in a(400):

1	3.178023378142063d+02	-1.779809103533928d-1110003
2	3.178023378142122d+02	1.746474578083257d-1110000
3	3.218481127785997d+02	-1.965191353925037d-17 84
4	3.257696993784073d+02	-9.957617660078389d-0710000
5	3.295846435084006d+02	3.355544339552613d-0710000

Resonances in b(400):

1	3.131421814887977d+02	-1.108248145432220d-15 153
2	3.173403755161717d+02	-7.704492411427572d-20 1093
3	3.213915776466965d+02	-2.075519660230418d-16 1322
4	3.253188886244375d+02	-2.301976869784451d-1210001
5	3.291398733273542d+02	4.610115880199789d-1210000

LOCATION OF RESONANCES FOR A SPHERICAL DROPLET

Copyright 1988 by C. D. Cantrell

All rights reserved

Last revision 25 July 1988

Input parameters:

n.re = 1.4700000 n.im = 0.0000000

lowest order = 418 highest order = 418 max. iterations =10000

no. of sig. figs. = 10

Resonances in a(418):

1	3.708035476502230d+02	-1.142023442995433d-11	82
2	3.740888591341063d+02	-2.697915615648040d-12	77
3	3.773338991915193d+02	-4.217614117281766d-12	101
4	3.805412617163936d+02	-9.999905162336487d-11	99
5	3.837132059769156d+02	-1.458398480167589d-09	85
6	3.868516870276012d+02	-2.053028165106940d-08	85
7	3.899583692869204d+02	-2.431224833757974d-07	89

Resonances in b(418):

1	3.704143630725936d+02	-2.324040908865826d-13	96
2	3.737091961342014d+02	-2.856902911494923d-12	75
3	3.769644335807272d+02	-8.641561709576437d-12	95
4	3.801827547235292d+02	-4.613872761411132d-11	111
5	3.833665240539122d+02	-7.935261163423574d-10	77
6	3.865178292546860d+02	-1.103767070675247d-08	71
7	3.896385070640293d+02	-1.289999193690414d-07	64