MEMORANDUM
RM-4549-ARPA
AUGUST 1965

COMPUTATIONAL ASPECTS OF INVERSE PROBLEMS IN ANALYTICAL MECHANICS, TRANSPORT THEORY, AND WAVE PROPAGATION

Harriet Kagiwada

PREPARED FOR:
ADVANCED RESEARCH PROJECTS AGENCY

The RAND Corporation
SANTA MONICA • CALIFORNIA
DISCLAIMER NOTICE

THIS DOCUMENT IS THE BEST QUALITY AVAILABLE.
COPY FURNISHED CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.
MISSING PAGE
NUMBERS ARE BLANK
AND WERE NOT FILMED
MEMORANDUM
RM-4549-ARPA
AUGUST 1965

COMPUTATIONAL ASPECTS OF INVERSE PROBLEMS IN ANALYTICAL MECHANICS, TRANSPORT THEORY, AND WAVE PROPAGATION
Harriet Kagiwada

This research is supported by the Advanced Research Projects Agency under Contract No. SD-79. Any views or conclusions contained in this Memorandum should not be interpreted as representing the official opinion or policy of ARPA.

DDC AVAILABILITY NOTICE
Qualified requesters may obtain copies of this report from the Defense Documentation Center (DDC).

The RAND Corporation

Approved for release by the Clearinghouse for Federal Scientific and Technical Information
PREFACE

Inverse problems are basic problems in science, in which physical systems are to be identified on the basis of experimental observations. It is shown in this Memorandum that a wide class of inverse problems may be readily solved with high speed computers and modern computational techniques. This is demonstrated by formulating and solving some inverse problems which arise in celestial mechanics, transport theory and wave propagation. FORTRAN programs are listed in the Appendix. Computational aspects of inverse problems are of interest to physicists, engineers and biologists who are engaged in system identification, in the planning of experiments and the analysis of data, and in the construction of mathematical models. This study was supported by the Advanced Research Projects Agency.

The author wishes to express her gratitude for the inspiration and guidance of Professor Suseo Ueno of the University of Kyoto, and of Dr. Richard Bellman and Dr. Robert Kalaba of the RAND Corporation.
SUMMARY

Inverse problems are basic problems in science, in which physical systems are to be identified on the basis of experimental observations. Inverse problems are especially important in the fields of astrophysics and astronomy, for their objects of investigation are frequently not observable in a direct fashion. Solar and stellar structure, for example, is estimated from the study of spectra, while the structure of a planetary atmosphere may be deduced from measurements of reflected sunlight.

We show that a wide class of inverse problems may now be solved with high speed computers and modern computational techniques. Many problems may be formulated in terms of systems of ordinary differential equations of the form

\[ \dot{x} = f(x, \alpha) . \]

Here, \( t \) is the independent variable, \( x \) is an \( n \)-dimensional vector whose components are the dependent variables, and \( \alpha \) is an \( m \)-dimensional vector whose components represent the structure of the system. For instance, in an orbit determination problem, Eqs. (1) are the dynamical equations of motion, and the masses of the bodies involved may be given by the vector \( \alpha \). When the system parameters and a complete set of initial conditions,

\[ x(0) = c , \]
are known, an integration of (1) produces the solution \( x(t) \) on the interval \( 0 < t < T \). This is done speedily and accurately with a digital computer.

On the other hand, in an inverse problem, the solution \( x(t) \) or some function of \( x(t) \) is known at various times, while the parameters are not directly observable. We wish to determine the structure of the system as given by the parameter vector, \( \alpha \), and a complete set of initial conditions, \( c \). We regard this as being a nonlinear boundary value problem in which the unknowns are some of the \( c \)'s and \( \alpha \)'s. We require that the solution agree with the observations,

\[
(3) \quad x(t_i) \approx b_i ,
\]

in some sense, e.g., in a least squares sense.

Frequently, problems which do not naturally occur in the form of systems of ordinary differential equations may be expressed in that form in an approximate representation. In this thesis, we show how we may reduce a partial differential-integral equation to a system of ordinary differential equations with the use of a quadrature formula. Also, we may express a partial differential equation, like the wave equation,

\[
(4) \quad \frac{\partial^2 u}{\partial x^2} = \frac{1}{c^2} \frac{\partial^2 u}{\partial t^2} ,
\]

in the desired form by applying Laplace transform methods which remove the time derivative. Other possibilities are clearly available.
Nonlinear boundary value problems can be solved by a variety of methods, which include quasilinearization, dynamic programming, and invariant imbedding. These techniques are especially suited to modern computers, for they reduce nonlinear boundary value problems to nonlinear initial value problems, which are more easily treated on digital computers.

These computational ideas are illustrated in this thesis by actually formulating and solving some inverse problems which arise in celestial mechanics, radiative transfer, neutron transport and wave propagation. In one of the problems, we estimate the stratification of a layered medium from reflection data. In another, we determine a variable wave velocity by observing a portion of the transients produced by a known stimulus. Numerical experiments are conducted to estimate the stability of the methods and the effect of the number and quality of observational measurements. Complete FORTRAN programs are given in the Appendices.

These computational aspects of inverse problems may prove to be of value to the physicist, engineer, or mathematical biologist who wishes to determine the structure of a system on the basis of observations. These ideas may be helpful in the planning of experiments and in the choice of apparatus. They may be used to design systems which have certain desired properties. In particular, these methods may be useful in the construction of stellar and planetary models.
TABLE OF CONTENTS

PREFACE .............................................. iii
SUMMARY ........................................... v

Section
I. INTRODUCTION ....................................... 1
   1. Introduction ...................................... 1
   2. Determination of Potential ....................... 7
   3. Quasilinearization, System Identification and Nonlinear Boundary Value Problems .... 11
   4. Solution of the Potential Problem ............... 19
   REFERENCES ....................................... 28

II. INVERSE PROBLEMS IN RADIATIVE TRANSFER: LAYERED MEDIA ........................................ 31
   1. Introduction ...................................... 31
   2. Invariant Imbedding ................................ 32
   3. The Diffuse Reflection Function for an Inhomogeneous Slab ................................. 33
   4. Gaussian Quadrature ................................ 41
   5. An Inverse Problem ................................ 44
   6. Formulation as a Nonlinear Boundary Value Problem ........................................ 49
   7. Numerical Experiments I: Determination of \( c \), the Thickness of the Lower Layer .... 50
   8. Numerical Experiments II: Determination of \( T \), the Overall Optical Thickness ...... 54
   10. Discussion ....................................... 57
   REFERENCES ....................................... 58

III. INVERSE PROBLEMS IN RADIATIVE TRANSFER: NOISY OBSERVATIONS ................................. 60
   1. Introduction ...................................... 60
   2. An Inverse Problem ................................ 60
   3. Formulation as a Nonlinear Boundary Value Problem ........................................ 62
   4. Solution via Quasilinearization .................... 63
   5. Numerical Experiments I: Many Accurate Observations ....................................... 68
   8. Numerical Experiments IV: Effect of Criterion .................................................. 75
   REFERENCES ....................................... 80
### IV. INVERSE PROBLEMS IN RADIATIVE TRANSFER: ANISOTROPIC SCATTERING

1. Introduction .......................... 81  
2. The S Function .......................... 82  
3. An Inverse Problem .................... 88  
4. Method of Solution .................... 90  
   REFERENCES ............................ 93

### V. AN INVERSE PROBLEM IN NEUTRON TRANSPORT THEORY

1. Introduction .......................... 95  
2. Formulation ........................... 96  
3. Dynamic Programming .................. 98  
4. An Approximate Theory ................. 100  
5. A Further Reduction ................... 104  
6. A Computational Procedure .............. 104  
7. Computational Results .................. 106  
   REFERENCES ............................ 115

### VI. INVERSE PROBLEMS IN WAVE PROPAGATION: MEASUREMENTS OF TRANSIENTS

1. Introduction .......................... 117  
2. The Wave Equation ..................... 118  
3. Laplace Transforms .................... 119  
4. Formulation ........................... 121  
5. Solution via Quasilinearization ........ 122  
6. Example 1 — Homogeneous Medium, Step  
   Function Force  ....................... 124  
7. Example 2 — Homogeneous Medium, Delta- 
   Function Input  ....................... 129  
8. Example 3 — Homogeneous Medium with  
   Delta-Function Input ................... 132  
9. Discussion ............................ 136  
   REFERENCES ............................ 137

### VII. INVERSE PROBLEMS IN WAVE PROPAGATION: MEASUREMENTS OF STEADY STATES

1. Introduction .......................... 138  
2. Some Fundamental Equations ............ 140  
3. Invariant Imbedding and the Reflection 
   Coefficient  ......................... 141  
4. Production of Observations ............. 145  
5. Determination of Refractive Index ...... 148  
6. Numerical Experiments .................. 151  
7. Discussion ............................ 155  
   REFERENCES ............................ 156
TABLE OF CONTENTS (continued)

VIII. DISCUSSION ........................................ 157
REFERENCES ........................................... 159

APPENDICES – THE FORTRAN PROGRAMS ................. 165

APPENDIX A – Programs for Orbit Determination .. 166
APPENDIX B – Programs for Radiative Transfer –
Layered Media ........................................ 177
APPENDIX C – Programs for Radiative Transfer –
Noisy Observations ................................. 209
APPENDIX D – Program for Radiative Transfer –
Anisotropic Scattering .............................. 239
APPENDIX E – Programs for Neutron Transport .. 251
APPENDIX F – Programs for Wave Propagation:
    Measurements of Transients ...................... 267
APPENDIX G – Programs for Wave Propagation:
    Measurements of Steady States ................. 288
1. INTRODUCTION

Inverse problems are fundamental problems of science [1–12]. Man has always sought knowledge of a physical system beyond that which is directly observable. Even today, we try to understand the dynamical processes of the deep interior of the sun by observing the radiation emerging from the sun's surface. We deduce the potential field of an atom from nuclear scattering experiments. The underlying theme is the relationship between the internal structure of a system and the observed output. The hidden features of the system are to be extracted from the experimental data.

Mathematical treatment of physical problems has been devoted almost exclusively to the "direct problem." A complete picture of the system is assumed to be given, and equations are derived which describe the output as a function of the system parameters. The inverse problem is to determine the parameters and structure of a system as a function of the observed output.
One can solve a given inverse problem by solving a series of direct problems: by assuming different sets of parameters, determining the corresponding outputs from the theoretical equations, and comparing theoretical versus experimental results. By trial and error, one may find a solution which approximately agrees with the experimental data. This is not a very efficient procedure. Another way to solve an inverse problem is to solve analytically for the unknown parameters as functions of the measurements. This method generally requires much abstract mathematics and simple approximations of complex functions. The resultant inverse solution may be valid only in very special circumstances.

What we seek are efficient, systematic procedures for solving a wide class of inverse problems - procedures which are suitable for execution on high speed digital computers. Computers are currently capable of integrating large systems of ordinary differential equations, given a complete set of initial conditions, with high accuracy. We would like to formulate our problems in terms of systems of ordinary differential equations. Partial differential equations, such as the wave equation,

\[ \frac{\partial^2 u(x,t)}{\partial x^2} = \frac{1}{c^2} \frac{\partial^2 u(x,t)}{\partial t^2}, \]
may be reduced to systems of ordinary differential equations in several ways which include the use of Laplace transform methods, Fourier decomposition, and finite difference schemes. Integro–differential equations, which frequently occur in transport theory, may be reduced to systems of ordinary differential equations by approximating the definite integrals by finite sums using Gaussian and other quadrature formulas. Other means of formulating problems in terms of ordinary differential equations are possible.

We desire to formulate our inverse problem in such a way that we deal with ordinary differential equations. First, as we shall show, we may express the problem as a nonlinear boundary value problem, in which we seek a complete set of initial conditions. The unknown system parameters will be calculated directly from the initial conditions. Next, we resolve the nonlinear boundary value problem, ordinarily a difficult task, by the use of some sophisticated techniques [13–24]. We may replace the nonlinear boundary value problem by a rapidly converging sequence of linear boundary value problems via the technique of quasilinearization [1–3,16,17]. We may, alternatively, treat the problem as a multi-stage decision process with the use of dynamic programming [18]. Or, we may solve directly for the missing initial conditions by applying the concept of invariant
imbedding \([13,24]\). From the solution of the nonlinear boundary value problem, we immediately obtain knowledge of the internal structure of the system.

In this thesis, we discuss some of these relatively new concepts, computational techniques, and applications. Our examples from celestial mechanics, transport theory and wave propagation are physically motivated. No specialized background is required on the part of the reader beyond a knowledge of elementary physics. We intend to be self-contained in the mathematical derivations, except for those matters which are well-treated elsewhere, such as dynamic programming, linear programming, and the numerical inversion of Laplace transforms. Again, no special mathematical knowledge is needed beyond the level of ordinary differential equations and linear algebraic equations. We will, however, assume that we have at our disposal a high-speed digital computer with a memory of about 32,000 words, plus a library of computer routines for numerical integration, matrix inversion, and linear programming. Our basic assumption is that our computer can integrate large systems of ordinary differential equations rapidly and accurately \([25,26]\).

In the first chapter, we wish to emphasize some important ideas. We are given geocentric observations of a heavenly body, taken at various times \([27-29]\). The orbit of this body lies in the potential field created by the sun and an unknown perturbing mass. We show how the mass may be identified and the orbital elements found. For simplicity,
we assume that the position of the perturbing mass is given; if desired, the position as a function of time could also be estimated. Since we are virtually forced by our modern computers to take a fresh look at old problems, we are not concerned with conic sections. A new methodology, based on high speed digital computers, is developed. The technique of quasilinearization, described in this chapter, enables us to solve this inverse problem with a minimum of effort. In spite of the newness of this solution of a long-standing problem in celestial mechanics, we employ this example for purely illustrative purposes.

Transport theory is intimately concerned with the determination of radiation fields within scattering and absorbing media [30–38]. Our first problem in radiative transfer (Chapter Two) serves to exemplify the philosophy and application of invariant imbedding. We derive the basic integro-differential equation for the diffuse reflection function, and we reduce it to a system of ordinary differential equations by the method of Gaussian quadrature. Then we formulate an inverse problem for the determination of layers in a medium from knowledge of the diffusely reflected light. We outline the computational procedure, and we present our results. In Chapter Three, our setting is again an inhomogeneous scattering medium. We investigate the effects of errors in our measurements, the number and quality of the observations, and the criterion function, on the estimates of the medium. Our criteria are either of least squares type, which leads to linear algebraic equations, or of
minimax form, which is suitable for linear programming. We also consider a variation of the inverse problem, the construction of a model atmosphere according to certain specifications. In Chapter Four, we consider an anisotropically scattering medium. The phase function is to be determined on the basis of measurements of diffusely reflected radiation in various directions.

An inverse problem in neutron transport (Chapter Five) is solved in a novel way. The dynamic programming approach leads to a determination of absorption coefficients in a rod, from measurements of internal fields. The calculation is done by an exact method, and is compared with a calculation based on an approximate theory. The approximate theory is accurate and less costly in computing time.

As we have already mentioned, the partial differential wave equation may be reduced to a system of ordinary differential equations by Laplace transform methods or by Fourier decompositions. In Chapter Six, we deal with ordinary differential equations for the Laplace transforms of the disturbances. In these equations, time appears only as a parameter. Our measurements of the disturbances at various times are converted to the corresponding transforms by means of Gaussian quadrature. We solve a nonlinear boundary value problem in order to determine the system parameters. The inverse Laplace transforms may be obtained by a numerical inversion technique. [22].
In Chapter Seven, we use a decomposition of the form 
\( u(x,t) = u(x)e^{-i\omega t} \), corresponding to a steady-state situation of wave propagation. We probe an inhomogeneous slab with waves of different frequencies and we "measure" the reflection coefficients. We wish to determine the index of refraction as a function of distance in the medium. Invariant imbedding leads to ordinary differential equations for the reflection coefficients, with known initial conditions. The unknown index of refraction in the equations and the observations of terminal values of the reflection coefficients make this a nonlinear boundary value problem. Quasi-linearization is used to solve the problem, and computational results are presented.

The final chapter is a general discussion of inverse problems. Appendices of all the FORTRAN programs written for the computational experiments are included.

2. \textbf{DETERMINATION OF POTENTIAL}

Consider the motion of a particle (or a wave) in a potential field \( V = V(x, y, z; k_1, k_2, \ldots, k_n) \) where we recognize the dependence on physical parameters \( k_1, k_2, \ldots, k_n \). Suppose that these parameters are unknown,
and that we have observations of the motion of the particle at various times. We wish to determine the potential function on the basis of these measurements.

Consider the following situation. A heavenly body H of mass m moves in the potential field created by the sun and a perturbing body P, whose masses are M and m_p, respectively, and m \ll m_p \ll M. All of the bodies concerned lie in the ecliptic plane. The potential energy varies inversely as the distance from the sun, r_s, and from the perturbing body, r_p:

\[ V = \frac{k_s}{r_s} - \frac{k_p}{r_p} \]

Here, k_s and k_p are the parameters

\[ k_s = \gamma m M, \quad k_p = \gamma m m_p, \]

where \( \gamma \) is the constant of gravitation. The quantity k_s may be assumed to be known. We choose our units so that \( k_s = m \), or \( M = 1 \). The parameter k_p is unknown and \( k_p < k_s \). We wish to determine k_p and thus V by observing the motion of H.

Let us take the plane of the ecliptic to be the (x, y) plane. The sun is situated at the origin, the earth at the point (1, 0), and the perturbing body at the location (\( \zeta, \eta \)) = (4, 1). The earth only enters into the discussion as the point from which measurements are taken. Its mass is neglected. The potential function is
Angular observations of $H$ are made at various times $t_i, i = 1, 2, ..., 5$. Fig. 1 illustrates the physical situation. Each solid arrow points to $H$ at a given time $t_i$. The angle between the line of sight and the $x$ axis is the observation. For comparison, see the dashed arrows which point to $H$ when the mass of $P$ is exactly zero, i.e., when $k_p = 0$. It is obvious that $k_p$ is small.

The equations of motion are

$$\ddot{x} = \frac{-x}{(x^2+y^2)^{3/2}} + \frac{a(x-x)}{[(x-x)^2+(y-y)^2]^{3/2}},$$

$$\ddot{y} = \frac{-y}{(x^2+y^2)^{3/2}} + \frac{a(y-y)}{[(x-x)^2+(y-y)^2]^{3/2}},$$

where the parameter $\alpha$,

$$\alpha = \frac{k_p}{k_s} = \frac{m_p}{M},$$

is the mass of $P$ relative to the mass of the sun. At times $t_i$, we obtain the angular data $\theta(t_i)$ which are, in radians,

$$\begin{align*}
\theta(0.0) &= 0.0, \\
\theta(0.5) &= 0.252188, \\
\theta(1.0) &= 0.507584, \\
\theta(1.5) &= 0.763641, \\
\theta(2.0) &= 1.01929.
\end{align*}$$

We wish to determine $\alpha, x(0), \dot{x}(0), y(0), \dot{y}(0)$ so that the conditions
Fig. 1. Angular observations of a heavenly body.
\[
\tan \theta(t_i) = \frac{-y(t_i)}{1-x(t_i)}
\]

are fulfilled. This is a nonlinear multipoint boundary value problem. The solution of this problem gives the relative mass of the perturbing body and the orbit of H as a function of time. The potential (3) is determined when \( \alpha \) is known. We may consider the problem then to be the determination of the orbit [19,23,27-29].

For an arbitrary potential field, we are unable to express the solution analytically. We solve the problem computationally using the technique of quasilinearization [16,17].

3. QUASILINEARIZATION. SYSTEM IDENTIFICATION AND NONLINEAR BOUNDARY VALUE PROBLEMS

Consider a physical system or process which is described by the system of \( N \) equations

\[
\dot{x} = f(x, \alpha),
\]

where \( x \) is a vector of dimension \( N \), a function of independent variable \( t \), with the \( N \) initial conditions

\[
x(0) = c.
\]

The vector \( x \) describes the state of the system at "time" \( t \), and \( \alpha \) is a parameter vector of the system. With \( \alpha \) given, Eqs. (1) and (2) completely describe the system, for the state at any time \( t \), \( x(t) \), may be calculated by a numerical integration of (1) with initial conditions (2).

Now let us suppose that we have a system described by Eqs. (1), but \( \alpha \) is unknown to us, and the initial conditions (2) are also unknown. However, we are able to make
measurements of certain components of the state of the system at various times \( t_i \). We wish to identify the system by determining \( \alpha \), and we wish to find a complete set of initial conditions \( x(0) = c \) so that the system is fully described.

We think of the system parameter vector as if it were a dependent variable which satisfies the vector equation

\[
\dot{\alpha} = 0
\]

with the unknown initial conditions

\[
\alpha(0) = \alpha_0.
\]

The multipoint boundary value problem which we have before us is to find the complete set of initial conditions

\[
\begin{align*}
  x(0) &= c, \\
  \alpha(0) &= \alpha_0,
\end{align*}
\]

such that the solution of the nonlinear system

\[
\begin{align*}
  \dot{x} &= f(x, \alpha) \\
  \dot{\alpha} &= 0,
\end{align*}
\]

agrees with the boundary conditions

\[
x(t_i) = b_i,
\]

where \( b_i \) is the observed state of the system at time \( t_i \). Let us suppose that we have exactly \( R = N + M \) measurements of the first component of \( x \), where \( N \) is the dimension of \( x \) and \( M \) is the dimension of \( \alpha \).
The boundary conditions are readily modified for a two point boundary value problem, or for more than $R$ observations, or for other types of measurements, for example linear combinations of the components of $x$.

Our approach to the problem is one of successive approximations. We solve a sequence of linear problems. We assume only that large systems of ordinary differential equations, whether linear or nonlinear, may be accurately integrated numerically if initial conditions are prescribed, and that linear algebraic systems may be accurately resolved.

Let us define a new column vector $x$ of dimension $R$, having as its elements the components of the original vector $x$ and the components of $a$,

$$
\begin{bmatrix}
    x_1 \\
    x_2 \\
    \vdots \\
    x_R
\end{bmatrix}
= 
\begin{bmatrix}
    x_1 \\
    x_2 \\
    \vdots \\
    x_N \\
    a_1 \\
    a_2 \\
    \vdots \\
    a_M
\end{bmatrix}
$$

(8)

This vector of dependent variables $x(t)$ satisfies the system of nonlinear equations

(9) \quad \dot{x} = f(x)
according to (6), and it has the unknown initial conditions

\[ x(0) = c. \]

according to (5). The boundary conditions are

\[ x_1(t_i) = b_i, \quad i = 1, 2, \ldots, R. \]

Mathematically, we need not distinguish between the components of this new vector \( x \) as state variables or system parameters.

An initial approximation starts the calculations. We form an estimate of the initial vector \( c \), and we integrate system (9) to produce the solution \( x(t) \) over the time interval of interest, \( 0 \leq t \leq T \), via numerical integration. The quasilinearization procedure is applied iteratively until a convergence to a solution occurs, or the solution diverges.

Let us suppose that we have completed stage \( k \) of our calculations and we have the current approximation \( x^k(t) \). In stage \( k + 1 \), we wish to calculate a new approximation \( x^{k+1}(t) \).

The vector function \( x^{k+1}(t) \) is the solution of the linear system

\[ x^{k+1} = f(x^k) + J(x^k)(x^{k+1} - x^k), \]

where \( J(x) \) is the Jacobian matrix with elements

\[ J_{ij} = \frac{\partial f_i}{\partial x_j}. \]

Since \( x^{k+1} \) is a solution of a system of linear differential equations, we know from general theory that it may be represented as the sum of a particular solution, \( p(t) \), and a
A linear combination of $R$ independent solutions of the homogeneous equations, $h^i(t)$, $i = 1, 2, \ldots, R,$

$$x^{k+1}(t) = p(t) + \sum_{i=1}^{R} c^i h^i(t).$$

The function $p$ satisfies the equation

$$\dot{p} = f(x^k) + J(x^k) (p - x^k),$$

and for convenience we choose the initial conditions

$$p(0) = 0.$$

The functions $h^i$ are solutions of the homogeneous systems

$$\dot{h}^i = J(x^k) h^i,$$

and we choose the initial conditions

$$h^i(0) = \text{the unit vector with all of its components zero, except for the } i^{th} \text{ which is one.}$$

The $h^i(0)$ form a linearly independent set. If the interval $(0, T)$ is sufficiently small, the functions $h^i(t)$ are also independent. The solutions $p(t), h^i(t)$ are produced by numerical integration with the given initial conditions.

There are $R+1$ systems of differential equations, each with $R$ equations, making a total of $R(R+1)$ equations which are integrated at each stage of our calculations.

After the functions $p$ and $h^i$ have been found over the interval, we must combine them so as to satisfy the boundary conditions (11).
This results in a system of \( R \) linear algebraic equations for the determination of the \( R \) unknown multipliers \( c_j \) of the standard form

\[
A \mathbf{c} = \mathbf{B},
\]

where the elements of the \( R \times R \) matrix of coefficients \( A \) are

\[
A_{ij} = h_i^j(t_i),
\]

and the components of the \( R \)-dimensional column vector \( B \) are

\[
B_i = b_i - p_i(t_i).
\]

Having determined the multipliers, we now know a complete set of initial conditions for the \((k+1)\text{st}\) stage,

\[
c = x^{k+1}(0) = p(0) + \sum_{j=1}^{R} c_j h_j^0(0).
\]

Because of our choice of initial conditions for \( p \) and \( h_j \), the initial values for each component of the vector \( x \) are identical with the multipliers \( c_j \),

\[
c_i = x_i^{k+1}(0) = c_i, \quad i = 1, 2, \ldots, R.
\]

Furthermore, we have a new approximation to the system parameter vector \( \alpha \).
The new approximation $x^{k+1}(t)$ for the interval $(0, T)$ may be produced either by the integration of the linear equations with the initial conditions just found, or by the linear combination of $p(t)$ and $h(t)$. The $(k+1)^{st}$ cycle is complete and we are ready for the $(k+2)^{nd}$. The process may be repeated until no further change is noted in the vector $c$.

The quasilinearization procedure is analogous to Newton's method for finding roots of an equation $f(x) = 0$. If $x^0$ is an approximate value of one of the roots of $f(x)$, then an improved value $x^1$ is obtained by applying the Taylor expansion formula to $f(x)$, and neglecting higher derivatives,

$$f(x^1) = f(x^0) + (x^1 - x^0) \frac{\partial f(x^0)}{\partial x^0}.$$  

Thus, the next approximation of the root is

$$x^1 = x^0 - \frac{f(x^0)}{f'(x^0)}.$$  

In quasilinearization, if the function $x^0(t)$ is an approximate solution of the nonlinear differential equation,

$$\dot{x} = f(x),$$

then an improved solution $x^1(t)$ may be obtained in the following manner. The function $f(x)$ is expanded around the current estimate $x^0(t)$, neglecting higher derivatives,
(29) \[ f(x^1) = x(x^0) + (x^1 - x^0) \frac{\partial f(x^0)}{\partial x^0}. \]

The improved approximation \( x^1(t) \) is the solution of the linear equation,

(30) \[ x^1 = f(x^0) + (x^1 - x^0) \frac{\partial f(x^0)}{\partial x^0}. \]

The method is easily extended to vector functions, as we have seen. The sequence of functions \( x^1(t), x^2(t), x^3(t), \ldots \) may be shown to converge quadratically in the limit[17]. Practically speaking, a good initial approximation leads to rapid convergence, with the number of correct digits approximately doubling with each additional iteration. On the other hand, a poor initial approximation may lead to divergence.

The quasilinearization technique provides a systematic way of treating nonlinear boundary value problems. The computational solution of such a problem is broken up into stages, in which a large system of ordinary differential equations is integrated with known initial conditions, and a linear algebraic system is resolved. The initial value integration problem is well-suited to the digital computer. With the aid of a formula such as the trapezoidal rule,

(31) \[ \int_{t_0}^{t_n} f(t)dt \approx \frac{\Delta}{2} (f_0 + f_1) + \frac{\Delta}{2} (f_1 + f_2) + \ldots + \frac{\Delta}{2} (f_{n-1} + f_n). \]

the integral of a function over an interval is rapidly computed. Moreover, higher order methods such as the Runge-Kutta and the Adams-Moulton, usually of fourth order, make it
possible to solve the integration problem accurately and rapidly. The accuracy may be as high as one part in $10^8$. The solution is available at each grid point $t_0, t_0 + \Delta, t_0 + 2\Delta, \ldots, t_n$, and may be stored in the computer's memory for use at some future time. The rapid access storage capability of a computer such as the IBM 7090 or 7044 is 32,000 words. The integration of several hundred first order equations is a routine affair.

On the other hand, the solution of a linear algebraic system is not a routine matter, computationally speaking. While formulas exist for the numerical inversion of a matrix, the solution may be inaccurate. The matrix may be ill-conditioned, and other techniques may have to be brought into play to remedy the situation [20]. The storage of the $n^{th}$ approximation for the calculation of the $(n + 1)^{st}$ approximation may become a problem; a suggestion for overcoming this difficulty is given in [21].

4. SOLUTION OF THE POTENTIAL PROBLEM

We follow the method of quasilinearization to identify the unknown mass and to solve the problem of potential determination of Section 2. The nonlinear system of equations is

$$\begin{align*}
\ddot{x} &= -\frac{x}{r^3} - \alpha \frac{x - z}{s^3} \\
\ddot{y} &= -\frac{y}{r^3} - \alpha \frac{y - \eta}{s^3} \\
\dot{a} &= 0
\end{align*}$$

(1)
with

\[ r^2 = x^2 + y^2 \quad s^2 = (x - \xi)^2 + (y - \eta)^2 \]

System (1) is equivalent to a system of five first order equations for \( x, \dot{x}, y, \dot{y}, \) and \( a. \) The system of linear equations for the \((k+1)^{\text{st}}\) stage is

\[
\begin{align*}
\dot{x}^{k+1} &= \left\{ -\frac{x^k}{r^3} - \frac{a^k x^k \xi}{s^3} \right\} \\
&\quad + (x^{k+1} - x^k) \left\{ -\frac{1}{r^3} + \frac{3x^{k^2}}{r^5} - \frac{a^k}{s^3} + \frac{3a^k(x^k \xi)^2}{s^5} \right\} \\
&\quad + (y^{k+1} - y^k) \left\{ \frac{3x^k y^k}{r^5} + \frac{3a^k(x^k \xi)(y^k \eta)}{s^5} \right\} \\
&\quad + (a^{k+1} - a^k) \left\{ -\frac{x^k \xi}{s^3} \right\} , \tag{3}
\end{align*}
\]

\[
\begin{align*}
\dot{y}^{k+1} &= \left\{ -\frac{y^k}{r^3} - \frac{a^k y^k \eta}{s^3} \right\} \\
&\quad + (x^{k+1} - x^k) \left\{ \frac{3x^k y^k}{r^5} + \frac{3a^k(x^k \xi)(y^k \eta)}{s^5} \right\} \\
&\quad + (y^{k+1} - y^k) \left\{ -\frac{1}{r^3} + \frac{3y^{k^2}}{r^5} - \frac{a^k}{s^3} + \frac{3a^k(y^k \eta)^2}{s^5} \right\} \\
&\quad + (a^{k+1} - a^k) \left\{ -\frac{y^k \eta}{s^3} \right\} , \\
\end{align*}
\]

\[
\begin{align*}
\dot{a}^{k+1} &= 0, \tag{4}
\end{align*}
\]

where

\[
\begin{align*}
(4) \quad r^2 &= (x^k)^2 + (y^k)^2 \quad s^2 = (x^k - \xi)^2 + (y^k - \eta)^2 .
\end{align*}
\]
We express the solution of (3) as the sum of a particular solution of (3) plus a linear combination of five independent solutions of the homogeneous form of (3).

\begin{align*}
x^{k+1}(t) &= p_x(t) + \sum_{j=1}^{5} c_j h_x^j(t), \\
y^{k+1}(t) &= p_y(t) + \sum_{j=1}^{5} c_j h_y^j(t), \\
\alpha^{k+1}(t) &= p_\alpha(t) + \sum_{j=1}^{5} c_j h_\alpha^j(t).
\end{align*}

(5)

Here, the symbol \( p_x(t) \) is meant to represent the \( x \) component of the particular solution, which is a vector of dimension five, and similarly for the symbols \( p_y(t), p_\alpha(t) \). The symbols \( h_x^j(t), h_y^j(t), h_\alpha^j(t) \) respectively correspond to the \( x, y, \) and \( \alpha \) components of the \( j \)th homogeneous solution vector, for \( j = 1, 2, \ldots, 5 \). The system which the particular solution satisfies is

\begin{align*}
\ddot{p}_x &= \left\{- \frac{x^k}{r^3} - \frac{\alpha^k x^k \eta}{s^3}\right\} \\
&\quad + (p_x - x^k) \left\{- \frac{1}{r^3} + \frac{3x^k}{5} - \frac{\alpha^k}{s^3} + \frac{3\alpha^k (x^k \eta)^2}{s^5}\right\} \\
&\quad + (p_y - y^k) \left\{\frac{3x^k y^k}{r^5} + \frac{3\alpha^k (x^k \eta)(y^k \eta)}{s^5}\right\} \\
&\quad + (p_\alpha - \alpha^k) \left\{- \frac{x^k \eta}{s^3}\right\}.
\end{align*}

(6)

\begin{align*}
\ddot{p}_y &= \left\{- \frac{y^k}{r^3} - \frac{\alpha^k y^k \eta}{s^3}\right\} \\
&\quad + (p_x - x^k) \left\{\frac{3x^k y^k}{r^5} + \frac{3\alpha^k (x^k \eta)(y^k \eta)}{s^5}\right\}.
\end{align*}
\[ + (p_y y^k) \left\{ - \frac{1}{r^3} + \frac{3 y^k k^2}{r^5} - \frac{\alpha k}{s^3} + \frac{3 r (y^k \eta)^2}{s^5} \right\} \]

\[ + (p_\alpha \alpha^k) \left\{ - \frac{y^k \eta}{s^3} \right\} , \]

\[ \dot{p}_\alpha = 0 . \]

with the initial condition

\[ p(0) = 0 . \]

The \( j \)th homogeneous solution satisfies the system

\[ \dot{h}_x^j = h_x \left\{ - \frac{1}{r^3} + \frac{3 y^k k^2}{r^5} - \frac{\alpha k}{s^3} + \frac{3 \alpha k (x^k \eta)^2}{s^5} \right\} \]

\[ + h_y^j \left\{ \frac{3 x^k y^k}{r^5} + \frac{3 \alpha k (x^k \eta)(y^k \eta)}{s^5} \right\} \]

\[ + h_\alpha^j \left\{ - \frac{x^k \eta}{s^3} \right\} \]

\[ \dot{h}_y^j = h_x \left\{ \frac{3 x^k y^k}{r^5} + \frac{3 \alpha k (x^k \eta)(y^k \eta)}{s^5} \right\} \]

\[ + h_y^j \left\{ - \frac{1}{r^3} + \frac{3 y^k k^2}{r^5} - \frac{\alpha k}{s^3} + \frac{3 \alpha k (y^k \eta)^2}{s^5} \right\} \]

\[ + h_\alpha^j \left\{ - \frac{y^k \eta}{s^3} \right\} . \]

\[ \dot{h}_\alpha^j = 0 . \]

Its five initial conditions are presented in the appropriate column of Table 1.
The particular and homogeneous solutions are produced by numerical integration and are known at the discrete times \( t = 0, \Delta, 2\Delta, 3\Delta, \ldots, T \).

Let us find the system of linear algebraic equations which is to be solved in the \((k+1)^{st}\) stage. The boundary conditions may be expressed as

\[
y^{k+1}(t_i) + [1-x^{k+1}(t_i)] \tan \theta(t_i) = 0,
\]

where \( \theta(t_i) \) is the observed angular position of the heavenly body \( H \) at time \( t_i \). Using relations (5), we obtain the five equations

\[
\sum_{j=1}^{5} c^j [h^j_y(t_i) - h^j_x(t_i) \tan \theta(t_i)]
\]

\[
= - \tan \theta(t_i) - p_y(t_i) + p_x(t_i) \tan \theta(t_i),
\]
(10) \[ i = 1, 2, \ldots, 5. \]

for the five unknowns \( c_1, c_2, \ldots, c_5 \).

The solution of (10) immediately gives us our new set of orbital parameters and the mass of the unknown perturbing body \( P \),

\[
\begin{align*}
x^{k+1}(0) &= c_1, \\
\dot{x}^{k+1}(0) &= c_2, \\
y^{k+1}(0) &= c_3, \\
\dot{y}^{k+1}(0) &= c_4, \\
\alpha^{k+1}(0) &= c_5.
\end{align*}
\]

Since we need \( x^{k+1}(t), y^{k+1}(t), \) and \( \alpha^{k+1}(t) \), for stage \( k+2 \), we use relations (5) for the evaluation of these functions at \( t = 0, \Delta, 2\Delta, 3\Delta, \ldots, T \). The cycle is ready to begin once more, and it is repeated until a solution of the nonlinear problem is found, or for a fixed number of stages.

We begin a numerical experiment with the initial guess that at time \( t = 0 \), the body \( H \) is at location \( (3,0) \) with velocity coordinates \( \dot{x} = 0, \dot{y} = 1 \), and we believe that the mass of \( P \) is about 0.3. We integrate equations (1) with the initial values

\[
(12) \quad x(0) = 3, \dot{x}(0) = 0, y(0) = 0, \dot{y}(0) = 1, \alpha(0) = 0.3.
\]
from $t = 0$ to $t = 2.5$, using a grid size of $\Delta = 0.01$ and an Adams-Moulton integration formula. This generates the curve labelled "Initial Approximation" in Fig. 2. This is a very poor approximation to the true orbit. After two stages of the quasilinearization scheme, our approximation has improved so that the orbit is represented by the curve labelled "Approximation 2" in Fig. 2. In five iterations, we converge to the true curve, $h(x,y)$, and we have found the correct value of 0.2 for the mass of the perturbing body. The rate of convergence is indicated in Table 2.

\begin{table}
\centering
\begin{tabular}{cccccc}
\hline
Approx. & $x(0)$ & $\dot{x}(0)$ & $y(0)$ & $\dot{y}(0)$ & $\alpha$ \\
\hline
0 & 3.0 & 0.0 & 0.0 & 1.0 & 0.3 \\
1 & 3.18421 & 0.221272 & 0.0 & 1.06544 & 0.120164 \\
2 & 2.37728 & -0.061370 & 0.0 & 0.690767 & -0.259144 \\
3 & 2.11189 & -0.018545 & 0.0 & 0.555666 & -0.070333 \\
4 & 2.01974 & -0.003194 & 0.0 & 0.509813 & -0.141922 \\
5 & 2.00023 & 0.000013 & 0.0 & 0.500120 & 0.198208 \\
\hline
True & 2.0 & 0.0 & 0.0 & 0.5 & 0.2 \\
\end{tabular}
\end{table}

Suppose that we also wish to know the position
Fig. 2. Successive approximations of the orbit.
TABLE 3.
PREDICTED LOCATION OF H AT TIME 2.5

<table>
<thead>
<tr>
<th>Approx.</th>
<th>x(2.5)</th>
<th>y(2.5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3.38098</td>
<td>2.47759</td>
</tr>
<tr>
<td>1</td>
<td>1.93764</td>
<td>2.72562</td>
</tr>
<tr>
<td>2</td>
<td>1.48932</td>
<td>1.53066</td>
</tr>
<tr>
<td>3</td>
<td>1.77202</td>
<td>1.21823</td>
</tr>
<tr>
<td>4</td>
<td>1.34124</td>
<td>1.1251</td>
</tr>
<tr>
<td>5</td>
<td>1.33503</td>
<td>1.10598</td>
</tr>
<tr>
<td>True</td>
<td>1.33494</td>
<td>1.10571</td>
</tr>
</tbody>
</table>

of H at some "future" time \( t = 2.5 \). Our sequence of approximations of the predicted location is given in Table 3. The entire calculations require only 1–1/2 minutes on the IBM 7044 computer, using a FORTRAN IV source language. The FORTRAN programs which generate the data and which determine the orbit and mass are listed in Appendix A.

The time involved is mainly due to the evaluation of the derivatives of the functions. The Adams-Moulton fourth order method requires the derivative to be evaluated twice for each integration step forward [25].

In another trial, beginning with the approximation that the orbit is a point at the earth's center [19], we find another solution which satisfies all of the conditions. However, the mass turns out to be greater than one, an allowed solution. Repeating the experiment with more closely spaced observations, we converge to the true solution. The determination of the optimal set of observations is itself an interesting question.


CHAPTER TWO

INVERSE PROBLEMS IN RADIATIVE TRANSFER: LAYERED MEDIA

1. INTRODUCTION

Some inverse problems in radiative transfer are concerned with the estimation of the optical properties of an atmosphere based on measurements of diffusely reflected radiation. The location and the intensity of the source of radiation are known. We consider a plane-parallel medium which is composed of two layers. Our aim is to determine the optical thickness and the albedo of each layer, from knowledge of the input radiation and the diffusely reflected light.

First we discuss the concept of invariant imbedding, and we apply this technique to the derivation of the equation for the diffuse reflection function of an inhomogeneous slab with isotropic scattering. The inverse problem is stated in terms of the reflection function, and we formulate the problem as a nonlinear boundary value problem. We then
show how the formalism of quasilinearization can be used to solve this problem. We conduct several numerical experiments for the determination of optical thicknesses and albedos of the layers. Computational results are presented, and the FORTRAN computer programs which produced the results are given in Appendix B.

2. INVARIANT IMBEDDING

The traditional approach to wave and particle transport processes leads to linear functional equations with boundary conditions. While linearity enables eigenfunction expansions to be made, one finds great difficulty in solving the equation of transfer. The integration of ordinary differential equations with given initial conditions is done extremely efficiently by digital computers. This suggests that problems be formulated in just this way, with the physical situation as the guide. Invariant imbedding provides a flexible manner in which to relate properties of one process to those of neighboring processes. This also leads to the generalized semigroup concept [1].

In a particle process, one is led by invariant imbedding to differential–integral equations for reflection and transmission functions. By the use of quadrature formulas [2], one reduces the equations from integral–differential form to approximate systems of ordinary differential equations. The wave equation, on the other hand, may be reduced to a system of ordinary differential equations in at least two ways:
(1) assume the time factor of the form $e^{i\omega t}$ and the problem simplifies to the steady state situation, or (2) use Laplace transform methods. Both alternatives are discussed in later sections.

Invariant imbedding is a useful formalism, theoretically and computationally speaking. Principles of invariance were first introduced by Ambarzumian in 1943 [3] and developed by Chandrasekhar [4]. The invariance concept was further extended and generalized by Bellman and Kalaba [5-10]. The form in which invariant imbedding is applied in these chapters is indicated by this example. Suppose that a neutron multiplication process takes place in a rod of length $x$ [11]. The investigator wishes to know the reflected flux $r$ for an input of one particle per second. Rather than study the processes within the rod extending from 0 to $x$, which would be quite difficult, the experimenter would like to vary the length of the rod and see how the reflected flux changes. The rod length is made a variable of the problem, so that $r = r(x)$. The original situation is imbedded in a class of similar cases, for all lengths of rod, and one obtains directly the reflected flux for a rod of any length including the length under investigation. This flux is rather easily computed and it is physically meaningful [15,16].

3. THE DIFFUSE REFLECTION FUNCTION FOR AN INHOMOGENEOUS SLAB

Consider an inhomogeneous, plane-parallel, non-emitting and isotropically scattering atmosphere of finite optical thickness $\tau_1$. The optical properties depend only on $\tau$, the
optical distance from the lower boundary ($0 \leq \tau \leq \tau_1$). The physical situation is sketched in Fig. 1. Parallel rays of light of net flux $\pi$ per unit area normal to their direction of propagation are incident on the upper surface, $\tau = \tau_1$. The direction is characterized by the parameter $\mu_0$ ($0 < \mu_0 \leq 1$), which is the cosine of the angle measured from the downward normal to the surface. The bottom surface is a completely absorbing boundary, so that no light is reflected from it. This assumption is not essential to our discussion.

Fig. 1. Incident and reflected rays for an inhomogeneous slab of optical thickness $\tau_1$. 
The direction of the outgoing radiation is characterized by \( \mu \), the cosine of the polar angle measured from the outward normal to the top surface. This parameter is the direction cosine of the vector representing the ray of light. The azimuth angle has no significance due to the symmetry of the situation.

By "intensity" we shall mean the amount of energy which is transmitted through an element of area \( d\sigma \) normal to the direction of flow, in a truncated elementary cone \( d\psi \) in time \( dt \). See Fig. 2, as well as Kourganoff [12]. We restrict ourselves to the steady-state situation at a fixed frequency.

We define the diffuse reflection function as follows:

(1) \( r(\mu, \mu_0, \tau_1) \) is the intensity of the diffusely reflected light in the direction whose cosine is \( \mu \) with respect to the outward normal, due to incident uniform parallel rays of radiation of constant net flux \( \pi \) in the direction whose cosine is \( \mu_0 \) with respect to the inward normal, the slab having optical thickness \( \tau_1 \).
Fig. 2. The incident and reflected intensities.
We define a related function $\rho$,

\[
\rho(\mu, \mu_0, \tau_1) = \frac{\mu r(\mu, \mu_0, \tau_1)}{\mu_0 \pi},
\]

which is the energy of the diffusely reflected light in the direction $\mu$ passing through a unit of horizontal area per unit solid angle per unit time, due to incident radiation of unit energy per unit horizontal area per unit solid angle per unit time, in the direction $\mu_0$. We may also interpret $\rho$ as the probability that a particle will emerge from a unit of horizontal area at $\tau = \tau_1$, the top of a slab of thickness $\tau_1$, going in direction $\mu$, per unit solid angle per unit time, due to an input of one particle per unit horizontal area per unit solid angle per unit time in the direction $\mu_0$.

Consider now a slab of thickness $\tau_1 + \Delta$ formed by adding a thin slab of thickness $\Delta$ to the top of the slab of thickness $\tau_1$, as illustrated in Fig. 3. By imbedding the original problem in a class of problems of similar nature, we will derive an integro–differential equation for the diffusely reflected function.

The diffuse reflection function for a slab of thickness $\tau_1 + \Delta$ with an input of net flux $\pi$ is

\[
r(\mu, \mu_0, \tau_1 + \Delta) = \pi \rho(\mu, \mu_0, \tau_1 + \Delta) \mu_0 / \mu.
\]

Applying the method of invariant imbedding in its particle counting form to the probability of emergence of a particle from a slab, we obtain the equation
Fig. 3. An inhomogeneous slab of optical thickness $\tau_1 + \Delta$. 
\[ (3) \quad \rho(\mu, \mu_0, \tau_1 + \Delta) = \rho(\mu, \mu_0, \tau_1) - \Delta \left( \frac{1}{\mu_0} + \frac{1}{\mu} \right) \rho(\mu, \mu_0, \tau_1) + \frac{\Delta}{\mu_0} \frac{\lambda(\tau_1)}{4\pi} + 2\pi \int_0^1 \rho(\mu', \mu_0, \tau_1) d\mu' \frac{\Delta}{\mu} \frac{\lambda(\tau_1)}{4\pi} \]

\[ + 2\pi \int_0^1 \frac{\Delta}{\mu_0} \frac{\lambda(\tau_1)}{4\pi} d\mu'' \rho(\mu, \mu'', \tau_1) \]

\[ + 2\pi \int_0^1 \rho(\mu', \mu_0, \tau_1) d\mu' \frac{\Delta}{\mu} \frac{\lambda(\tau_1)}{4\pi} 2\pi \int_0^1 \rho(\mu, \mu'', \tau_1) d\mu''. \]

\[ + o(\Delta). \]

The first term on the right hand side is the probability that a particle emerges without any interaction in the thin slab. The unit of distance is such that \( x \) is the probability of an interaction in a path of length \( x \). Hence the second term represents the losses due to interactions of the incoming and outgoing particles whose path lengths in \( \Delta \) are \( \Delta/\mu_0 \) and \( \Delta/\mu \) respectively. The third term is the probability of an interaction and re-emission isotropically into the direction given by \( \mu \). The function \( \lambda(\tau_1) \) is the probability of re-emission, and is called the albedo for single scattering. The next term is the probability that the particle is diffusely reflected from the slab between \((0, \tau_1)\) into the direction \( \mu' \) and interacts in \( \Delta \) and is re-emitted into the direction of emergence \( \mu \). The next term is the probability that an incoming particle interacts in \( \Delta \), enters the slab \((0, \tau_1)\) and is diffusely scattered into the
emergent direction \( \mu \). The sixth term is the probability of diffuse reflection in \((0, \tau_1)\), then interaction and re-emission in \( \Delta \), and diffuse reflection from \((0, \tau_1)\) with outgoing direction \( \mu \). All other probabilities are proportional to \( \Delta^2 \) or higher powers of \( \Delta \) and are accounted for in the term \( o(\Delta) \).

Let the diffusely reflected intensity be given by a new function \( R \), by means of the relation

\[
R(\mu, \mu_0, \tau_1) = \frac{R(\mu, \mu_0, \tau_1)}{4\mu},
\]

where \( R \) is related to \( \rho \) by the formula

\[
\rho(\mu, \mu_0, \tau_1) = \frac{R(\mu, \mu_0, \tau_1)}{4\mu}. \tag{5}
\]

Then \( R \) satisfies the equation

\[
R(\mu, \mu_0, \tau_1 + \Delta) = R(\mu, \mu_0, \tau_1) - \Delta \left( \frac{1}{\mu_0} + \frac{1}{\mu_1} \right) R(\mu, \mu_0, \tau_1)
\]
\[
\quad + \Delta \lambda \int_0^{1/2} \int_0^{1/2} R(\mu', \mu_0, \tau_1) \frac{\mu''}{\mu'} + \Delta \int_0^{1/2} R(\mu, \mu'', \tau_1) \frac{\mu''}{\mu}
\]
\[
\quad + \frac{1}{4} \left\{ \int_0^{1/2} R(\mu', \mu_0, \tau_1) \frac{\mu''}{\mu'} \int_0^{1/2} R(\mu, \mu'', \tau_1) \frac{\mu''}{\mu} \right\} + o(\Delta). \tag{6}
\]

We expand the lefthand side of the equation in powers of \( \Delta \),

\[
R(\mu, \mu_0, \tau_1 + \Delta) \approx R(\mu, \mu_0, \tau_1) + \frac{\partial R(\mu, \mu_0, \tau_1)}{\partial \tau_1} \Delta + o(\Delta). \tag{7}
\]
By letting Δ=0, we arrive at the integro–differential equation

\[
\frac{\partial R(\mu, \mu_0, \tau_1)}{\partial \tau_1} + \left( \frac{1}{\mu_0} + \frac{1}{\mu} \right) R
\]

\[
= \lambda(\tau_1) \left[ 1 + \frac{1}{2} \int_0^1 R(\mu', \mu_0, \tau_1) \frac{d\mu'}{\mu} \right]
\]

\[
\cdot \left[ 1 + \frac{1}{2} \int_0^1 R(\mu, \mu'', \tau_1) \frac{d\mu''}{\mu''} \right].
\]

The initial condition is

\[
R(\mu, \mu_0, 0) = 0,
\]

because no light is diffusely reflected when the medium has zero thickness. It is readily seen that the function R is symmetric [4,13,14,17], i.e.,

\[
R(\mu, \mu_0, \tau_1) = R(\mu_0, \mu, \tau_1).
\]

Equation (8) for R is the same as Chandrasekhar's equation for his scattering function S, when the medium is homogeneous and isotropic.

4. GAUSSIAN QUADRATURE

The above integrals may be evaluated by the use of Gaussian quadrature [4,13,14]. Since the limits of our integrals are zero to one, we use the approximate relation
which is exact if \( f(x) \) is a polynomial of degree \( 2N - 1 \) or less. The numbers \( a_k \) are roots of the shifted Legendre function \( P_N^*(x) = P_N(1-2x) \) on the interval \((0,1)\), and the numbers \( w_k \) are the corresponding weights. For a more detailed discussion and for tables of roots and weights, see [13].

Replacing integrals by Gaussian sums, we have the following equation which is approximately true,

\[
\frac{\partial R(\mu, \nu_0, \tau_1)}{\partial \tau_1} + \left( \frac{1}{\mu_0} + \frac{1}{\mu} \right) R = \lambda(\tau_1) \left[ 1 + \frac{1}{2} \sum_{k=1}^{N} R(\mu, \nu_0, \tau_1) \frac{w_k}{\mu_k} \right] \left[ 1 + \frac{1}{2} \sum_{k=1}^{N} R(\mu, \nu_k, \tau_1) \frac{w_k}{\mu_k} \right].
\]

For \( N \sim 7 \), this is a fairly good approximation [14,15].

We consider only those incident and outgoing directions for which the cosines take on the values of the roots \( \nu_k \).

For \( N = 7 \), the roots \( \nu_k \) and the corresponding angles, are cosine \( \mu_k \), are listed in Table 1, in order of increasing \( \mu \).
TABLE 1
ROOTS OF SHIFTED LEGENDRE POLYNOMIALS OF
DEGREE N = 7, AND CORRESPONDING ANGLES

<table>
<thead>
<tr>
<th>k</th>
<th>Roots $u_k$</th>
<th>Arc cosine $u_k$ (in degrees)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.023446044</td>
<td>88.541891</td>
</tr>
<tr>
<td>2</td>
<td>0.12923441</td>
<td>82.574646</td>
</tr>
<tr>
<td>3</td>
<td>0.29707742</td>
<td>72.717849</td>
</tr>
<tr>
<td>4</td>
<td>0.50000000</td>
<td>60.000000</td>
</tr>
<tr>
<td>5</td>
<td>0.70292258</td>
<td>45.338044</td>
</tr>
<tr>
<td>6</td>
<td>0.87076559</td>
<td>29.452271</td>
</tr>
<tr>
<td>7</td>
<td>0.97455396</td>
<td>12.953079</td>
</tr>
</tbody>
</table>

We define the functions of one argument,

(3) $p_{ij}(\tau_1) = R(u_i, u_j, \tau_1),$

for $i = 1, 2, \ldots, N$, $j = 1, 2, \ldots, N$. Then (2) becomes a system of ordinary differential equations

(4) $\frac{dR_{ij}(\tau_1)}{d\tau_1} + \left(\frac{1}{u_i} + \frac{1}{u_j}\right)R_{ij} =$

$\lambda(\tau_1) \left[ 1 + \frac{1}{2} \sum_{k=1}^{N} R_{kj}(\tau_1) \frac{\omega_k}{u_k} \right] \left[ 1 + \frac{1}{2} \sum_{k=1}^{N} R_{ik}(\tau_1) \frac{\omega_k}{u_k} \right],$

with optical thickness $\tau_1$ as the independent variable. The initial conditions are, of course,

(5) $R_{ij}(0) = 0.$
The system of $N^2$ first order differential equations reduce to a system of $N(N+1)/2$ equations by the use of the symmetry property of $R$. This is a large saving of computational effort.

5. AN INVERSE PROBLEM

Consider the inhomogeneous medium composed of two layers as illustrated in Fig. 4.

![Fig. 4. A layered medium](image)

The total thickness of the medium is 1.0, the thickness of each slab is 0.5, and the albedos are 0.4 in the lower layer, 0.6 in the upper layer. In order to have a continuous function for the albedo, we assume that $\lambda$ is given by the function

\[
\lambda(\tau) = 0.5 + 0.1 \tanh 10(\tau - 0.5).
\]
This function is plotted in Fig. 5.

Fig. 5. The albedo function \( \lambda(\tau) = 0.5 + 0.1 \tanh 10(\tau - 0.5) \) for a slab of thickness 1.0.

Parallel rays of net flux \( \pi \) are incident on the upper surface of the medium in a direction characterized by the parameter \( u_j \). We obtain \( N^2 \) measurements of the intensity of the diffusely reflected light, \( b_{ij} \approx r_{ij}(\tau_1) \), for incident directions \( u_j, j = 1,2,...,N \), and reflection directions \( u_i, i = 1,2,...,N \). We wish to determine the nature of the medium from the knowledge of the reflected radiation.

Let the total optical thickness of the slab be \( T \), and let the thickness of the lower layer be \( c \). Let the albedos be \( \lambda_1 \) and \( \lambda_2 \), for the lower and upper slabs respectively, where the albedo as a function of optical elevation is
\[(2) \quad \lambda(\tau) = a + b \tanh 10(\tau-c)\]

and \[\lambda_{1} = a-b,\]

\[(3) \quad \lambda_{2} = a+b,\]

where a and b are unknown parameters. For the "true" situation,

\[(4) \quad T = 1.0, a = 0.5, b = 0.1, c = 0.5.\]

The inverse problem which we wish to solve is to determine the quantities T, a, b, and c in such a way as to have best agreement, in the least square sense, between the estimated solution using the ordinary differential equations for \(r_{ij}\) and the observed reflection pattern. Mathematically speaking, we wish to minimize the expression

\[(5) \quad \sum_{i=1}^{N} \sum_{j=1}^{N} [r_{ij}(T) - b_{ij}]^2\]

over all choices of the unknown parameters.

In Table 2, we present the measurements \([b_{ij}]\) for \(N = 7\). In Fig. 6 we plot some of the measurements as a function of the cosine of the reflection angle, \(\mu \sim \mu_{1}\), for input directions \(\mu_0 \sim \mu_j \approx .025, .5,\) and .975. The discrete observations are shown as dots, and for clarity we draw smooth curves through these points. For comparison, we show what the corresponding measurements would be if the medium were homogeneous with albedo \(\lambda = 0.5\).
TABLE 2

THE MEASUREMENTS $[b_{ij}]$

<table>
<thead>
<tr>
<th></th>
<th>i = 1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>j = 1</td>
<td>0.079914</td>
<td>0.028164</td>
<td>0.014304</td>
<td>0.009104</td>
<td>0.006707</td>
<td>0.005515</td>
<td>0.004970</td>
</tr>
<tr>
<td>2</td>
<td>0.143038</td>
<td>0.091522</td>
<td>0.058437</td>
<td>0.040826</td>
<td>0.031405</td>
<td>0.026378</td>
<td>0.023989</td>
</tr>
<tr>
<td>3</td>
<td>0.167000</td>
<td>0.134331</td>
<td>0.099653</td>
<td>0.075106</td>
<td>0.060044</td>
<td>0.051445</td>
<td>0.047248</td>
</tr>
<tr>
<td>4</td>
<td>0.178898</td>
<td>0.157955</td>
<td>0.126408</td>
<td>0.099392</td>
<td>0.081253</td>
<td>0.070435</td>
<td>0.065042</td>
</tr>
<tr>
<td>5</td>
<td>0.185284</td>
<td>0.170817</td>
<td>0.142072</td>
<td>0.114229</td>
<td>0.094495</td>
<td>0.082423</td>
<td>0.076332</td>
</tr>
<tr>
<td>6</td>
<td>0.188723</td>
<td>0.177733</td>
<td>0.150791</td>
<td>0.122665</td>
<td>0.102104</td>
<td>0.089349</td>
<td>0.082870</td>
</tr>
<tr>
<td>7</td>
<td>0.190354</td>
<td>0.180898</td>
<td>0.154995</td>
<td>0.126773</td>
<td>0.105829</td>
<td>0.092748</td>
<td>0.086083</td>
</tr>
</tbody>
</table>
Fig. 6. Some of the measurements $b_{ij}$ for a layered medium.
6. FORMULATION AS A NONLINEAR BOUNDARY VALUE PROBLEM

We formulate this inverse problem as a nonlinear boundary value problem. To the system of \(N^2\) nonlinear differential equations

\[
\frac{dR_{ij}}{d\tau} + \left( \frac{1}{\mu_i} + \frac{1}{\mu_j} \right) R_{ij} = \lambda(\tau_1) \left[ 1 + \frac{1}{2} \sum_{k=1}^{N} R_{kj} \frac{w_k}{\mu_k} \right] \left[ 1 + \frac{1}{2} \sum_{k=1}^{N} R_{ik} \frac{w_k}{\mu_k} \right],
\]

where

\[
\lambda(\tau_1) = a + b \tan 10(\tau_1 - c),
\]

we add the differential equations

\[
\frac{da}{d\tau_1} = 0, \quad \frac{db}{d\tau_1} = 0, \quad \frac{dc}{d\tau_1} = 0, \quad \frac{dT}{d\tau_1} = 0
\]

because \(a, b,\) and \(c\) and \(T\) are unknown constants. The boundary conditions are

\[
R_{ij}(0) = 0,
\]

and

\[
\frac{\partial S}{\partial a} = 0, \quad \frac{\partial S}{\partial b} = 0, \quad \frac{\partial S}{\partial c} = 0, \quad \frac{\partial S}{\partial T} = 0,
\]

where

\[
S = \sum_{i=1}^{N} \sum_{j=1}^{N} \left( R_{ij}(T) - 4\mu_i b_{ij} \right)^2.
\]
7. **NUMERICAL EXPERIMENTS I. DETERMINATION OF c, THE THICKNESS OF THE LOWER LAYER.**

Let us try to determine the quantity c, which is the thickness of the lower layer of the stratified medium. We assume that all of the other parameters a, b, and T are known. The parameter c is considered to be a function of optical height $\tau_1$ described by the equation $dc/d\tau_1 = 0$.

By following the method of quasilinearization as described previously, we obtain a system of linear differential equations for the $(k+1)^{st}$ approximation of $R_{ij}$ and c:

\[
\frac{dR_{ij}^{k+1}}{d\tau_1} = f(R_{ij}^k, c^k) + \sum_{i,j} (R_{ij}^{k+1} - R_{ij}^k) \frac{\partial f}{\partial R_{ij}^k} + (c^{k+1} - c^k) \frac{\partial f}{\partial c^k},
\]

\[
\frac{dc^{k+1}}{d\tau_1} = 0,
\]

where

\[
f(R_{ij}^k, c^k) = -\left(\frac{1}{\mu_1} + \frac{1}{\mu_j}\right)R_{ij}^k + \lambda(c^k)(1 + \frac{1}{2} \sum_{t=1}^N R_{it}^k \frac{w_t}{\mu_t}) \cdot (1 + \frac{1}{2} \sum_{t=1}^N R_{it}^k \frac{w_t}{\mu_t}),
\]

\[
\lambda(c^k) = a + b \tanh 10(\tau_1 - c^k).
\]

After simplifying, we have
\[
\frac{dR_{ij}^{k+1}}{d\tau_1} = \left\{ -\left( \frac{1}{\mu_i} + \frac{1}{\mu_j} \right) R_{ij}^k + \lambda(c^k)(1 + \frac{1}{Z} \sum_{t=1}^{N} \frac{R_{it}^k w_t}{\mu_t}) \times \right. \\
\left. \left( 1 + \frac{1}{Z} \sum_{t=1}^{N} R_{tj}^k \frac{w_t}{\mu_t} \right) \right\} + \left\{ -\left( \frac{1}{\mu_i} + \frac{1}{\mu_j} \right)(R_{ij}^{k+1} - R_{ij}^k) + \right. \\
\frac{1}{2} \lambda(c^k) \left[ (1 + \frac{1}{Z} \sum_{t=1}^{N} R_{it}^k \frac{w_t}{\mu_t}) \times \sum_{t=1}^{N} (R_{tj}^{k+1} - R_{tj}^k) \frac{w_t}{\mu_t} \right] + \\
\left. \left( 1 + \frac{1}{Z} \sum_{t=1}^{N} R_{tj}^k \frac{w_t}{\mu_t} \right) \times \sum_{t=1}^{N} (R_{ti}^{k+1} - R_{ti}^k) \frac{w_t}{\mu_t} \right\} + \\
\left\{ \lambda(c^{k+1} - c^k)(1 + \frac{1}{Z} \sum_{t=1}^{N} R_{it}^k \frac{w_t}{\mu_t}) \times \left( 1 + \frac{1}{Z} R_{tj}^k \frac{w_t}{\mu_t} \right) \times \right. \\
\left. (-10 \text{ sech}^2(10(\tau_1 - c^k))) \right\}, \\
\]

\[
\frac{dc^{k+1}}{d\tau_1} = 0.
\]

Since \( N = 7 \), there are basically \( 7^2 + 1 = 50 \) differential equations, which reduce to \( 7 \cdot 8/2 + 1 = 29 \) differential equations by the use of the symmetry property

\[(5) \quad R_{ij}^{k+1}(\tau_1) = R_{ji}^{k+1}(\tau_1).\]

While the computations are reduced, the full set of values \( R_{ij}^{k+1} \) representing a \( 7 \times 7 \) matrix is always available.

Now let the 50-dimensional vector \( x^{k+1}(\tau_1) \) have the components
(6) \( x^{k+1}_{i}(\tau_1) = R_{ij}^{k+1}(\tau_1) \),

for \( t = 1,2,\ldots,49 \) as \( i = 1,2,\ldots,7 \) and \( j = 1,2,\ldots,7 \), and

(7) \( x^{k+1}_{50}(\tau_1) = c^{k+1}(\tau_1) \).

Since \( x^{k+1}(\tau_1) \) is a solution of a system of linear differential equations, we may represent it as the sum of a particular vector solution, \( p(\tau_1) \), and a vector solution of the homogeneous system, \( h(\tau_1) \),

(8) \( x^{k+1}(\tau_1) = p(\tau_1) + m h(\tau_1) \).

The system of differential equations for \( p(\tau_1) \) is obtained by substituting the appropriate component of \( p \) where ever \( R_{ij}^{k+1} \) or \( c^{k+1} \) occurs in (4). We choose the initial conditions \( p(0) = 0 \). The system of equations for the homogeneous solution is similarly obtained, but of course all terms not involving the \( (k+1)^{st} \) approximation are dropped. The initial vector \( h(0) \) has all of its components zero except for the last, which is unity. The boundary conditions \( R_{ij}^{k+1}(0) = 0 \) are identically satisfied. The solutions \( p(\tau_1) \) and \( h(\tau_1) \) are produced on the interval \( 0 \leq \tau_1 \leq 1.0 \) by numerical integration.

The multiplier \( m \) is chosen to minimize the quadratic form,
where the observations are \( b_t \approx x_t^{k+1}(1) \). It is required that
\[
\frac{\delta S}{\delta m} = 0,
\]
and so the value of \( m \) is
\[
m = \frac{\sum_{t=1}^{49} h_t(1)[b_t - p_t(1)]^2}{\sum_{t=1}^{49} [h_t(1)]^2}.
\]

The thickness of the lower layer in the new approximation is
\[
c^{k+1} = m.
\]

The initial approximation required for this successive approximation scheme is produced by numerically integrating the nonlinear system of equations for \( R \) using a rough estimate of \( c \). The results of three experiments with initial guesses \( c = 0.2, 0.8, \) and 0.0 respectively are given in Table 3. The values of \( c \) obtained in the first, second, third and fourth approximations are tabulated.

**TABLE 3**

**SUCCESSIVE APPROXIMATIONS OF \( c \), THE LEVEL OF THE INTERFACE**

<table>
<thead>
<tr>
<th>Approximation</th>
<th>Run 1</th>
<th>Run 2</th>
<th>Run 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.2</td>
<td>0.8</td>
<td>0.0</td>
</tr>
<tr>
<td>1</td>
<td>0.62</td>
<td>0.57</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.5187</td>
<td>0.5024</td>
<td>No convergence</td>
</tr>
<tr>
<td>3</td>
<td>0.500089</td>
<td>0.499970</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.499990</td>
<td>0.499991</td>
<td></td>
</tr>
<tr>
<td>True Value</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>
The initial guess of $c$ in Run 1 is 60\% too low, and in Run 2, 60\% too high. Yet the correct value of $c$ is accurately found in 3 to 4 iterations. The time required for each run is about 2 minutes on the IBM 7044 digital computer, using an Adams-Moulton fourth order integration scheme with a grid size of $\Delta \tau_1 = 0.01$. Each iteration requires the integration of $2 \times 29 = 58$ differential equations with initial values, and the values of $p_\ell(\tau_1)$ and $h_\ell(\tau_1)$ thus produced are stored in the rapid access memory of the computer at each of a hundred and one grid points, $\tau_1 = 0, .01, .02, \ldots, 1.0$. The current approximation of $R_{ij}^k$ is also stored at a hundred and one points.

Run 3 is an unsuccessful experiment because the initial guess for $c$, i.e., a single layer approximation, is very poor. The solution diverges.

8. NUMERICAL EXPERIMENTS II. DETERMINATION OF $T$, THE OVERALL OPTICAL THICKNESS

Now let us try to estimate the total optical thickness $T$ of the stratified medium, assuming that we know all of the other parameters of the system. Again we are provided with 49 measurements of $\{b\}$, the intensity of the diffusely reflected radiation in various directions.

The quantity $T$ is the end point of the range of integration, i.e., $0 \leq \tau_1 \leq T$. In order to have a known end point, we define a new independent variable $\sigma$,

$$\sigma T = \tau_1,$$
so that the integration interval is fixed, \( 0 \leq \sigma \leq 1 \). Then \( T \) satisfies the equation, \( dT/d\sigma = 0 \). Our system of non-linear equations is

\[
\frac{dR_{ij}(\sigma)}{d\sigma} = T \left\{ -\left( \frac{1}{\mu_i} + \frac{1}{\mu_j} \right) R_{ij} \right. \\
+ \lambda \left[ 1 + \frac{1}{2} \sum_{k=1}^{N} R_{ij} \frac{w_k}{\mu_k} \right] \left[ 1 + \frac{1}{2} \sum_{k=1}^{N} R_{kj} \frac{w_k}{\mu_k} \right],
\]

\[
\frac{dT}{d\sigma} = 0,
\]

where \( \lambda = a + b \tanh 10(\sigma T - c) \).

The solution is subject to the conditions

\[
R_{ij}(0) = 0,
\]

\[
\min_{T} \sum_{i=1}^{N} \sum_{j=1}^{N} \left[ R_{ij}(T) - 4\mu_i b_{ij} \right]^2.
\]

Linear differential equations are obtained in the same manner as before, and we solve a sequence of linear boundary value problems.

Three trials are made to determine the thickness \( T \), with initial guesses \( T = 0.9, 1.5, \) and \( 0.5 \), while the correct value is \( 1.0 \). Four iterations yield a value of \( T \) which is correct to one part in a hundred thousand, in each of the three experiments. The total computing time is four minutes. The experiment is successful even when the initial guess is only one-half of the true value.

Given 49 measurements of the diffusely reflected light, we wish to determine the two albedos

\[ \lambda_1 \approx a - b, \quad \lambda_2 \approx a + b, \]

and the thicknesses of the two layers. We assume that we know the overall thickness \( T = 1.0 \), and so if the thickness of the lower layer is \( c \), the thickness of the upper layer is given by \( T - c \). The unknown parameters are \( a \), \( b \), and \( c \). Since there are three unknowns, we have three homogeneous solutions and of course a particular solution to compute in each iteration of the experiment. Each solution has \( 28 + 3 = 31 \) components, so that there are \( 4 \times 31 = 124 \) linear differential equations being integrated during each stage of the quasilinearization scheme. The three multipliers form the solution of a third order linear algebraic system. They are found by a matrix inversion using a Gaussian elimination method. Table 4 summarizes the results of an experiment which is carried out in about 2 minutes on the IBM 7044. The FORTRAN IV computer programs for all three series of experiments are given in Appendix B.
### TABLE 4
SUCCESSIVE APPROXIMATIONS OF $\lambda_1$, $\lambda_2$, AND $c$

<table>
<thead>
<tr>
<th>Approximation</th>
<th>$\lambda_1 = a-b$</th>
<th>$\lambda_2 = a+b$</th>
<th>$c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.51</td>
<td>0.69</td>
<td>0.4</td>
</tr>
<tr>
<td>1</td>
<td>0.4200</td>
<td>0.6052</td>
<td>0.5038</td>
</tr>
<tr>
<td>2</td>
<td>0.399929</td>
<td>0.599995</td>
<td>0.499602</td>
</tr>
<tr>
<td>3</td>
<td>0.399938</td>
<td>0.599994</td>
<td>0.499878</td>
</tr>
</tbody>
</table>

True Value 0.4 0.6 0.5

10. DISCUSSION

The approach which is discussed above is readily extended to other inverse problems with different physical situations. The numerical experiments in this chapter make use of many accurate observations of the reflected light while in the next chapter, the effect of errors in the measurements is examined. We note that initial approximations must be good enough to insure convergence. A rational initial estimate may be made from knowledge of the diffuse reflection fields for various homogeneous slabs, as calculated for example by Bellman, Kalaba and Prestrud [14]. Other inverse problems might deal with the transmission function, the source function, the $X$ and $Y$ functions, and the emergence probabilities [18–22].
REFERENCES


CHAPTER THREE

INVERSE PROBLEMS IN RADIATIVE TRANSFER:
NOISY OBSERVATIONS

1. INTRODUCTION

The techniques of invariant imbedding and quasilinearization are applied to some inverse problems of radiative transfer through an inhomogeneous slab in which the albedo for single scattering has a parabolic dependence on optical height. The results of many numerical experiments on the effect of the angle of incidence of radiation, errors in observations, and minimax versus least squares criterion are reported. Other experiments are carried out to design an optical medium according to specified requirements. The knowledge gained through this type of numerical experimentation should prove useful in the planning of laboratory or satellite experiments as well as for the reduction of data and the construction of model atmospheres.

2. AN INVERSE PROBLEM

Consider an inhomogeneous, plane parallel, non-emitting and isotropically scattering atmosphere of finite optical thickness $\tau$. Its optical properties depend only on the optical distance $\tau$ from the bottom surface. The bottom surface is a completely absorbing boundary, so that no light
is reflected from it. See Fig. 1 for a sketch of the physical situation. Parallel rays of light of net flux \( \pi \) per unit area normal to their direction of propagation are incident on the upper surface. The direction is specified by \( \mu_0 \) \((0 < \mu_0 \leq 1)\), the cosine of the angle measured from the normal to the surface [1,2].

![Fig. 1. The physical situation](image)

Let \( r(\mu, \mu_0, \tau_1) \) denote the intensity of the diffusely reflected light in the direction \( \mu \) and set \( R(\mu, \mu_0, \tau_1) = 4 \pi r \). Then the function \( R \) satisfies the integro-differential equation

\[
\frac{\partial R}{\partial \tau_1} = - \left( \frac{1}{\mu} + \frac{1}{\mu_0} \right) R + \lambda(\tau_1) \left\{ 1 + \frac{1}{2} \int_0^\tau R(\mu', \mu_0, \tau_1) \frac{d\mu'}{\mu'} \right\}
\]

(1)

with initial condition
The function $\lambda(\tau_1)$ is the albedo for single scattering.

We wish to consider the inverse problem of estimating the optical properties of the medium as represented by $\lambda(\tau)$ as well as the optical thickness of the slab, based on measurements of the diffusely reflected light.

3. FORMULATION AS A NONLINEAR BOUNDARY VALUE PROBLEM

Let us consider the case in which the albedo may be assumed to have a parabolic form,

\[ \lambda(\tau) = \frac{1}{2} + a\tau + b\tau^2, \]

where $a$ and $b$ are constants for a particular slab. For example, let us take $a = 2$ and $b = -2$, and we choose the optical thickness $\tau = 1.0$. The albedo as a function of optical height is shown in Fig. 2.

![Fig. 2. A parabolic albedo function, $\lambda(\tau) = \frac{1}{2} + 2\tau - 2\tau^2$, for a slab of thickness 1.0](image-url)
We replace the integro-differential equation by the discrete approximate system obtained by the use of Gaussian quadrature,

\[
\frac{dR_{ij}}{d\tau_1} = - \left( \frac{1}{u_i} + \frac{1}{u_j} \right) R_{ij} + \lambda(\tau_1) \left\{ 1 + \frac{1}{2} \sum_{k=1}^{N} R_{ik}(\tau_1) \frac{W_k}{u_k} \right\} \cdot \left\{ 1 + \frac{1}{2} \sum_{k=1}^{N} R_{kj}(\tau_1) \frac{W_k}{u_k} \right\}.
\]

In these equations, \( R_{ij}(\tau_1) \) represents \( R(u_i, u_j, \tau_1) \).

We produce "observations" of the diffusely reflected light by choosing \( N = 7 \), and integrating (2) from \( \tau_1 = 0 \) to \( \tau_1 = 1.0 \), and then setting \( b_{ij} = \frac{R_{ij}}{4u_i} \). Then \( \{b_{ij}\} \) is the set of measurements for \( \tau_1 = 1 \).

Starting with the observations \( \{b_{ij}\} \simeq \{r_{ij}(c)\} \), we wish to determine the quantities \( a, b, \) and the optical thickness \( c \) which minimize the expression

\[
S = \sum_{i,j} \{r_{ij}(c) - b_{ij}\}^2,
\]

where \( r_{ij}(\tau_1) = 4u_i r_{ij}(\tau_1) \) is the solution of the nonlinear system (2). This inverse problem may be viewed as a non-linear boundary-value problem.

4. SOLUTION VIA QUASILINEARIZATION

Since the terminal value of the independent variable \( \tau_1 \) is unknown, we make the following transformation to a new independent variable \( \sigma \).
(1) \( \sigma = \tau_1/c \),

which has initial value 0 and terminal value 1.0. Then the parameters \( a, b, \) and the thickness \( c \) satisfy the equations

(2) \[
\frac{da}{d\sigma} = 0, \quad \frac{db}{d\sigma} = 0, \quad \frac{dc}{d\sigma} = 0.
\]

Eqs. (2) are added to the system

\[
\frac{dR_{ij}}{d\sigma} = \left\{ \frac{1}{u_i} + \frac{1}{u_j} \right\} R_{ij} + \lambda(\sigma) \left[ 1 + \frac{1}{2} \sum_{k=1}^{N} R_{ik} \frac{W_k}{u_k} \right]
\]

(3) \[
\cdot \left[ 1 + \frac{1}{2} \sum_{k=1}^{N} R_{kj} \frac{W_k}{u_k} \right]
\]

where

(4) \[
\lambda(\sigma) = \frac{1}{2} + ac\sigma + bc^2\sigma^2.
\]

The application of the technique of quasilinearization [2] yields the linear system for the \((n+1)^{st}\) approximation,

\[
\frac{dR_{ij}^{n+1}}{d\sigma} = c^n \left\{ \frac{1}{u_i} + \frac{1}{u_j} \right\} R_{ij}^{n} + \lambda(a^n, b^n, c^n, \sigma) f_i(R^n) f_j(R^n)
\]

\[
+ c^n \left\{ - \left( \frac{1}{u_i} + \frac{1}{u_j} \right) (R_{ij}^{n+1} - R_{ij}^{n})
\]

\[
+ \frac{1}{2} \lambda(a^n, b^n, c^n, \sigma) \left[ f_i \sum_{k=1}^{N} (R_{kj}^{n+1} - R_{kj}^{n}) \frac{W_k}{u_k}
\]

\[
+ f_j \sum_{t=1}^{N} (R_{it}^{n+1} - R_{it}^{n}) \frac{W_t}{u_t} \right\}
\]
\[ + (a^{n+1} - a^n) \left( c^n (c^n) \right) f_i (R^n) f_j (R^n) \]
\[ + (b^{n+1} - b^n) \left( c^n (c^n)^2 \right) f_i (R^n) f_j (R^n) \]
\[ + (c^{n+1} - c^n) \left\{ \left( \frac{1}{m_i} + \frac{1}{m_j} \right) R_{ij}^n + \lambda(a^n, b^n, c^n, \sigma) f_i (R^n) f_j (R^n) \right\} + [a^n c^n + 25^n (c^n)^2] f_i (R^n) f_j (R^n) \]

\[ \frac{d a_{n+1}}{d \sigma} = 0 , \]
\[ \frac{d b_{n+1}}{d \sigma} = 0 , \]
\[ \frac{d c_{n+1}}{d \sigma} = 0 , \]

where
\[ \lambda(a^n, b^n, c^n, \sigma) = \frac{1}{2} + a^n (c^n) + b^n (c^n)^2 , \]
\[ f_i (R^n) = 1 + \frac{1}{2} \sum_{j=1}^{N} R_{ij}^n \frac{w_j}{u_j} . \]

The solution of Eqs. (5) may be represented in the form

\[ R_{ij}^n (\sigma) = p_{ij} (\sigma) + \frac{3}{k=1} c^k \left( h_{ij} (\sigma) \right) , \]
\[ a^n (\sigma) = q_1 (\sigma) + \frac{3}{k=1} c^k w_1 (\sigma) , \]
\[ b^n (\sigma) = q_2 (\sigma) + \frac{3}{k=1} c^k w_2 (\sigma) , \]
\[ c^n (\sigma) = q_3 (\sigma) + \frac{3}{k=1} c^k w_3 (\sigma) . \]
where the vector \( P \), constituted of elements \( p_{ij}(\sigma) \) and \( q_{i}(\sigma) \), is a particular solution of (5), and the vectors \( H^{k} \) composed of elements \( h_{ij}^{k}(\sigma) \) and \( w_{i}^{k}(\sigma) \), are three independent solutions of the homogeneous form of (5), for \( k = 1, 2, 3 \). We choose the initial conditions \( P(0) \) identically zero, and \( H^{k}(0) \) having all of its elements zero except for that component which corresponds to \( w^{k}, k = 1, 2, 3 \). The choice of initial conditions allows us to identify the multipliers \( c^{k} \) (not to be confused as powers of \( c \)) as

\[
\begin{align*}
a &= a(0) = c^{1}, \\
b &= b(0) = c^{2}, \\
c &= c(0) = c^{3}.
\end{align*}
\]

We seek the three missing initial values (7).

Let us make the conversion from measurements of \( r_{ij}(\sigma) \) to measurements of \( R_{ij}(1) \) by setting

\[
\beta_{ij} = 4u_{i}b_{ij}.
\]

Then we write the expression to be minimized as

\[
S = \sum_{i,j} \left( R_{ij}^{n+1}(1) - \beta_{ij} \right)^{2}.
\]

This expression is a minimum when the following requirements are met:

\[
\frac{\partial S}{\partial a} = 0, \quad \frac{\partial S}{\partial b} = 0, \quad \frac{\partial S}{\partial c} = 0.
\]
By means of (7), these conditions are equivalent to

\[ \frac{\partial S_i}{\partial c_1} = 0, \quad \frac{\partial S_i}{\partial c_2} = 0, \quad \frac{\partial S_i}{\partial c_3} = 0. \]  

We replace \( R_{ij}^{n+1}(1) \) in (9) by its representation (6). Then Eqs. (11) lead us to a third order system of linear algebraic equations of the form

\[ AX = B, \]

where the elements of the matrix \( A \) and the vector \( B \) are, respectively,

\[ A_{ij} = \sum_{m,n} h_{mn}^i(1) h_{mn}^j(1), \]

\[ B_i = \sum_{m,n} h_{mn}^i(1) [\beta_{mn} - p_{mn}(1)]. \]

and the solution vector \( X \) has as its components the multipliers \( c_1, c_2, c_3 \). In this way we obtain the current approximation to the parameters \( a \) and \( b \) in the albedo function, and the thickness of the slab. To begin the calculations, we produce an initial approximation by integrating the system of nonlinear differential equations (3) with \( R(0) = 0 \) and using estimated values of the parameters. Several iterations of the method are usually sufficient to attain convergence, if convergence takes place at all.
5. NUMERICAL EXPERIMENTS I: MANY ACCURATE OBSERVATIONS

Some of the observations \( \{ \overline{e}_{ij} \} \approx \{ R_{ij}(1) \} \) are plotted in Fig. 3.

Several types of numerical experiments are carried out. In the first class of experiments, 49 perfectly accurate (to about 8 decimal figures) observations are used to determine the quantities a, b, and c. The 49 observations correspond to measurements for 7 outgoing angles for each of 7 incident directions, as listed in Table 1, Chapter II. In one of the trials, the initial approximation is generated with the guesses \( a = 2.2 (+10\% \text{ in error}), \) \( b = -1.8 (+10\% \text{ in error}), \) and \( c = 1.5 (+50\% \text{ in error}). \) After four iterations, our estimates are decidedly better: \( a = 1.99895 (-.005\% \text{ in error}), \) \( b = -1.99824 (+.014\% \text{ in error}), \) and \( c = 1.004 (+.04\% \text{ in error}). \) We repeat the experiment, with one change: our initial estimate of the thickness is 0.5, only one-half of the correct value. This time the solution diverges and the procedure fails.

Fig. 4a illustrates the rapid rate of convergence to the correct solution for the albedo function \( \lambda(\tau) \), for the successful trial. The initial approximation is designated in the figure by the numeral 0, the first approximation by 1. The fourth approximation coincides with the true solution. Fig. 4b shows how the initial approximation to the function \( R_{ij}(c) \) for incident direction cosine 0.5 deviates from the observed values as indicated by the curve.
Fig. 3. Some of the observations \{\tau_{ij}\}
Fig. 4. (a) Successive approximations of the albedo function.
(b) Successive approximations of the function $R_{ij}(c)$. 
labelled "True". The first approximation lies very close to the correct values, and the fourth approximation is graphically identical with the correct solution.

6. NUMERICAL EXPERIMENTS II: EFFECT OF ANGLE OF INCIDENCE

In a second series of experiments, the incident angle is held fixed and accurate observations are made of the outgoing radiation in seven directions. The incident direction is varied from one trial to the next in order to study the effect of the position of the source. The initial approximation used in each trial is the same, the correct solution. Due to a possible lack of information in the observations for a given trial, the successive approximations may drift away from the correct solution and converge to another. Several iterations are carried out in each run. The results of the seven runs with each of seven angles are given in Table 1. The incident angle is given in degrees, and the fourth approximations to the constants \( a, b, \) and the thickness \( c \) are tabulated.
Table 1 indicates that the results are very good, no matter what the incident angle is. Examination of the computer output shows that convergence has occurred, in each trial, to about four significant figures. Angles 13° through 72.7° give nearly perfect values of the constants. Angles 82.6° and 88.5°, close to grazing incidence, give values which are only slightly poorer, 0.1% to 0.2% off.

7. NUMERICAL EXPERIMENTS III: EFFECT OF NOISY OBSERVATIONS

In this study, errors of different kinds and amounts are introduced into the observations, and the results of the determination of parameters are compared with the
results of Experiments I and II in which no errors were present. Errors are given in percentages with plus or minus signs. The errors in a given trial are either of equal magnitude or they occur in a Gaussian distribution. Let \( t_1, t_2, \ldots, t_7 \) be seven true measurements of \( R \). When we speak of noisy observations of \( \pm 5\% \) equal magnitude errors, we mean that the noisy observations are

\[
\begin{align*}
n_1 &= (1 + 0.05)t_1, \\
n_2 &= (1 - 0.05)t_2, \\
&\quad \cdots \\
n_7 &= (1 + 0.05)t_7.
\end{align*}
\]

(1)

Let \( g_1, g_2, \ldots, g_7 \) be seven (signed) Gaussian deviates, with standard deviation unity. Noisy observations with \( 5\% \) Gaussian distribution of errors are defined to be

\[
\begin{align*}
m_1 &= (1 + 0.05g_1)t_1, \\
m_2 &= (1 + 0.05g_2)t_2, \\
&\quad \cdots \\
m_7 &= (1 + 0.05g_7)t_7.
\end{align*}
\]

(2)

The results of numerical experiments with noisy observations, with one or seven angles of incidence, are presented in Table 2. Clearly, the accuracy of the estimation of the three constants is in proportion to the
TABLE 2.
NUMERICAL RESULTS WITH ERRORS IN OBSERVATIONS

<table>
<thead>
<tr>
<th>Incident Angle</th>
<th>± 1% Equal Mag. Error</th>
<th>± 2% Equal Mag. Error</th>
<th>± 5% Equal Mag. Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>a</td>
<td>b</td>
<td>c</td>
</tr>
<tr>
<td>88.5°</td>
<td>1.89</td>
<td>-1.80</td>
<td>1.05</td>
</tr>
<tr>
<td>82.6°</td>
<td>1.99</td>
<td>-1.96</td>
<td>1.013</td>
</tr>
<tr>
<td>72.7°</td>
<td>1.96</td>
<td>-1.93</td>
<td>1.016</td>
</tr>
<tr>
<td>60.0°</td>
<td>1.95</td>
<td>-1.91</td>
<td>1.016</td>
</tr>
<tr>
<td>45.3°</td>
<td>1.94</td>
<td>-1.90</td>
<td>1.016</td>
</tr>
<tr>
<td>29.5°</td>
<td>1.93</td>
<td>-1.89</td>
<td>1.016</td>
</tr>
<tr>
<td>13.0°</td>
<td>1.93</td>
<td>-1.89</td>
<td>1.016</td>
</tr>
<tr>
<td>All 7</td>
<td>1.99</td>
<td>-1.98</td>
<td>1.003</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Incident Angle</th>
<th>1% Gaussian Error</th>
<th>2% Gaussian Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>88.5°</td>
<td>1.46</td>
<td>-1.2</td>
</tr>
<tr>
<td>82.6°</td>
<td>1.64</td>
<td>-1.41</td>
</tr>
<tr>
<td>72.7°</td>
<td>1.71</td>
<td>-1.55</td>
</tr>
<tr>
<td>60.0°</td>
<td>1.75</td>
<td>-1.63</td>
</tr>
<tr>
<td>45.3°</td>
<td>1.77</td>
<td>-1.67</td>
</tr>
<tr>
<td>29.5°</td>
<td>1.78</td>
<td>-1.68</td>
</tr>
<tr>
<td>13.0°</td>
<td>1.79</td>
<td>-1.69</td>
</tr>
<tr>
<td>All 7</td>
<td>1.95</td>
<td>-1.93</td>
</tr>
</tbody>
</table>
accuracy of the observations. In contrast to the trials with perfect measurements, experiments using noisy observations are more successful when there is an abundance of data, and when the data are limited, these experiments show the effect of the incident direction. Errors with Gaussian errors give poorer results, which may be due to the particular set of 7 or 49 Gaussian deviates chosen arbitrarily from a book of random numbers [3].

8. NUMERICAL EXPERIMENTS IV: EFFECT OF CRITERION

This series of experiments is intended to investigate the effect of using a minimax criterion rather than a least squares condition for the determination of the unknown parameters $a$, $b$ and $c$. The condition requires that the constants be chosen to minimize the maximum of the absolute value of the difference between $R_{ij}^{n+1}(1)$ and $\beta_{ij}$, where $R_{ij}^{n+1}(1)$ is the solution of (4.5). This is formulated as a linear programming problem in which we have the linear inequalities [4],

$$
\frac{1}{P_{ij}} |p_{ij}(1)| + \sum_{k=1}^{3} c^k h_{ij}^k (1) - \beta_{ij} \leq \epsilon_{ij}, \\
\epsilon_{ij} \leq \epsilon.
$$

where the subscripts take on the values appropriate to the trial under consideration. A standard linear programming code [5] is used to determine the constants $c^k$, $\epsilon_{ij}$, and the maximum deviation $\epsilon$. Two numerical experiments are
carried out, one with \( \pm 2\% \) equal magnitude errors in the observations, the other with 2\% Gaussian errors. The incident angle is 60\(^\circ\). The results are given in Table 3, where we show the values of the two constants in the albedo functions, \( a \) and \( b \), the thickness \( c \), and the maximum deviation \( \epsilon \), for each approximation. The results for the case where the errors are all of the same relative size are excellent. The trial using Gaussian errors yields constants which are not quite as good, yet these results are surprisingly better than one might expect.

**TABLE 3.**

**NUMERICAL RESULTS USING MINIMAX CRITERION**

<table>
<thead>
<tr>
<th>Type of Errors</th>
<th>Approximation</th>
<th>( a )</th>
<th>( b )</th>
<th>( c )</th>
<th>Maximum Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \pm 2% )</td>
<td>0</td>
<td>2.00000</td>
<td>-2.00000</td>
<td>1.00000</td>
<td></td>
</tr>
<tr>
<td>equal magnitude</td>
<td>1</td>
<td>1.99948</td>
<td>-1.99961</td>
<td>1.00001</td>
<td>.0200000</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1.99948</td>
<td>-1.99959</td>
<td>1.00001</td>
<td>.0200000</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1.99948</td>
<td>-1.99960</td>
<td>1.00001</td>
<td>.0200000</td>
</tr>
<tr>
<td>2% Gaussian</td>
<td>0</td>
<td>2.00000</td>
<td>-2.00000</td>
<td>1.00000</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1.76462</td>
<td>-1.67267</td>
<td>1.03357</td>
<td>.0294158</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1.76265</td>
<td>-1.67487</td>
<td>1.03841</td>
<td>.0293736</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1.76279</td>
<td>-1.67484</td>
<td>1.03852</td>
<td>.0293722</td>
</tr>
</tbody>
</table>

9. **NUMERICAL EXPERIMENTS V: CONSTRUCTION OF MODEL ATMOSPHERES**

Suppose that we desire to construct a model atmosphere with the optical property that whenever light is incident at angles near the normal, the distribution of diffusely reflected
light is greatest close to $90^\circ$ from the normal. We require that the optical thickness $c$ be about 1.0, and the albedo profile is to be parabolic,

\[ \lambda(\tau) = \frac{1}{2} + a\tau + b\tau^2, \]

where the constants $a$ and $b$ are to be suitably chosen. The albedo should not be greater than unity.

The reflection pattern, for an incident angle of $13^\circ$, is to have the form indicated by the seven $\times$'s in Fig. 5a. The units are given relative to an incident net flux of $\pi$. As our initial estimate, we believe that the slab should have thickness one, and that the parameters be $a = 2$, and $b = -2$. Then the albedo has the form given in Fig. 5b by the curve labelled "Initial", the horizontal line at $\tau = 1$ indicating the upper surface $c$. The reflection function has the form given in Fig. 5a by the dots, whose values are much too low in the region $80^\circ - 90^\circ$. How should the optical design of this slab be modified for better agreement with the requirements? The answer is not at all obvious.

We carry out a numerical experiment in which a better model is to be found, which makes the sum of the squares of the deviations from the desired values a minimum. The condition is to minimize the sum $S$,

\[ S = \sum_{i=1}^{7} (d_i - r_{ik})^2, \]
Fig. 5. Several model atmospheres. (a) The diffusely reflected intensity with incident angle $13^\circ$.
(b) The albedo function.
where $d_i$ is the desired value of the reflection function for output angle arc cosine $u_i$, and $r_{ik}$ is the solution of the differential equations for the $r$ function, and $k = 7$, corresponding to input angle of $13^\circ$. This problem is mathematically the same as the earlier inverse problems of this chapter. The method of solution is also similar, and after five iterations and 3 minutes of computing time, we obtain the solution $a = 1.383, b = -1.140, c = 1.117$. The albedo function is shown in Fig. 5b by the curve labelled "Least squares", the reflection function is indicated by the circled dots in Fig. 5a. A smooth curve is drawn between the dots, showing the probable continuous distribution. This curve is in better agreement with the requirements at $83^\circ$ and $88.5^\circ$.

We perform another experiment in which the criterion is to minimize the maximum deviation. This condition is given by Eqs. (8.1), where $\beta_{ij} = 4u_i d_i$ and $j = 7$. After five iterations, the solution is $a = 0.744, b = -0.415$, and optical thickness $c = 1.431$. The albedo has the form represented in Fig. 5b by the curve labelled "Minimax", and the reflection function is indicated by dots within squares in Fig. 5a. The reflection function for this slab is in very good agreement with the requirements.

Other possible approaches to problems of design include dynamic programming and invariant imbedding [6, 7].
REFERENCES


CHAPTER FOUR

INVERSE PROBLEMS IN RADIATIVE TRANSFER:
ANISOTROPIC SCATTERING

1. INTRODUCTION

Inverse problems in radiative transfer for a medium with anisotropic scattering [1, 2] may be treated in a manner similar to that used in the isotropic case. We consider a plane parallel slab of finite optical thickness \( \tau_1 \). For simplicity let us suppose that both the albedo \( \lambda \) and the anisotropic phase function are independent of optical height. Let us take the phase function to be

\[
p(\hat{n}) = 1 + a \cos \theta, \quad \frac{1}{4\pi} \int_{\Omega} p(\Omega) \, d\Omega = 1,
\]

where \( \theta \) is the scattering angle, and \( a \) is a parameter of the medium and is to be determined on the basis of measurements of the diffusely reflected radiation. An integration over solid angle gives the normalization condition in (1). Let us consider the case in which \( a = 1 \). This approximately corresponds to the forward phase diagram of Saturn [3]. It may be noted that Horák considers inverse problems in planetary atmospheres in [3].

Parallel rays of light of net flux \( \pi \) per unit area normal to the direction of propagation are incident
on the upper surface of the slab. The direction of the rays is characterized by \( u_0 (0 < u_0 \leq 1) \), the cosine of the polar angle measured from the downward normal, and by the azimuth angle \( \varphi_0 \). The phase function may be written as a function of polar and azimuth angles of incidence and scattering, \( p(u, \varphi; u_0, \varphi_0) \). The lower surface of the medium is a perfect absorber.

Let the diffusely reflected intensity in the direction specified by \((u, \varphi)\) be \( r(\tau_1, u, \varphi; u_0, \varphi_0) \), where \( \varphi \) is the azimuth angle \((0 \leq \varphi \leq 2\pi)\). Measurements of \( r \), the set \([b]\), are made and we wish to determine the value of \( a \) for the slab.

2. THE \( S \) FUNCTION

Let us define a function \( S \), which is related to the diffusely reflected intensity function \( r \), by the formula

\[
(1) \quad r(\tau_1, u, \varphi; u_0, \varphi_0) = \frac{S(\tau_1, u, \varphi; u_0, \varphi_0)}{4\mu}
\]

We wish to derive a differential-integral equation for the \( S \) function by the method of invariant imbedding \([2, 4-6]\).

Let us define another function \( \rho \),

\[
(2) \quad \rho(\tau_1, u, \varphi; u_0, \varphi_0) = \frac{u}{u_0} \frac{r(\tau_1, u, \varphi; u_0, \varphi_0)}{\pi}
\]
which is the reflected radiation per unit horizontal area produced by a unit input of radiation on a unit of area in the top surface. Now we add a thin layer of thickness $\Delta$ to the top of the slab of thickness $\tau_1$. The $\rho$ function for this slab may be expressed in the form.

$$\rho(\tau_1 + \Delta; u, \tau_0; u_0, \tau_0) =$$

$$\rho(\tau_1; u, \tau_0; u_0, \tau_0) - \Delta \left( \frac{1}{\mu} + \frac{1}{\mu_0} \right) \rho(\tau_1; u, \tau_0; u_0, \tau_0)$$

$$+ \frac{\Delta}{\mu_0} \frac{1}{4\pi} p(u, \tau; -u_0, \tau_0)$$

$$+ \int_0^{2\pi} \int_0^{2\pi} \frac{1}{4\pi} p(u, \tau; \tau', \tau_0) \frac{\Delta}{\mu} \frac{1}{4\pi} p(u', \tau'; \tau_0, \tau_0) du'd\tau'$$

$$+ \int_0^{2\pi} \int_0^{2\pi} \frac{1}{4\pi} p(-u, \tau; -u_0, \tau_0) \frac{\Delta}{\mu} \frac{1}{4\pi} p(-u', \tau'; u_0, \tau_0) du'd\tau'$$

$$+ \int_0^{2\pi} \int_0^{2\pi} \frac{1}{4\pi} p(u, \tau; u_0, \tau_0) du'd\tau + o(\Delta).$$

The terms on the right hand side of the equation represent the following processes: (1) no interaction in $\Delta$, (2) absorption in $\Delta$ of the incoming and outgoing rays, (3) a single scattering in $\Delta$, (4) multiple scattering in the slab of thickness $\tau_1$ followed by an interaction in $\Delta$, (5) interaction in $\Delta$ followed by multiple scattering in the slab below, (6) multiple scattering in the slab of thickness $\tau_1$ followed by an interaction in $\Delta$, followed
by multiple scattering in the slab of thickness \( \lambda \), and 
(7) \( o(\lambda) \) represents other processes which involve \( \lambda^2 \),
or higher powers of \( \lambda \).

Now the relation between \( S \) and \( P \) is

\[
\rho(\tau_1, \mu, \nu; \mu_0, \nu_0) = \frac{S(\tau_1, \mu, \nu; \mu_0, \nu_0)}{4\pi\mu_0}
\]

so that when we substitute this expression into (3), we find that \( S \) satisfies the equation

\[
S(\tau_1 + \tau, \mu, \nu; \mu_0, \nu_0) = S(\tau_1, \mu, \nu; \mu_0, \nu_0)
- \lambda(1 - \frac{1}{\mu})S(\tau_1, \mu, \nu; \mu_0, \nu_0)
+ \Delta \rho(\mu, \nu; \mu_0, \nu_0)
+ \frac{\lambda}{4\pi} \int_0^{2\pi} \int_0^1 \rho(\mu', \nu'; \mu_0, \nu_0) S(\tau_1, \mu', \nu'; \mu_0, \nu_0) d\mu' d\nu'
\]

(5)

\[
+ \frac{\lambda}{4\pi} \int_0^{2\pi} \int_0^1 S(\tau_1, \mu', \nu'; \mu_0, \nu_0) \rho(-\mu', \nu'; \mu_0, \nu_0) d\mu_0 d\nu_0
\]

\[
+ \frac{\lambda^2}{(4\pi)^2} \int_0^{2\pi} \int_0^1 S(\tau_1, \mu', \nu'; \mu_0, \nu_0) \rho(\mu', \nu'; \mu_0, \nu_0) d\mu_0 d\nu_0 + o(\lambda)
\]

We expand the left hand side of Eq. (5) in powers of \( \lambda \).
We let \( \Delta \to 0 \) and we obtain the desired integro-differential equation

\[
\frac{\partial S(\tau_1, \mu, \varphi; \mu_0, \varphi_0)}{\partial \tau_1} + \left( \frac{1}{\mu} + \frac{1}{\mu_0} \right) S = \lambda \left\{ p(\mu, \varphi; -\mu_0, \varphi_0) + \frac{1}{(4\pi)^2} \int_0^{2\pi} \int_0^1 S(\tau_1, \mu', \varphi'; \mu_0, \varphi_0) p(-\mu_0', \varphi_0'; -\mu_0, \varphi_0) \frac{d\mu_0'}{\mu_0} \right\}.
\]

A simplification arises if it is assumed that the phase function may be expanded in the Fourier series

\[
p(\mu, \varphi, \mu_0, \varphi_0) = \sum_{m=0}^{M} c_m P_m(\cos \theta),
\]

where \( P_m(x) \) is the Legendre polynomial of degree \( m \).

The angular dependence of expansion (8) may be decomposed into polar and azimuth factors by the use of the addition rule of Legendre functions. Then Eq. (8) becomes
The function \( p^{(m)}(x) \) is the associated Legendre function of degree \( i \), order \( m \). Noting the form of this equation, we expand the \( S \) function in a similar manner.

\[
S(\tau_{1, \mu_1} \ldots \mu_0, 0) = \sum_{m=0}^{M} S^{(m)}(\tau_{1, \mu_1} \ldots \mu_0) \cos m(\omega - \omega_0).
\]

Substitution of (10) in (7) leads to the equations for the Fourier components of \( S \):

\[
\frac{d}{d\tau_1} S^{(m)} + \left( \frac{1}{\mu} + \frac{1}{\mu_0} \right) S^{(m)} = \gamma(2 - \delta_{0m}) \sum_{i=m}^{M} (-1)^{i+m} \frac{\mu_0}{i(i+m)!} \frac{d}{d\tau_1} p^{(m)}(\mu) y^{(m)}(\mu_0)
\]

where

\[
\gamma^{(m)}(\mu) = p^{(m)}(\mu) + \frac{(-1)^{i+m}}{2(2-\delta_{0m})} \int_0^{1} S^{(m)}(\tau_{1, \mu, u'}) p^{(m)}(u') \frac{du'}{u_0}
\]

for \( m = 0, 1, 2, \ldots, M \). The functions \( S^{(m)}(\tau_{1, \mu, \mu_0}) \) possess the symmetry property

\[
S^{(m)}(\tau_{1, \mu, \mu_0}) = S^{(m)}(\tau_{1, \mu_0, \mu})
\]

The initial conditions are

\[
S^{(m)}(0, \mu, \mu_0) = 0.
\]
By the use of Gaussian quadrature on the interval
(0,1), the integrals (12) are replaced by sums. Also,
the function \( S^{(m)}(\tau_1, u, u_0) \) is replaced by a function
of one independent variable, \( S^{(m)}(\tau_1) \), where the angles
are discretized such that \( u_0 = \omega_i \) and \( \omega = \omega_i \),
i,j = 1, 2, ..., N. Then we have the approximate system.

\[
\frac{dS^{(m)}(\tau_1)}{d\tau_1} + \left( \frac{1}{u_i} + \frac{1}{u_j} \right) S^{(m)} = \\
\lambda(2-\epsilon_{0m}) \sum_{k=m}^{M} (1)^{k+m} \left( \frac{k-m}{k+\omega} \right) c^{m}_{k,i} k^{m}_{j,k} .
\]

\[ (m = 0, 1, ..., M; \quad i = 1, 2, ..., N; \quad j = 1, 2, ..., N) , \]

where

\[
\omega^m_{k,j} = \frac{\lambda^m_{k}}{p^m_{k}(u_j)} + \frac{(-1)^{k+m}}{2(2-\epsilon_{0m})} \sum_{j=1}^{N} S^{(m)}(\tau_1) \frac{p^m_{k}(u_j)}{u_j} W_j .
\]

The discrete cosines \( \omega^m_{j} \) are the roots of the shifted
Legendre polynomial of degree \( N \), \( P_N(x) \) and the quantities
\( W_j \) are the corresponding weights. The initial conditions
are

\[
S^{(m)}_{ij}(0) = 0 .
\]

The solution of this initial value integration problem for
a system of ordinary differential equations (15) is approx-
imately equal to the solution of the original integro-
differential system.
3. AN INVERSE PROBLEM

Consider the case in which the slab is of thickness \( \tau_1 = 0.2 \), the albedo is \( \lambda = 1 \), and we choose \( N = 7 \) for the quadrature. For the phase function

\[ p = 1 + a \cos \theta, \]

the parameters are

\[ M = 1, \ c_0 = 1, \ c_1 = a = 1. \]

For a numerical experiment, we take Eqs. (2.15) and integrate from \( \tau_1 = 0 \) with initial conditions (2.17) to \( \tau_1 = 0.2 \), using an integration grid size of \( \Delta \tau_1 = 0.01 \).

Using (2.10) and (2.1), we produce

\[ b_{ijk} = r(0.2, \ \omega_i, \ \omega_k; \ \omega_j, \ \varphi_0) \]

for \( \varphi_0 = 0 \), and \( \omega_k = 0^\circ, \ 30^\circ, \ 60^\circ, \ \ldots, \ 180^\circ \) as \( k = 1, 2, \ldots, 7 \). The set \( \{b_{ijk}\} \) represents our measurements of the diffuse reflection field, from which we hope to estimate the unknown parameter \( a \). The condition shall be to minimize the sum of squares of deviations,

\[ \sum_{i,j,k} [r(0.2, \ \omega_i, \ \omega_k; \ \omega_j, \ \varphi_0) - b_{ijk}]^2 \]

where the function \( r \) is the solution of the Eqs. (2.15) - (2.17) using (2.1) and (2.10). The measurements for \( \tau = 0^\circ \) and for \( \tau = 180^\circ \) are shown in Fig. 1. when \( \varphi_0 = 0.5 \).

These data were produced numerically with the
Fig. 1. Fourteen observations of the diffusely reflected intensity
\( r(u, \phi; u_0, \varphi_0) \), \( u_0 = 0.5 \), \( \varphi_0 = 0^\circ \), phase function \( p(\varphi) = 1 + \cos \theta \).
use of Eqs. (2.1), (2.10) - (2.14). The program for the calculation of the $r$ function is given in Appendix D.

4. METHOD OF SOLUTION

This problem may be solved by successive approximations using quasilinearization [7.8]. Let us write the function $S_{ij}^{(m)}$ as $S_{mij}$, and similarly $\psi_{ki}^{m} - \psi_{mkj}^{m}$.

The linear equations for the $(n+1)^{st}$ approximation are

$$
\frac{dS_{n+1}^{mij}}{d\tau_{l}} = \left( \frac{1}{\omega_{i}} + \frac{1}{\omega_{j}} \right) S_{n}^{mij}
$$

$$+ \chi(2^{r} \sum_{0}^{m}) \frac{1}{\omega_{i}} (\sum_{k=m}^{n} \frac{(k-m)!}{(k+m)!} \psi_{mki} \psi_{mkj})
$$

$$+ (S_{mij}^{n+1} - S_{mij}^{n})(-1)\left( \frac{1}{\omega_{i}} + \frac{1}{\omega_{j}} \right)
$$

$$+ \chi(2^{r} \sum_{0}^{m}) \sum_{l=1}^{N} \left[ (S_{mlj}^{n+1} - S_{mlj}^{n}) \psi_{mlj}^{n} + (S_{mlj}^{n+1} - S_{mlj}^{n}) \psi_{mlj}^{n} \right]
$$

$$+ \chi(2^{r} \sum_{0}^{m}) (r_{1}^{n+1} - r_{1}^{n})(-1)14m (1-m)^{l} \psi_{mlj}^{m} \psi_{mlj}^{m}
$$

(m = 0, 1; i = 1, 2, ..., 7; j = 1, 2, ..., 7).

$$
\frac{da^{n+1}}{d\tau_{l}} = 0 \text{ for } a = c_{1}.
$$

where
\[
\begin{align*}
\text{(3)} & \quad \frac{\partial^2 S}{\partial t^2} = \frac{1}{2(2-\delta_{0m})} \sum_{k=m}^{n} \left(\sum_{j=1}^{n} P^{m_1}_j \frac{W_j}{u_j} \right) (-1)^{k+m} \frac{(k-m)!}{(k+n)!} \delta_{mk} n^m k

\text{and}

\text{(4)} & \quad \frac{\partial S}{\partial t} = P^{m_1}_j + \frac{(-1)^{k(m-1)} K}{2(2-\delta_{0m})} \sum_{j=1}^{n} S^m_{m_1,j} P^{m_1}_j \frac{W_j}{u_j}.
\end{align*}
\]

In these equations

\[
\begin{align*}
\text{(5)} & \quad a^{n+1} = c^{n+1}, \quad a^n = c^n, \quad c_0^{n+1} = c_0^n = 1.
\end{align*}
\]

The initial conditions are

\[
\begin{align*}
\text{(6)} & \quad S^{n+1}_{mij}(0) = 0
\end{align*}
\]

and the boundary condition is

\[
\begin{align*}
\text{(7)} & \quad \sum_{i,j,k} \left\{ \sum_{m=0}^{n} S^{n+1}_{mij}(0.2) \cos \mu_k - 4 \mu_1 b_{ijk} \right\} = 0.
\end{align*}
\]

Let us represent the \((n+1)\)st approximation of \(S\) as a linear combination of a particular solution and a homogeneous solution

\[
\begin{align*}
\text{(8)} & \quad S^{n+1}_{mij}(\tau_1) = P^{m}_{mij}(\tau_1) + h^{m}_{mij}(\tau_1).
\end{align*}
\]

In terms of numerically known quantities,

\[
\begin{align*}
\text{(9)} & \quad a^{n+1} = \sum_{i,j,k} \left(4 \mu_1 b_{ijk} - P_{0ij} - P_{iij c c s k}(h_{0ij} + h_{iij c c s k}) \right) \frac{\sum_{i,j,k} \left(h_{0ij} + h_{iij c c s k}\right)}{\sum_{i,j,k} \left[h_{0ij} + h_{iij c c s k}\right]^2}
\end{align*}
\]
where the functions \( p \) and \( h \) are evaluated at \( \tau_1 = 0.2 \), and the initial conditions for \( p \) and \( h \) are suitably chosen.

By making use of the symmetry property of \( S \) we need consider not a system of \( 2N^2 \) equations, but only \( 2N(N+1)/2 = N(N+1) \) equations. For \( N = 7 \), this means that \( 7 \times 8 = 56 \) equations define the particular solution, and another 56 define the homogeneous solution. Twenty-one integration grid points cover the range \( 0 \leq \tau_1 \leq 0.2 \) with \( \Delta \tau_1 = 0.01 \). The storage requirements are \( 21 \times 56 \) for the \( p \) solution, \( 21 \times 56 \) for the \( h \) solution and \( 21 \times 56 \) for \( S_{\text{mij}}^n \). This problem is certainly feasible for numerical solution with the IBM 7044 or the 7090. Numerical experiments will be carried out in the near future. Such studies should prove useful in the planning and analysis of investigations of planetary atmospheres [3,9–18], stellar radiation in the galaxy [19], and radiation fields in the sea [20–22].
REFERENCES


CHAPTER FIVE

AN INVERSE PROBLEM IN NEUTRON TRANSPORT THEORY

1. INTRODUCTION

The theory of neutron transport and the theory of radiative transfer [1] are devoted to problems of determining the properties of radiation fields produced by given sources in a given medium. Inverse problems in transport theory are those in which we seek to determine the properties of the medium, given those of the incident radiation and the radiation fields [2-4].

In this chapter, we study inverse problems in transport theory from the point of view of dynamic programming [5]. Our aim is to produce a feasible computational method for estimating the properties of the medium based upon measurements of radiation fields within the medium. The invariant imbedding approach to transport theory is sketched in Ref. 6.

For ease of exposition we consider a one-dimensional transport process. The method described here can be generalized to the vector-matrix case, and thus to the slab geometry with anisotropic scattering, to wave propagation [7], and to transmission lines.
2. FORMULATION

Consider the one dimensional medium shown in Fig. 1.

\[ 0 \rightarrow \frac{a_1}{b_0=0} \frac{a_2}{b_1} \frac{a_3}{b_2} \ldots \frac{a_N}{b_{N-1}} c \]

Fig. 1. A one dimensional transport process.

It consists of \( N \) homogeneous sections \((b_i, b_{i+1})\), \( i = 0, 1, 2, \ldots, N - 1 \). When a neutron travels through a distance \( \Delta \) in the \( i^{th} \) section, there is probability \( a_i \) that it will interact with the medium. The result of an interaction is that the original neutron is absorbed and two daughter neutrons appear, one traveling in each direction. Suppose that \( c \) neutrons per unit time are incident from the right and zero neutrons per unit time from the left. We denote the average number of particles per unit time passing the point \( x \) and moving to the right by \( u(x) \) and the same quantity for the leftward moving particles by \( v(x) \). Suppose that measurements on the internal intensities are made at various points \( x = x_i \neq b_j \); e.g.,

\[ u(x_i) \approx w_i, \quad i = 1, 2, \ldots, M. \]

Our aim is to estimate the characteristics of the medium, the quantities \( a_i, \quad i = 1, 2, \ldots, N \), on the basis of these observations.
As is shown in Ref. 6, the internal intensities satisfy the differential equations

\[ \dot{u} = a_i v , \]
\[ \dot{v} = a_i u , \]

where the dot indicates differentiation with respect to \( x \).

The analytical solution is of no import, since we wish to consider this as a prototype of more complex processes for which a computational treatment is mandatory. In addition, \( u(x) \) and \( v(x) \) are continuous at the interfaces

\[ u(b_i - 0) = u(b_i + 0) \]
\[ v(b_i - 0) = v(b_i + 0) , \quad i = 1, 2, \ldots, N , \]

and

\[ u(0) = 0 \]
\[ v(b_N) = c . \]

We wish to select the \( N \) constants \( a_1, a_2, \ldots, a_N \), so as to minimize the sum of the squares of the deviations \( S \),

\[ S = \sum_{i=1}^{M} \left( u(x_i) - w_i \right)^2 . \]
3. DYNAMIC PROGRAMMING

Let us suppose that the functions \( u \) and \( v \) are subject to the conditions of Section 2 and

\[
(1) \quad u(b_K) = c_1 \\
(2) \quad v(b_K) = c_2 .
\]

In addition we write

\[
(3) \quad f_K(c_1, c_2) = \min \sum_{i=1}^{M_K} \{u(x_i) - w_i\}^2 .
\]

where the minimization is over the absorption coefficients \( a_1, a_2, \ldots, a_K \). The number of observations on the first \( K \) intervals is \( M_K \). We view \( K \) as a parameter taking on the values 1, 2, …, and \( c_1 \) and \( c_2 \) are also viewed as variables. Then we write

\[
(4) \quad f_1(c_1, c_2) = \sum_{i=1}^{M_1} \{u(x_i) - w_i\}^2 ,
\]

where

\[
(5) \quad u(b_1) = c_1 \\
(6) \quad v(b_1) = c_2 \\
(7) \quad \dot{u} = a_1 v \\
(8) \quad -\dot{v} = a_1 u .
\]

The absorption coefficient \( a_1 \) is chosen so that
(9) \( u(0) = 0 \).

In addition, the principle of optimality yields the relationship

\[
f_{K+1}(c_1, c_2) = \min \left\{ d_{K+1} + f_K(c_1', c_2') \right\}.
\]

(10) \( K = 1, 2, \ldots \).

(11) \( d_{K+1} = \sum_i \left( u(x_i) - \nu_i \right)^2 \).

with \( i \) ranging over integer values for which

(12) \( b_K < x_i < b_{K+1} \),

and

(13) \( \dot{u} = a_{K+1} v \), \( u(b_{K+1}) = c_1 \)

(14) \( \dot{v} = a_{K+1} u \), \( v(b_{K+1}) = c_2 \).

In addition we have put

(15) \( c_1' = u(b_K) \)

(16) \( c_2' = v(b_K) \).

In the usual manner of dynamic programming this leads to a computational scheme for computing the sequence of functions of two variables \( f_1(c_1, c_2), f_2(c_1, c_2), \ldots \).
and in principle solves our estimation problem. In the event that we do not wish to require that \( u(0) = 0 \), we may determine the function \( f_1(c_1, c_2) \) this way:

\[
f_1(c_1, c_2) = \frac{a_1}{\lambda} \left[ |X_{u_2}(0)| + \sum_{i=1}^{M} |u(x_i) - w_i| \right] .
\]

where \( \lambda \) is a suitably large parameter.

4. AN APPROXIMATE THEORY

While the original physical problem is a two-dimensional problem, it may be well-represented as a one-dimensional problem. Suppose that there are \( K \) segments of the medium and that the input is \( v(b_K) = c \). The absorption coefficients \( a_1, a_2, \ldots, a_K \) should be chosen to secure a minimum sum of squares of deviations from the measurements. Having picked the absorption coefficients, we may calculate the reflection coefficient \( r(v(b_K)) \) for this segmented medium. At the end \( b_K \), the function \( u \) is essentially determined by \( v \) and \( r(v) \), \( u(b_K) = v(b_K) r(v(b_K)) \). The single variable \( v(b_K) = c \) may then suffice to specify the state at the right end of the \( K^{th} \) segment.

Let us define the function \( g_N(c) \)

\[
g_N(c) = \text{the smallest sum of squares of deviations on the first } N \text{ segments when the input is } c,
\]

and the function \( R_N(c) \),
\[ R_N(c) = \text{the reflection coefficient that results when} \]
\[ \text{the optimal set of absorption coefficients is} \]
\[ \text{used on the first } N \text{ segments, the input being} \]
\[ c = v(b_N). \]

The function \( g_{N+1}(c) \) satisfies the inequality
\[ g_{N+1}(c) \leq \min \left\{ d_{N+1} + g_N(c') \right\}, \]
where
\[ d_{N+1} = \sum_i [u(x_i) - v_i]^2, \quad b_N < x_i < b_{N+1}, \]
and
\[ \dot{u} = a v, \quad v(b_{N+1}) = c, \]
\[ -\dot{v} = a v - v(b_N)R_N(v(b_N)) = u(b_N), \]
and
\[ c' = v(b_N). \]

We do not have recurrence relations for the sequences of
functions \( g_N(c) \) and \( R_N(c) \). We replace Eqs. (3), (4),
and (5) by an approximate set, where instead of \( g_N(c) \) we
introduce the sequence \( f_N(c) \), and instead of \( R_N(c) \) we
introduce \( r_N(c) \). In our approximate theory we produce
\( f_{N+1}(c) \) from the recurrence formula
\[ f_{N+1}(c) = \min \left\{ d_{N+1} + f_N(c') \right\}, \]
where

\[(8) \quad d_{N+1} = \sum_i \left[ u(x_i) - w_i \right]^2, \quad b_N < x_i < b_{N+1}. \]

and

\[c' = v(b_N).\]

The following boundary value problem.

\[(9) \quad \dot{u} = a\ v, \quad v(b_{N+1}) = c, \]

\[-\dot{v} = a\ u, \quad v(b_N) = r_N(v(v_N)) = u(b_N).\]

must be satisfied. The quantity

\[(10) \quad r_{N+1}(c) = r(b_{N+1})\]

is obtained as the solution of the initial value problem

\[(11) \quad \dot{r} = a(1 + r^2), \quad r(b_N) = r_N(c').\]

For \(N = 1\) we define

\[(12) \quad f_1(c) = \min_i a \left[ u(x_i) - w_i \right]^2, \]

where the summation is over indices for which

\[(13) \quad 0 < x_i < b_1, \]

and \(
\) is a weighting constant. Also we have

\[(14) \quad \dot{u} = a\ v, \quad u(0) = 0, \]

\[-\dot{v} = a\ u, \quad v(b_1) = c.\]
We define
\begin{equation}
(15) \quad r_1(c) = r(b_1)
\end{equation}
where
\begin{equation}
(16) \quad r = a(1 + r^2), \quad 0 \leq x \leq b_1.
\end{equation}

The purpose of introducing the weight $a \geq 1$ is to insure a good fit over the first segment.

Assuming that a unique minimizing solution exists, we can show that the results of our approximate theory are exact, if the observations $w_{11}$ are perfectly accurate. We reason inductively. For the one segment process, there exists an input $c_1$ for which $f_1(c_1) = 0$ by Eq. (12), and the reflection coefficient is $r_1(c_1)$. We assume that there exists an input to the medium of $N$ segments, $c_N$, such that $f_N(c_N) = 0$, and that the reflection coefficient for this medium is $r_N(c_N)$. For the medium of $N+1$ segments, there is an input $c_{N+1}$ such that $d_{N+1} = 0$, and the input (to the left) at $b_N$ which satisfies condition (9) is $v(b_N) = c_N$. Therefore $f_{N+1}(c_{N+1}) = 0$, and the solution is exact.

In this manner we have reduced the original multi-dimensional optimization process to a sequence of one-dimensional processes.
5. A FURTHER REDUCTION

The solving of the nonlinear boundary value problem of Eq. (4.9) can be a source of difficulty. To aid in this process we note that we can write

\[
(1) \quad v(b_N) = c T + u(b_N) R.
\]

which follows simply from one of Chandrasekhar's invariance principles [1]. The transmission coefficient \( T \) and the reflection coefficient \( R \) of the \((N+1)\)st segment are calculated from the solutions of the initial-value problems [6]

\[
(2) \quad \dot{r} = a(1 + r^2), \quad r(0) = 0.
\]

\[
(3) \quad \dot{t} = a r t, \quad t(0) = 1,
\]

and

\[
(4) \quad R = r(b_{N+1} - b_N).
\]

\[
(5) \quad T = t(b_{N+1} - b_N).
\]

In this way the second condition in Eq. (4.9) becomes

\[
(6) \quad r_N(v)v = (v - c T) / R.
\]

a nonlinear equation for \( v = v(b_N) \)

6. COMPUTATIONAL PROCEDURE

The calculation of \( f_{N+1} \) for a given value of the parameter \( c \) may proceed as follows. We take a value of
the coefficient \( a \), and we produce numerically the reflection and transmission coefficients, \( R \) and \( T \). Assuming we can solve Eq. (5.6) for \( v(b_N) \), we go on to solve the linear two-point boundary-value problem of Section 4 by producing numerically two independent solutions of these homogeneous equations and determining constants so that the boundary conditions are met. Then the sum of squares of deviations \( d \) is computed, and the cost \( \{d + f_N(v(b_N))\} \) is evaluated. We go through these steps for all the admissible choices of \( a \), and the costs are compared. The value of \( a \) which makes the cost a minimum is the choice for the \((N+1)^{st}\) slab. The whole procedure is repeated for the range of values of \( c \) and of \( N \).

It may be noted that in the calculation of the reflection coefficient \( r_{N+1} \), the initial condition \( r_N \) is known only computationally on a grid of values of the argument. Experiments are needed to determine the required fineness of grid to achieve the required accuracy.

It is possible to derive recurrence relations for \( f_N'(c) \) and \( r_N'(c) \), and these can be employed in a variety of ways to improve the accuracy of the method. Numerical experimentation would have to be carried out to obtain reliable estimates of running times and accuracies [9]. The method proposed here can be extended to treat the case where the interface points are not known, though the computational effort will be greatly increased.
Experience with many similar problems leads us to believe that the proposed procedure is perfectly feasible [8, 9].

7. COMPUTATIONAL RESULTS

Production of observations. We consider a homogeneous rod of unit length with absorption coefficient $a = 0.5$. We produce the internal fluxes to the right and to the left due to a unit input flux to the left at the end $x = 1$, and no input at the other end $x = 0$. To do this, we use the fact that the quantity $v(1)$ is the reflection coefficient for the slab which is $\tan a$ [6]. We integrate the transport equations with the initial values $u(1) = 1$, $v(1) = \tan a$, from $x = 1$ to $x = 0$. This procedure yields $u(x)$ and $v(x)$ throughout the rod.

Two-dimensional dynamic programming procedure for the determination of the absorption coefficients. The rod is divided up into 10 homogeneous sections of equal length. From the set of exact measurements, $w_j = u(x_j)$, we wish to determine the set of optimizing parameters $a_N$ in each section. The correct solution is $a_N = 0.5$ for $N = 1, 2, ..., 10$.

In stage one of the multi-stage decision process, the rod is considered to consist of one segment extending from $x = 0$ to $x = 0.1$. If $c_1 = u(0.1), c_2 = v(0.1)$, we choose the coefficient which makes $u(v) = 0,
regardless of the measurements in this segment. The minimum cost is 
\[ f_1(c_1', c_2') = \sum_{i} [u(x_i) - w_i]^2 \]
where \( u(x_i) = \sin a x_i \), \( 0 < x_i < 0.1 \). This calculation is carried out for each value of \( c_1 \) and \( c_2 \).

The computations for the other stages \( N = 2, 3, 4, \ldots \), may be best indicated by the following outline:

**TWO-DIMENSIONAL DYNAMIC PROGRAMMING CALCULATIONS**

For each stage \( N = 2, 3, 4, \ldots \),

1. Print \( N \)
2. For each \( c_1 \)
   1. For each \( c_2 \)
      1. For each \( a \)
         1. Integrate to produce \( c_1' = u(b_{N-1}), c_2' = v(b_{N-1}) \),
            \[
            \begin{cases}
            \dot{u} = a v, u(b_N) = c_1 \\
            -\dot{v} = a u, v(b_N) = c_2
            \end{cases}
            \]
      2. Compute \( d = \sum_{i} [u(x_i) - w_i]^2 \)
      3. Find \( f_{N-1}(c_1', c_2') \) by interpolation
      4. Set \( S(a) = d + f_{N-1}(c_1', c_2') \)
   2. Search for \( f_N(c_1', c_2') = \min_a \{ S(a) \} \)
   3. Print \( c_1, c_2, a_N, c_1', c_2', f_N(c_1', c_2') \)
3. For each \( c_1 \)
   1. For each \( c_2 \)
      1. Shift \( f_N(c_1, c_2) \rightarrow f_{N-1}(c_1, c_2) \)
There are four levels of computation: the stage $N$, the state $c_1$, the state $c_2$, the parameter $a$. The large brackets over the steps which must be carried out at each level. By the statement "shift $f_N \rightarrow f_{N-1}$", we represent the discarding of the costs for stage $N-1$, and the replacement of $f_{N-1}$ by the just computed costs for stage $N$, in readiness for the next stage. This saving in storage is allowed by the recurrence formula linking the current cost with the cost of only the previous stage. The interpolation may be carried out by the use of a linear formula in two dimensions, $c_1$ and $c_2$. The print-out value of $a$ is, of course, the optimal value.

In our numerical trial, we execute the algorithm for three stages only, the rod then extending from $x = 0$ to $x = 0.3$. The exact observations are

\[
\begin{align*}
  u(0.02) &= 0.11394757 \times 10^{-1} \\
  u(0.05) &= 0.28484388 \times 10^{-1} \\
  u(0.08) &= 0.45567610 \times 10^{-1} \\
  u(0.12) &= 0.68328626 \times 10^{-1} \\
  u(0.15) &= 0.85381951 \times 10^{-1} \\
  u(0.18) &= 1.0241607 \times 10^{0} \\
  u(0.22) &= 1.12509171 \times 10^{0} \\
  u(0.25) &= 1.14206610 \times 10^{0} \\
  u(0.28) &= 1.15900853 \times 10^{0}. \\
\end{align*}
\]

The range of $N$ is 1 to 3, the section interfaces lying at $x = 0.1, 0.2, 0.3$. The range of $c_1$ is 0.00 (0.01) 0.20, 21 values; the range of $c_2$ is 1.120 (0.002) 1.140, 11 values. The five allowed values of $a$ are 0.3 (0.1)
From the direct calculation, i.e., when the true structure of the rod is given, we know the conditions at the right end \( x = 0.3 \) which are \( u(0.3) = 0.17028385 \), \( v(0.3) = 1.1266986 \). The inverse calculations do not produce clearly the correct results \( a_1 = a_2 = a_3 = 0.5 \). It is believed that the grids of values of \( c_1 \) and of \( c_2 \) are not sufficiently fine, and that substantially improved results cannot be obtained without a great increase in computing expense. The computing time for the IBM 7044 is 1-1/2 minutes for these three stages. The one-dimensional reduction appears attractive in view of these results.

One-dimensional dynamic programming approximation for the determination of the absorption coefficient. The rod of unit length is divided into five sections of equal length 0.2. Armed with the internal measurements \( w_i \approx u(x_i) \), we wish to determine the absorption coefficient of each slab. The correct choices are \( a_N = 0.5 \) for \( N = 1, 2, \ldots, 5 \). In the one-dimensional case, the only state variable is \( c = v(b_N) \).

The outline immediately following lists the calculations for producing \( a_1, f_1(c) \) and \( r_1(c) \) for \( N = 1 \), and the next outline shown the general scheme, \( N = 2, 3, \ldots \).
For stage $N = 1$

1. Print $N$
2. For each $c = v(b_N)$
   1. For each $a$
      1. Solve 2 point boundary-value problem for $v(0) = c'$.
         \[
         \begin{cases}
         \dot{u} = a \cdot v, \quad u(0) = 0 \\
         \dot{v} = a \cdot u, \quad v(0.2) = c
         \end{cases}
         \]
      2. Integrate to produce $u(x)$.
         \[
         \begin{cases}
         \dot{u} = a \cdot v, \quad u(0) = 0 \\
         \dot{v} = a \cdot u, \quad v(0) = c'
         \end{cases}
         \]
      and simultaneously calculate $d = \sum_{i} [u(x_i - w_i)]^2$.
      and keep a running estimate of $f_1(c) \approx \min_{a} \{d\}$.
   2. Integrate to produce $r_1(c) = \rho(0.2)$,
      \[
      \dot{\rho} = a(1 + \rho^2), \quad \rho(0) = 0
      \]
3. Print $c$, $a$, $c'$, $r_1(c)$, $f_1(c)$
For each stage $N = 2, 3, 4, \ldots$

1. Print $N$
2. For each $c$
   1. For each $a$
      1. Produce $R = \rho(0.2)$, $T = t(0.2)$ by integration
         \[
         \dot{\rho} = a(1 + \rho^2), \quad \rho(0) = 0
         \]
         \[
         \dot{t} = a \rho \ t, \quad t(0) = 1
         \]
      2. Solve the nonlinear equation for $c' = \nu(k_{N-1})$ and $r_{N+1}(c')$
         \[
         R \ c' r_{N-1}(c') = c' - c \ T
         \]
      3. Solve 2 point boundary-value problem for $e' = u(0) = u(b_{N-1})$
         \[
         \dot{u} = a \ v, \quad v(0) = c'
         \]
         \[
         \dot{v} = a \ u, \quad v(0.2) = c
         \]
      4. Integrate to produce $u(x)$
         \[
         \dot{u} = a \ v, \quad u(0) = e'
         \]
         \[
         \dot{v} = a \ u, \quad v(0) = c'
         \]
         and calculate $d = \sum_{i} [u(x_i) - w_i]$, calculate $d$
      5. Find $f_{N-1}(c')$ by interpolation
      6. Set $S(a) = d + f_{N-1}(c')$ and keep a running estimate of $f_N(c) = \min_a [S(a)]$
   2. Integrate to produce $r_{N}(c) = \rho(0.2)$
      \[
      \dot{\rho} = a(1 + \rho^2), \quad \rho(0) = r_{N-1}(c')
      \]
   3. Print $c$, $a_N$, $c'$, $r_{N}(c)$, $f_N(c)$
To solve the nonlinear equation for \( c' = v(b_N) \) where \( r(c') \) is known only on a grid of points, we compute the expressions \( g_1 = R c' r_N(c') \), \( g_2 = c' - c_T \), and we take their difference \( D = g_1 - g_2 \). If \( D = 0 \), then \( c' \) has been found. Otherwise we repeat the procedure for each discrete value \( c_i' \), until the sign of \( D_i \) is opposite to that of \( D_{i-1} \). We then interpolate linearly to find the quantity \( c' \) which makes \( D = 0 \). If the sign of \( D \) does not change, i.e., the curves \( g_1 \) and \( g_2 \) do not intersect, then the corresponding value of \( a \) is definitely not allowed to be the coefficient for the segment in question.

If the minimum cost \( f_N(c) \) is large for a given state and all remaining states may be deleted from further consideration. This provides a saving in computing time for each state to be considered requires many calculations. Of course, the precaution must be taken to order the \( c' \)'s properly so that no potentially vital state is lost.

The proposed one-dimensional scheme has been tested numerically. The range of \( N \) is 1 to 3, the interfaces of the sections being located at \( x = 0.2, 0.4, \) and \( 0.6 \). The states \( v(b_N) = c \) are 1.04955 (0.00015) 1.13385. 563 in all. This number is reduced in stage 2 to 546, by the use of the above test. Four values of the absorption coefficient \( a \) are allowed: 0.1, 0.3, 0.5, and 0.7. There are nine perfectly accurate observations of \( u \) per segment, a total of 27 data points. The integration method is Adams-Moulton with a grid size of 0.01.
From the output of our computation, we see that the minimum value of $f_3(c)$ is $0.387 \times 10^{-8}$ and occurs when the input flux is $v(0.6) = c = 1.08855$ and the absorption coefficient for the segment of the medium between $x = 0.4$ and $x = 0.6$ is taken to be $a = 0.5$. This is very close to the true answer, $v(0.6) = 1.08860$, and the value of the parameter $a$ is correct. The calculation tells us that the next state at $x = 0.4$ will be $v = 1.11673$. The nearest grid point in $c$ is $1.11675$, and the cost $f_2(1.11675)$ is indeed a minimum, $0.824 \times 10^{-9}$. The absorption coefficient for segment 2 is 0.5, the correct solution. The next state at $x = 0.2$ is predicted to be $1.13377$. The nearest discrete state is $1.13385$, possessing a cost $f_1(1.13385) = 0.181 \times 10^{-9}$. The absorption coefficient is 0.5, again the correct answer. The solutions at each state are clearly found, the minimum cost differing from the others by at least an order of magnitude. These dynamic programming calculations of about 20 minutes have very accurately determined the input, and they have identified the medium.

Now we wish to test the one-dimensional method of determining the structure of the medium when the measurements are few and of limited accuracy. We consider the rod of length 0.8 consisting of 4 segments of equal length 0.2. There are again the same 563 discrete states in $c$, and the same four possible absorption coefficients 0.1, 0.3, 0.5 and 0.7. However, there are only three observations per segment and these are correct to only two
significant figures. Knowing the inputs to the first three stages $N = 1$, $2$, $3$, we see from the output of the calculations that the absorption coefficients are $a_1 = a_2 = a_3 = 0.5$, the correct solution in this region. On the other hand, we are not able to accurately identify the input to a given segment on the basis of these calculations, because the minimum of the function $f$ is broad and it is not centered at the correct value of the input $c$. For stage $N = 4$, the value of $a_4$ is determined to be $0.3$, and incorrect value. These experiments might serve as a warning to the experimental investigator. They show that the processing of data with a small number of measurements requires higher accuracy than two figures, and that if the measurements are of limited accuracy, many measurements should be made. This trial consumes 34 minutes of IBM 7044 computing time. This time of calculation could be reduced greatly by streamlining the calculations. No attempt to do this was made here; feasibility was our sole concern.

For other approaches to transport theory, see Refs. 10–14.
REFERENCES


CHAPTER SIX

INVERSE PROBLEMS IN WAVE PROPAGATION:
MEASUREMENTS OF TRANSIENTS

1. INTRODUCTION

The wave equation

\[ \Delta u = \frac{1}{c^2} u_{tt}, \]

is one of the basic equations of mathematical physics. If we suppose that the local speed of propagation is a function of position

\[ c = c(x, y, z) \]

then the difficulties in studying the various initial and boundary value problems which arise are well known [1, 2, 3]. In the sections which follow, we wish to study some of the inverse problems which arise when we attempt to determine the properties of a medium on the basis of observations of a wave passing through the medium. Such problems are of central importance in such varied areas as ionospheric and tropospheric physics, seismology, and electronics. Some early results are due to Ambarzumian [4] and Borg [5].
We shall discuss some one-dimensional problems. Our basic technique is to reduce the partial differential equation in (1) to a system of ordinary differential equations either by using Laplace transforms or by considering the steady-state situation. Then our previously developed methodology is applicable. For simplicity and specificity we shall employ the nomenclature associated with the problem of the vibrating string. In passing, we note that our methodology is applicable to the diffusion equation, to the telegrapher's equation, and to other similar propagation equations.

2. THE WAVE EQUATION

Consider an inhomogeneous medium which extends from $x = 0$ to $x = 1$, for which the wave equation

\begin{equation}
   u_{tt} = c^2 u_{xx}
\end{equation}

is applicable. In this equation, the disturbance $u(x,t)$ is a function of position and time. Let us assume that the wave speed $c$ satisfies the equation

\begin{equation}
   c^2 = a + bx,
\end{equation}

where $a$ and $b$ are constants, as yet unknown, which are to be determined on the basis of experiments.

Let the initial conditions be
Let the boundary conditions be

\begin{align}
(5) \quad & u(0, t) = 0 , \\
(6) \quad & Tu_\times(1, t) = f(t) .
\end{align}

Eqs. (1) – (6) may, for example, describe an inhomogeneous string, which is fixed at the end \( x = 0 \), while a force \( f(t) \) is applied perpendicular to the string at \( x = 1 \), and \( T \) is the known tension.

The disturbance at the end \( x = 1 \), \( u(1, t) \), is measured at \( n \) instants of time. On the basis of these observations, we wish to estimate the values of the parameters \( a \) and \( b \), and thus to deduce the inhomogeneity of the medium.

3. LAPLACE TRANSFORMS

In order to reduce the partial differential wave equation to a system of ordinary differential equations, we take Laplace transforms of both sides of (2.1). We denote transforms by capital letters, for example,

\begin{align}
(1) \quad & U_s(x) = U(x,s) = L\{u(x,t)\} .
\end{align}

Equation (1) becomes

\begin{align}
(2) \quad & s^2U(x,s) - su(x,0) - u_\times(x,0) = c^2U_{xx} .
\end{align}
Using (2.2) – (2.4), we obtain the desired system of ordinary differential equations,

$$ (a + bx)U_{xx} = s^2U(x,s) - sg(x) - v(x), $$

in which $s$ is a parameter, $s = 1, 2, \ldots, N$. The boundary conditions are

$$ U(0,s) = 0, \quad TU_x(1,s) = F(s). $$

The unknown constants $a$ and $b$ are to be determined by minimizing the expression

$$ \sum_{s=1}^{N} [U_{\text{obs}}(1,s) - U(1,s)]^2. $$

The quantities $U_{\text{obs}}(1,s)$ are the Laplace transforms of the experimentally observed values $u(1,t_i)$, while the quantities $U(1,s)$ are the solutions of equations (3) and (4). The use of Gaussian quadrature [6] leads to the approximate formula for the Laplace transform of the observations,

$$ U_{\text{obs}}(1,s) \approx \sum_{i=1}^{N} r_i^{s-1} u(1,t_i)w_i, \quad s = 1, 2, \ldots, N. $$

Similarly, the transform of the force may be produced with the use of the formula

$$ F(s) \approx \sum_{i=1}^{N} r_i^{s-1} f(t_i)w_i, \quad s = 1, 2, \ldots, N. $$

In these equations, $r_i$ are the roots of the shifted Legendre polynomial $P_N^*(x) = P_N(1 - 2x)$ and $w_i$ are
the related weights. In addition, the times of evaluation are

\begin{equation}
    t_i = -\log e r_i, \ i = 1, 2, \ldots, N.
\end{equation}

Interpolation may be necessary in order to have the data for these special times. After the solution has been found for \( U(x,s) \), the inverse transforms \( u(x,t) \) may be obtained by a numerical inversion method [7].

4. FORMULATION

The constants \( a \) and \( b \) are to be thought of as functions of \( x \) which satisfy the differential equations

\[ a_x = 0, \ b_x = 0. \]

The complete system of equations for this nonlinear boundary value problem is

\[ U_{xx} = \frac{1}{a+bx} \left[ s^2 U(x,s) - s g(x) - v(x) \right], \ s = 1, 2, \ldots, N, \]

\[ a_x = 0, \]

\[ b_x = 0. \]

This is equivalent to a system of \( 2N + 2 \) first order equations, so there must be \( 2N + 2 \) boundary conditions. These conditions are

\begin{align}
(2) \quad &U(0,s) = 0, \ s = 1, 2, \ldots, N, \\
(3) \quad &U_x(i,s) = \frac{F(s)}{T}, \ s = 1, 2, \ldots, N.
\end{align}
5. SOLUTION VIA QUASILINEARIZATION

The nonlinear boundary value problem may be resolved using the technique of quasilinearization [8, 9, 10]. In each step of the successive approximation method, we must solve the linear differential equations

\[
\frac{dU^n}{dx} = W^n,
\]

\[
\frac{dW^n}{dx} = \frac{s^2U^n}{a+bx} - \frac{a^2b^n}{(a+bx)^2} s^2U^n + \frac{sU^n}{a+bx},
\]

(1)

\[
\frac{da^n}{dx} = 0,
\]

\[
\frac{db^n}{dx} = 0,
\]

where the superscripts \( n \) indicate the solution in the \( n \)th approximation, while the un-superscripted variables belong to the \((n-1)\)st approximation. The boundary conditions are

\[
U^n_s(0) = 0,
\]

(2)

\[
W^n_s(1) = \frac{F(s)}{T},
\]

(3)
We represent the solution in the \( n \)th approximation as a linear combination of a particular vector solution and \( N + 2 \) homogeneous vector solutions. If we let the column vector \( X(x) \) represent the solution in the \( n \)th approximation, where the components of \( X \) are \((u_1^n, u_2^n, \ldots, u_N^n, w_1^n, w_2^n, \ldots, w_N^n, a^n, b^n)\), and if we let the column vectors \( P(x), H_1(x), H_2(x), \ldots, H_{N+2}(x) \) represent the particular and homogeneous solutions, then we may write

\[
X(x) = P(x) + \sum_{i=1}^{N+2} H_i(x) y_i .
\]

Since the system of differential equations is of order \( 2N + 2 \), and since \( N \) initial conditions are prescribed, there are \( N + 2 \) missing initial conditions, represented by the \( N + 2 \) dimensional column vector \( Y \),

\[
Y = (w_1^n(0), w_2^n(0), \ldots, w_N^n(0), a^n(0), b^n(0))^T .
\]

The particular and homogeneous solutions are computationally produced. In terms of these, the boundary conditions (2) – (5) require the solution of system of \( N + 2 \) linear algebraic equations,
(8) \[ A Y = B , \]

where the elements of matrix \( A \) are

\[ A_{ij} = H_{N+1}^i(1) , \quad i = 1, 2, \ldots, N , \]

\[ (9) \]

\[ = \sum_{s=1}^{N} H_{s}^i(1) H_{s}^j(1) , \quad i = N + 1, N + 2 , j = 1, 2, \ldots, N, N + 1, N + 2 , \]

and where the components of vector \( B \) are

\[ B_i = \frac{F(i)}{T} - P_{N+1}^{(1)} , \quad i = 1, 2, \ldots, N , \]

\[ = \sum_{s=1}^{N} [U_{0bs}(1,s) - P_s(1)] H_{s}^i(1) , \quad i = N + 1, N + 2 . \]

The method is applied iteratively for a fixed number of stages, about five, or it may be terminated when the approximations converge or diverge. The displacement function \( u(x,t) \) may be obtained from its transform by a numerical inversion method of Bellman, et al. [7].

6. EXAMPLE 1 – HOMOGENEOUS MEDIUM. STEP FUNCTION FORCE

In this and the following example, we consider a homogeneous medium and make use of the analytical solution. In Example 3, we consider the more general problem of an inhomogeneous medium characterized by two unknown constants.

Consider the case in which we have a constant speed \( c \) which is given by the equation
(1) \[ c^2 = a = 1. \]

The value of \( T \) is unity, the input \( f(t) \) is the Heaviside unit step function, \( H(t) \), and the initial conditions are \( g(x) = v(x) = 0 \). The wave equation for the function \( U(x,s) \) is

(2) \[ U_{xx} = \frac{s^2}{c^2} U(x,s). \]

The solution which satisfies (2), as well as the boundary conditions \( U(0,s) = 0 \), \( TU_x(1,s) = F(s) \) is

(3) \[ U(x,s) = \frac{c F(s) \sinh \frac{sx}{c}}{T s \cosh \frac{s}{c}}. \]

Noting that the Laplace transform of the force is

(4) \[ F(s) = L\{H(t)\} = \frac{1}{s}, \]

we may explicitly evaluate \( U \) at the boundary \( x = 1 \), and we obtain the values

(5) \[ U(1,s) = \frac{c}{T} \frac{1}{s^2} \tanh \frac{s}{c} = \frac{1}{s} \tanh s. \]

The inverse transform,

(6) \[ u(1,t) = L^{-1}\left\{ \frac{c}{T} \frac{1}{s^2} \tanh \frac{s}{c} \right\}, \]

is shown in Fig. 1.

We decide to use a seven point quadrature, so that \( N = 7 \). Making use of the known solution, we "produce" the
observations at the specified times $t_i$, which are listed in Table 1.

![Graph showing the analytical solution of the wave equation.](image)

Fig. 1. The analytical solution of the wave equation at $x = 1$, with a step function input:

$$u(1,t) = \exp^{-\frac{t}{T}} \tanh \left( \frac{s}{c} \right).$$

### TABLE 1

SEVEN OBSERVATIONS FOR EXAMPLE 1

<table>
<thead>
<tr>
<th>$t_i$</th>
<th>$u(1,t_i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.671195</td>
<td>0.328805</td>
</tr>
<tr>
<td>2.046127</td>
<td>1.953873</td>
</tr>
<tr>
<td>1.213762</td>
<td>1.213762</td>
</tr>
<tr>
<td>0.693147</td>
<td>0.693147</td>
</tr>
<tr>
<td>0.352509</td>
<td>0.352509</td>
</tr>
<tr>
<td>0.138382</td>
<td>0.138382</td>
</tr>
<tr>
<td>0.025775</td>
<td>0.025775</td>
</tr>
</tbody>
</table>
The approximate transforms \( U_{0bs} \) are computed using the formula from Gaussian quadrature. In Table 2, these quantities are compared with the exact transforms using the analytical solution. The transforms of the input, \( F(s) \), are computed with the aid of the approximate formula. The approximate transforms, \( U_{0bs}(1,s) \) and \( F(s) \), are used in the calculations because in the general case, the analytical transforms will be unobtainable.

### TABLE 2

<table>
<thead>
<tr>
<th>( s )</th>
<th>Approximate ( U_{0bs}(1,s) )</th>
<th>Exact ( U_{0bs}(1,s) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.759442</td>
<td>0.761594</td>
</tr>
<tr>
<td>2</td>
<td>0.242907</td>
<td>0.241007</td>
</tr>
<tr>
<td>3</td>
<td>0.110753</td>
<td>0.110561</td>
</tr>
<tr>
<td>4</td>
<td>0.0624686</td>
<td>0.064580</td>
</tr>
<tr>
<td>5</td>
<td>0.0399963</td>
<td>0.0399964</td>
</tr>
<tr>
<td>6</td>
<td>0.0277773</td>
<td>0.0277774</td>
</tr>
<tr>
<td>7</td>
<td>0.0204081</td>
<td>0.0204081</td>
</tr>
</tbody>
</table>

There are only \( 2N + 1 \) variables in this example, so that when \( N = 7 \), we have a solution of dimension 15. During each stage of the calculations, we have to produce a particular solution and \( N + 1 = 8 \) homogeneous solutions. i.e., \( 15 + 8 = 135 \) differential equations must be integrated. For the initial conditions on \( P \), we choose \( P(0) \) identically zero. We also
choose for $H^j(0)$, the unit vector which has all of its components zero except the $(N+j)^{th}$, which is unity. Any linear combination of these $P$ and $H$ vectors identically satisfies the conditions $U_s(0) = 0$, $s = 1, 2, \ldots, N$.

For the remaining boundary conditions, we must invert the $8 \times 8$ matrix $A$.

As a first check case, we try an initial approximation $a_0^0 = 1$ which is the correct value of $a$. We estimate the initial slopes to be $W_s(0) = 10^{-3}$. The initial approximation is generated by integrating the nonlinear equations (31) with this set of estimates, as initial conditions. In three iterations we obtain better estimates of the slopes $W_s(0)$, but the value of $a$ has drifted to 1.00023. This value may be used as a comparison for other trials. The results of three experiments are shown in the following table. The initial approximation $a_0^0$ is listed in Table 3, followed by the successive approximations $a_n^N$, $n = 1, 2, \ldots$, for each of the three trials.
TABLE 3
SUCCESSIVE APPROXIMATIONS OF THE VELOCITY a IN EXAMPLE 1

<table>
<thead>
<tr>
<th>Approximation</th>
<th>Run 1</th>
<th>Run 2</th>
<th>Run 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.2</td>
<td>1.5</td>
<td>0.5</td>
</tr>
<tr>
<td>1</td>
<td>1.00991</td>
<td>0.46752</td>
<td>0.50922</td>
</tr>
<tr>
<td>2</td>
<td>1.00186</td>
<td>0.48284</td>
<td>0.80612</td>
</tr>
<tr>
<td>3</td>
<td>1.00018</td>
<td>0.67047</td>
<td>0.97736</td>
</tr>
<tr>
<td>4</td>
<td>------</td>
<td>0.89366</td>
<td>1.00049</td>
</tr>
<tr>
<td>5</td>
<td>------</td>
<td>0.99110</td>
<td>1.00022</td>
</tr>
<tr>
<td>6</td>
<td>------</td>
<td>1.00041</td>
<td>------</td>
</tr>
</tbody>
</table>

In Run 2, \( U \) and \( U_x \) at \( x = 1.0 \) are consistent to two significant figures with the conditions. In Run 3 \( U \) is in agreement with the observations to four places, and \( U_x \) agrees with the conditions to five figures. Recall that the conditions on \( U_x \) are supposed to be exact, and those on the \( U(1,s) \) are of a least squares nature, which may help to explain why \( U_x \) is in better agreement than \( U \) for Run 3.

7. EXAMPLE 2 - HOMOGENEOUS MEDIUM, DELTA-FUNCTION FORCE

In Example 2, we have a homogeneous medium and zero initial conditions. The boundary conditions are again
\[ u(0,t) = 0, \ u_x(1,t) = f(t), \] where now the input is \( f(t) = \delta(t) \), the delta function. The Laplace transform of the delta function is \( F(s) = 1 \). The analytical solution for \( x = 1 \) is
(1) \( u(l,t) = L^{-1} \left\{ \frac{c}{T} \frac{1}{s} \tanh \frac{s}{c} \right\} . \)

This function is sketched in Fig. 2, for the case \( c = 1, \)
\( T = 1. \)

![Graph](image)

Fig. 2. The analytical solution of the wave equation
at \( x = 1, \) with a delta function input:
\( u(l,t) = L^{-1} \left\{ \frac{1}{s} \tanh s \right\} . \)

We again take \( N = 7. \) The observations are

(1) \( u(l,t_i) = 1, \) for \( i = 1, 2, \ldots, 5, \)
\( = -1, \) for \( i = 6, 7. \)

The transforms of the observations, \( U_{obs}(l,s), \) are computed
using the quadrature approximation. A comparison of these
values against the exact transforms using (1) is given in
Table 4.
TABLE 4
THE LAPLACE TRANSFORMS $U_{0bs}(1,s)$ for EXAMPLE 2

<table>
<thead>
<tr>
<th>s</th>
<th>Approximate $U_{0bs}(1,s)$</th>
<th>Exact $U_{0bs}(1,s)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.59080964</td>
<td>0.76159415</td>
</tr>
<tr>
<td>2</td>
<td>0.46055756</td>
<td>0.48201379</td>
</tr>
<tr>
<td>3</td>
<td>0.32857798</td>
<td>0.33168492</td>
</tr>
<tr>
<td>4</td>
<td>0.24939415</td>
<td>0.24983232</td>
</tr>
<tr>
<td>5</td>
<td>0.19992192</td>
<td>0.19998184</td>
</tr>
<tr>
<td>6</td>
<td>0.16665658</td>
<td>0.16666462</td>
</tr>
<tr>
<td>7</td>
<td>0.14285584</td>
<td>0.14285690</td>
</tr>
</tbody>
</table>

All initial approximations in the following experiments are produced by integrating the nonlinear equations with a set of estimated initial conditions. The check case with initial approximation $a^0 = 1$, a correct guess, results in a convergence to the wrong value $a \approx 0.9$. With $a^0 = 0.5$, the estimate is again 0.9. With $a^0 = 1.5$, the value -0.8 is obtained. It is suspected that the discontinuous nature of the function $u(1,t)$ is the cause of the difficulty in determining $a$. A more reasonable formulation of the problem should include damping terms to overcome this obstacle. In spite of the poor estimates of $a$ in the first two trials, the final approximations are quite close to the exact observations $U_{0bs}(1,s)$, rather than the approximate, and the conditions $U_x(1,s) = F(s)/T$ are met, to within 0.001%.
8. EXAMPLE 3 — INHOMOGENEOUS MEDIUM WITH DELTA-FUNCTION INPUT

As an example of the inverse problem for an inhomogeneous medium as originally posed, consider the case in which the wave velocity is indeed given by the equation

\[ c^2 = a + bx, \]

where \( a = 1 \) and \( b = 0.5 \). We again set the initial conditions \( u(x,0) = u_t(x,0) = 0 \), and the tension \( T = 1 \). We exert a delta-function force, \( f(t) = \delta(t) \), on the boundary \( x = 1 \), and we observe the displacement \( u(1,t) \) as a function of time. Laplace transforms \( U_{obs}(1,s) \) are computed. The parameters \( a \) and \( b \) are determined for best agreement with these transforms of observations.

In this study, the experimenter obtains his data with the use of the digital computer, rather than by the actual performance of laboratory experiments. The exact solution for this inhomogeneous wave problem is not readily available analytically. We must produce the solution computationally, by solving the wave equation with its boundary conditions. Since we prefer to deal with the ordinary differential equation for the function \( U_s(x) \), we solve the approximately equivalent linear two-point boundary value problem

\[ U_{xx} = \frac{1}{a+bx} s^2 U(x,s), \]
(3) \[ U(0,s) = 0 , \]

(4) \[ U_x(1,s) = 1 , \]

for \( s = 1, 2, \ldots, N \). We produce a particular solution and \( N \) independent homogeneous solutions which, when combined to satisfy conditions (3) and (4), also produce the data of Table 5. These are the "observations".

**TABLE 5**

**THE LAPLACE TRANSFORMS \( U_{\text{obs}}(1,s) \) FOR EXAMPLE 3**

<table>
<thead>
<tr>
<th>( s )</th>
<th>( U_{\text{obs}}(1,s) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.811967</td>
</tr>
<tr>
<td>2</td>
<td>.551174</td>
</tr>
<tr>
<td>3</td>
<td>.390695</td>
</tr>
<tr>
<td>4</td>
<td>.297835</td>
</tr>
<tr>
<td>5</td>
<td>.239837</td>
</tr>
<tr>
<td>6</td>
<td>.200613</td>
</tr>
<tr>
<td>7</td>
<td>.172392</td>
</tr>
</tbody>
</table>

These quantities \( U_{\text{obs}}(1,s) \) can be inverted numerically to produce the function \( u(1,t) = L^{-1} \{ U_{\text{obs}}(1,s) \} \), which are the observations of the disturbance in the space of \( x \) and \( t \). However, we need the set of transforms for use in determining the parameters \( a \) and \( b \), and so we decide to utilize these numbers directly, as they appear in the table.
Two series of experiments are performed (see Tables 6 and 7). In one, the observations are given correct to 6 significant figures, and the initial approximations are varied. The true values of the unknown parameters are \( a = 1.0 \) and \( b = 0.5 \).

**TABLE 6**

SERIES I RESULTS FOR EXAMPLE 3

Observations are correct to six significant figures.

<table>
<thead>
<tr>
<th>Run</th>
<th>( a^0 = 0.9 )</th>
<th>( a^3 = 0.9998 )</th>
<th>( b^0 = 0.6 )</th>
<th>( b^3 = 0.5002 )</th>
<th>( W_s(0) ) correct to 1 figure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Run 2</td>
<td>( a^0 = 1.2 )</td>
<td>( a^3 = 0.9998 )</td>
<td>( b^0 = 0.3 )</td>
<td>( b^3 = 0.5002 )</td>
<td>( W_s(0) ) correct to 1 figure</td>
</tr>
<tr>
<td>Run 3</td>
<td>( a^0 = 1.2 )</td>
<td>( a^3 = 0.9996 )</td>
<td>( b^0 = 0.3 )</td>
<td>( b^3 = 0.50005 )</td>
<td>( W_s(0) = 0.05 )</td>
</tr>
</tbody>
</table>

**TABLE 7**

SERIES II RESULTS FOR EXAMPLE 3

Observations are in error by specified amounts

<table>
<thead>
<tr>
<th>Run 4</th>
<th>( a^0 = 1.2 )</th>
<th>( a^3 = 0.872 )</th>
<th>( b^0 = 0.3 )</th>
<th>( b^3 = 0.5182 )</th>
<th>( W_s(0) ) correct to 1 figure; Observations: +1% error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run 6</td>
<td>( a^0 = 1.2 )</td>
<td>( a^3 = 0.937 )</td>
<td>( b^0 = 0.3 )</td>
<td>( b^3 = 0.590 )</td>
<td>( W_s(0) ) correct to 1 figure; Observations: +5% error</td>
</tr>
</tbody>
</table>
In the series I experiments, with accurate observations, rapid convergence to the correct values of the parameters occurs. The higher approximations of the initial slopes $W_s(0)$ are not listed, but these are considerably improved values.

In the series II experiments, noisy observations are used. For example in Run 4, the observations $U_{\text{obs}}(1,s)$ are in error by the relative amounts $+1\%, -1\%, +1\%, \ldots, +1\%$ for $s = 1, 2, 3, \ldots, 7$ respectively. The relative errors in the third approximations $a^3 = .9872, b^3 = .5182$, are $1.3\%$ and $3.6\%$ respectively. The results of this trial may be contrasted with the final approximations of Run 2. In Run 2, observations which are correct to six significant figures produce values of the parameters which are correct to less than $0.04\%$. Run 4 may also be compared with Run 6, in which case we are comparing the effect of $1\%$ errors against $5\%$ errors. The results of Run 6 involve errors of $-6\%$ in the value of $a$, and $+18\%$ in $b$.

The time required for these calculations is about one-half minute per iteration, with the IBM 7044. Each iteration includes the integration of $(N+3)(2N+2) = 10 \times 16 = 160$ differential equations, and the inversion of a $9 \times 9$ matrix. The FORTRAN programs for all of the cases treated are to be found in Appendix F.
9. DISCUSSION

The methods presented here are of practical use in identifying a system described by a wave equation or by linear differential equations or by a weighting function [11].
REFERENCES


CHAPTER SEVEN

INVERSE PROBLEMS IN WAVE PROPAGATION:
MEASUREMENTS OF STEADY STATES

1. INTRODUCTION

Consider the propagation of waves in a plane parallel stratified medium [1-5] extending from \( x = 0 \) to \( x = b \), with index of refraction \( n(x) \) varying continuously throughout the slab. The slab is bounded by a vacuum to the left \( (n_0 = 1) \) and a homogeneous medium with index of refraction \( n_1 \) to the right, as shown in Fig. 1. We assume a lossless dielectric medium in which \( n(x) \) is independent of frequency.

\[
\begin{array}{ccc}
n_0 & n(x) & n_1 \\
\text{Incident wave} & \rightarrow & \\
\text{Reflected wave} & \leftarrow & \\
x=0 & & x=b
\end{array}
\]

Fig. 1. The physical situation
The wave equation is

\[ k^2(x)u_{tt} = u_{xx}, \]

where \( k \) is the wave number. The wave number is related to the index of refraction by the formula

\[ k(x) = \frac{\omega}{c_0} n(x), \]

where \( c_0 \) is the speed of light in a vacuum, which we normalize to unity, and \( \omega \) is the angular frequency. We are interested in solutions of the form

\[ u(x,t) = e^{-i\omega t} u(x), \]

corresponding to the steady-state case where the transients have died down. The function \( u(x) \) satisfies the ordinary differential equation.

\[ u''(x) + \omega^2 k^2(x)u(x) = 0. \]

We shall often neglect the function \( e^{-i\omega t} \) in all of the solutions, and speak of the functions \( u(x) \) as waves.

We conduct a series of experiments in which waves of different frequencies \( \omega_i \) are normally incident in the medium from the left, i.e., the incident wave is \( e^{i(k_0x-\omega_it)} \), or simply \( e^{ik_0x} \). The reflected waves at each frequency are observed. We wish to determine the index of refraction \( n(x) \) throughout the slab on the basis of these measurements.
2. SOME FUNDAMENTAL EQUATIONS [1, 2]

Consider the case of two adjacent homogeneous media, as illustrated in Fig. 2.

(a) Incident

(b) Transmitted

Reflected

Fig. 2. Waves at an interface.

A plane wave of frequency \( \omega \) traveling in medium (a) is incident at the interface \( x = 0 \). Let the wave numbers of medium (a) and medium (b) be \( k_a \) and \( k_b \) respectively.

The incident wave is \( e^{i k_a x - i \omega t} \) and the reflected wave is \( -e^{i k_a x - i \omega t} \), where

\[
(1) \quad r = \frac{k_a - k_b}{k_a + k_b}.
\]

The transmitted wave is \( e^{i k_b x - i \omega t} \), where

\[
(2) \quad t = \frac{2k_a}{k_a + k_b}.
\]
Now consider the case of two interfaces between three homogeneous media, (a), (b), and (c). The interfaces are separated by a distance $\Delta$.

Fig. 3. Waves at two interfaces.

The incident wave is again $e^{ik_a x - i \omega t}$. The wave which is transmitted through $x = 0$, reflected at $x = \Delta$, and transmitted again through $x = 0$ is $ve^{ik_a x - i \omega t}$, where

$$v = \frac{2k_a}{k_a + k_b} \cdot \frac{k_b - k_c}{k_b + k_c} e^{2ik_b \Delta} \cdot \frac{2k_b}{k_b + k_a} + o(\Delta),$$

and $o(\Delta)$ includes the terms proportional to the second and higher powers of $\Delta$. This equation shows how $v$ depends on frequency by means of the exponential factor $e^{2ik_b \Delta} = e^{2i\pi n_b \Delta}$.


Now we turn our attention to the reflection coefficient $r$ as a function of thickness of the medium. We assume that the slab is inhomogeneous and that it extends from $x = z$ to $x = b$. The right boundary $x = b$ is to
be considered fixed, while the left boundary \( x = z \) is variable, as shown in Fig. 4. The incident wave is \( e^{ik(z_\_)(x-z)} \), deleting the time dependent factor \( e^{-i\omega t} \), where \( k(z_\_) = k(z-0) \) is the wave number of the homogeneous medium to the left, and where the expression \( e^{ik(z_\_)(x-z)} \) is used rather than \( e^{ik(z_\_)x} \) in order to normalize the incoming intensity at \( x = z \).

![Diagram of an inhomogeneous medium of thickness \( b-z \).]

Fig. 4. An inhomogeneous medium of thickness \( b-z \).

The reflected wave is \( r(z)e^{-ik(z_\_)(x-z)} \).

Using the technique of invariant imbedding, we relate the reflection coefficient for a slab extending from \( z \) to \( b \) to that for a slab extending from \( z + \Delta \) to \( b \). The reflected wave may be expressed, to terms of order zero and one in \( \Delta \), as arising from three processes:

(a) immediate reflection at \( z \);

(b) transmission through the interface at \( x = z \), reflection at \( z + \Delta \) from the slab \((z + \Delta, b)\), and transmission through \( z \);
(c) transmission through the interface at $z = z$, reflection at $z + \Delta$ from the slab $(z + \Delta, b)$, reflection at $z$, reflection at $z + \Delta$, and finally transmission through $z$.

These three cases are represented in Fig. 5.

![Diagram of three processes in a stratified slab](image)

**Fig. 5. Three processes in a stratified slab.**

The wave which is reflected from the slab $(z, b)$ is

\[(1) \quad r(z)e^{-ik(z_+) (x-z)} = [r_a + r_b + r_c + o(\Delta)]e^{-ik(z_+) (x-z)},\]

where

\[(2) \quad r_a = \frac{k(z_+) - k(z_+\Delta)}{k(z_+) + k(z_+\Delta)},\]

\[(3) \quad r_b = \frac{2k(z_+)}{k(z_+) + k(z_+\Delta)} \cdot z(z+\Delta)e^{2ik(z_+\Delta)} \cdot \frac{2k(z_+\Delta)}{k(z_+\Delta) + k(z_+)},\]
\[ r_c = \frac{2k(z_-)}{k(z_-) + k(z_+\Delta)} \cdot r(z+\Delta)e^{2ik(z_+\Delta)\Delta} \]

\[ \cdot \frac{k(z_+\Delta) - k(z_-)}{k(z_+\Delta) + k(z_-)} \cdot r(z+\Delta)e^{2ik(z_+\Delta)\Delta} \]

\[ \cdot \frac{2k(z_-\Delta)}{k(z_+\Delta) + k(z_-)} , \]

and \( k(z_+\Delta) = k(z+\Delta - 0) \) is the wave number in the region immediately to the left of the interface \( z\Delta \). Simplifying to terms of order \( \Delta \), we have

\[ r(z) = \frac{k(z_-) - k(z_+\Delta)}{[k(z_-) + k(z_+\Delta)]} + \frac{4k(z_-)k(z_+\Delta)}{[k(z_-) + k(z_+\Delta)]^2} \frac{[1 + 2ik(z_+\Delta)\Delta]r(z+\Delta)}{[1 + 4ik(z_+\Delta)\Delta]r^2(z+\Delta)} \]

\[ + O(\Delta) . \]

Making use of the formula for the derivative of \( r \),

\[ \frac{dr}{dz} = \lim_{\Delta \to 0} \frac{r(z+\Delta) - r(z)}{\Delta} , \]

we obtain the Riccati equation

\[ \frac{dr}{dz} = \frac{k'}{2k} - 2ikr - \frac{k'}{2k} r^2 . \]

The "initial" condition reduces to the formula for an interface between two media

\[ r(b) = \frac{k(b-0) - k_1}{k(b-0) + k_1} . \]
In terms of the index of refraction, Eqs. (7) and (8) are

\[
\frac{dr}{dz} = \frac{n'}{2n} - 2i\omega \frac{n'}{2n} r^2 ,
\]

where \( n = n(z) \), and

\[
r(b) = \frac{n(b)-n_1}{n(b)+n_1} .
\]

The reflection coefficient for any inhomogeneous slab in which \( n \) varies as a function of \( x \) may be found by a simple (numerical) integration of (9) with the given initial condition (10). The integration is carried out from the right boundary \( z = b \) to the left boundary \( z = 0 \).

4. PRODUCTION OF OBSERVATIONS

In place of performing laboratory experiments for obtaining reflection data [6, 7], we produce the observations computationally, for \( N \) different frequencies. The incident waves are \( e^{i\omega_j n_0 x} \), and the reflected waves are \( r_j(0)e^{-i\omega_j n_0 x} \), \( j = 1, 2, \ldots, N \). We solve the initial value problems

\[
\frac{dr_j}{dz} = \frac{n'}{2n} - 2i\omega_j r_j - \frac{n'}{2n} r_j^2 ,
\]

\[
r_j(b) = \frac{n(b)-n_1}{n(b)+n_1} , b > z > 0 ,
\]

for the desired coefficients \( r_j(0) \).

Since \( r_j \) is a complex reflection coefficient, we let
\( r_j = R_j + iS_j \),

where \( R_j \) and \( S_j \) are real functions which satisfy the equations

\[
\frac{dR_j}{dz} = \frac{n_j}{2n} + 2\sum_{j} n_j S_j - \frac{n_j}{2n} (R_j^2 - S_j^2),
\]

\[
\frac{dS_j}{dz} = -2\sum_{j} n_j R_j - \frac{n_j}{n} R_j S_j,
\]

\[
R_j(b) = \frac{n(b)-n_1}{n(b)+n_1}, \quad S_j(b) = 0,
\]

for \( j = 1, 2, \ldots, N \).

For the numerical experiment, we take

\[
n(x) = a_1 + a_2(x-1)^2
\]

where \( a_1 = 1, a_2 = 0.5 \). We also choose

\[
b = 1,
\]

\[
N = 3,
\]

\[
w_1 = 2\pi,
\]

\[
w_2 = 4\pi,
\]

\[
w_3 = 6\pi.
\]

We assume that \( n_1 = n(b) \), so that \( R_j(b) = 0 \).

We have chosen to normalize the speed,

\[
c_0 = 3 \times 10^{10} \text{ cm/sec} = \frac{\text{one length unit}}{\text{one time unit}}.
\]
We have chosen

\( b = 1 \) length unit

and we set

\( b = 3 \) cm

\( \approx 1 \) X-band microwave wave length.

Then

\( 1 \) length unit = 3 cm

and

\( 1 \) time unit = \( 10^{-10} \) sec.

To produce \( R_j \) and \( S_j \), the real and imaginary parts of the reflection coefficients, for incident waves of frequencies 10, 20, and 30 kilo megacycles, we integrate Eqs. (4) with initial conditions \( R_j(1) = 0, S_j(1) = 0 \), for \( j = 1, 2, 3 \). We use a step length of \( 0.00^\circ \) and the Adams–Moulton integration scheme. The values \( R_j(0) \), and \( S_j(0) \) are the "observed" reflection coefficients. These are

\[
\begin{align*}
R_1(0) &= 0.13217783 \times 10^{-2}, \\
S_1(0) &= 0.14843017 \times 10^{-1}, \\
R_2(0) &= 0.32313148 \times 10^{-3}, \\
S_2(0) &= 0.95414704 \times 10^{-2}, \\
R_3(0) &= 0.38854984 \times 10^{-3}, \\
S_3(0) &= 0.58976205 \times 10^{-2}.
\end{align*}
\]
5. **DETERMINATION OF REFRACTIVE INDEX**

We consider the inhomogeneous slab extending from $x = 0$ to $x = 1$. We are given observations of the real and imaginary parts of the reflection coefficients, $A_i \approx R_i$, $B_i \approx S_i$, where

\[
\begin{align*}
A_1 &= .132178 \times 10^{-2}, & B_1 &= .148430 \times 10^{-1}, \\
A_2 &= .323131 \times 10^{-3}, & B_2 &= .954147 \times 10^{-2}, \\
A_3 &= -.388550 \times 10^{-3}, & B_3 &= .589762 \times 10^{-2},
\end{align*}
\]

which correspond to frequencies $\omega_1 = 10$, $\omega_2 = 20$, and $\omega_3 = 30$ kilomegacycles [6, 7]. We seek to determine the values of the constants $a$ and $b$ in the equation for the index of refraction as a function of position,

\[
n(x) = a + b(x-1)^2,
\]

in such a manner as to minimize the expression

\[
S = \sum_{i=1}^{3} \left[ (A_i - R_i(0))^2 + (B_i - S_i(0))^2 \right].
\]

The form $S$ is the sum of squares of deviations between the solution of Eqs. (4.4) and (4.5), and the (perhaps inaccurate) observations (1).

The system of nonlinear equations is

\[
R'_j = \frac{n^t}{2n} + 2n_{ij}S_j - \frac{n^t}{2n} (R^2_j - S^2_j)
\]
\[ S'_j = -2n_j R_j - \frac{n'_j}{n} R_j S_j, \quad j = 1, 2, 3, \]

\[ a' = 0, \]
\[ b' = 0, \]

where

\[ n = a + b(x-1)^2, \]

and

\[ n' = 2b(x-1). \]

We obtain a system of linear differential equations by applying quasilinearization [8]. In the following linear equations, so as not to clutter the equations with superscripts indicating the approximations and subscripts indicating the components, we write the variables of the current \( k \)th approximation as \( R, S, a, b \) (also \( n \) and \( n' \)). Corresponding quantities in the previous \((k-1)\)st approximation are \( \rho, \sigma, \alpha, \beta \) (and \( n \) and \( n' \)). The linear equations obtained via quasilinearization are

\[
R' = \frac{n'}{2n} + 2\eta_\omega \sigma - \frac{n'}{2n} (\rho^2 - \sigma^2)
\]

\[ + (R-\rho)(- \frac{n'}{\eta} \rho) + (S-\sigma)(2\eta_\omega + \frac{n'}{\eta} \sigma) \]

\[ + (a-a)\left[ \frac{1}{2} \frac{\partial}{\partial \alpha} \left( \frac{n'}{\eta} \right) \right] + 2\omega \sigma \frac{\partial n}{\partial \alpha} - \frac{1}{2} (\rho^2 - \sigma^2) \frac{\partial}{\partial \alpha} \left( \frac{n'}{\eta} \right) \]

\[ + (b-b)\left[ \frac{1}{2} \frac{\partial}{\partial \beta} \left( \frac{n'}{\eta} \right) \right] + 2\omega \sigma \frac{\partial n}{\partial \beta} - \frac{1}{2} (\rho^2 - \sigma^2) \frac{\partial}{\partial \beta} \left( \frac{n'}{\eta} \right).\]
\[ S' = -2\eta \mu \rho - \frac{\gamma\prime}{\eta} \rho \sigma \]
\[ + (R - \rho)(-2\eta u - \frac{n\prime}{\eta} \sigma) + (S - \sigma)(-\frac{n\prime}{\eta} \rho) \]
\[ + (a - \alpha) [-2\mu \rho \frac{\partial \eta}{\partial \alpha} - \rho \sigma \frac{\partial \alpha}{\partial \alpha} \frac{\eta\prime}{\eta}] \]
\[ + (b - \beta) [-2\mu \rho \frac{\partial \eta}{\partial \beta} - \rho \sigma \frac{\partial \beta}{\partial \beta} \frac{\eta\prime}{\eta}] \]
(9)
\[ \alpha' = 0 \]
(10) \[ \beta' = 0 \]

In these equations, we must make the substitutions

(11) \[ \frac{\partial \eta}{\partial \alpha} = 1, \quad \frac{\partial \eta}{\partial \beta} = -\frac{\eta\prime}{\eta^2} \]

\[ \frac{\partial \eta}{\partial \beta} = (x-1)^2, \quad \frac{\partial \eta}{\partial \beta}(\frac{\eta\prime}{\eta}) = 2^{x-1} - \frac{n\prime}{\eta^2}(x-1)^2. \]

For each iteration of the successive approximation scheme, we produce numerically a particular vector solution \( p(x) \) and two homogeneous vector solutions \( h^1(x) \) and \( h^2(x) \) of the system (7) – (10). We set the components of the reflection coefficients equal to a linear combination of the components of \( p(x) \), \( h^1(x) \), and \( h^2(x) \),

\[ R^k_j = p^j(x) + a h^1_j(x) + b h^2_j(x), \quad j = 1, 2, 3 \]
(12)
\[ S^k_j = p^j_{j+3}(x) + a h^1_{j+3}(x) + b h^2_{j+3}(x), \quad j = 1, 2, 3 \]
\[ a^k = p^\gamma(x) + a h^1_\gamma(x) + b h^2_\gamma(x) = a \]
The multipliers \( a \) and \( b \) are given by the equations
\[
\frac{\partial}{\partial a} \left\{ \frac{3}{i=1} \left[ (A_i - R_i^k(0))^2 + (B_i - S_i^k(0))^2 \right] \right\} = 0 ,
\]
\[
\frac{\partial}{\partial b} \left\{ \frac{3}{i=1} \left[ (A_i - R_i^k(0))^2 + (B_i - S_i^k(0))^2 \right] \right\} = 0 .
\]

After making the substitutions (12), we obtain the values of \( a \) and \( b \) in the current approximation,
\[
a = \frac{(f_1 e_{22} - f_2 e_{12})}{(e_{11} e_{22} - e_{12} e_{21})} ,
\]
\[
b = \frac{(n_{11} f_1 - e_{21} f_1)}{(e_{11} e_{22} - e_{12} e_{21})} ,
\]
where the right hand sides are given in terms of known quantities,
\[
f_i = \frac{3}{i=1} h_i^1(0)(A_i - p_i^1(0)) + \frac{3}{i=1} h_i^1(0)(B_i - p_i^1(0)) ,
\]
\[
e_{ij} = \frac{6}{i=1} h_i^j(0) h_i^j(0) , \quad j = 1, 2 ,
\]
\[
i = 1, 2 .
\]

6. NUMERICAL EXPERIMENTS

Using the given data, and the initial approximation for refractive index \( n(x) = 1.2 + 0.2(x-1)^2 \), we determine the constants \( a \) and \( b \) in the function \( n(x) = a + b(x-1)^2 \) to one part in \( 10^6 \) after five iterations of quasilinearization. The successive approximations of the constants \( a \)
and \( b \) are listed in Table 1, labelled Trial 1, and the approximations of the index of refraction are shown in Fig. 6.

For the next experiment, we use data which are in error by \( \pm 2\% \):

\[
\begin{align*}
A_1 &= 0.134822 \times 10^{-2}, & B_1 &= 0.145461 \times 10^{-1}, \\
A_2 &= 0.316668 \times 10^{-3}, & B_2 &= 0.935364 \times 10^{-2}, \\
A_3 &= -0.396321 \times 10^{-3}, & B_3 &= 0.601557 \times 10^{-2}.
\end{align*}
\]

(1)

After five iterations, the initial approximation being the same as before, the constant \( a \) is found correct to within 0.3\%, and \( b \) is correct to about 3\%. On the other hand, the error in \( n(x) \) ranges from 0.3\% at \( x = 1 \) to only 0.7\% at \( x = 0 \). The results are given in Table 1.

For each trial, the step length of integration is \(-0.0025\), and the integration scheme is Adams–Moulton. The time of calculations is 2 min. 12 sec. on the IBM 7044. The FORTRAN programs are found in Appendix G.
<table>
<thead>
<tr>
<th>Approximation</th>
<th>Trial 1</th>
<th>Trial 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>0</td>
<td>1.2</td>
<td>0.2</td>
</tr>
<tr>
<td>1</td>
<td>0.9570026</td>
<td>0.3901124</td>
</tr>
<tr>
<td>2</td>
<td>1.0231246</td>
<td>0.4476300</td>
</tr>
<tr>
<td>3</td>
<td>1.0024801</td>
<td>0.4955928</td>
</tr>
<tr>
<td>4</td>
<td>0.9999989</td>
<td>0.4999477</td>
</tr>
<tr>
<td>5</td>
<td>0.9999993</td>
<td>0.4999996</td>
</tr>
</tbody>
</table>
7. DISCUSSION

Inverse problems in wave propagation, as well as in particle processes, can be computationally solved. The wave equation, being a partial differential equation, is replaced by a system of ordinary differential equations in one of several ways. In the previous chapter, we used Laplace transform methods. In this chapter, we assumed a solution of the form \( u(x,t) = u(x) e^{-i\omega t} \), and we obtained ordinary differential equations for \( u(x) \). Another Fourier decomposition might be

\[
u(x,t) = \sum_{n=1}^{N} a_n(x) \sin nt,
\]

which results in second order ordinary differential equations for the functions \( a_n(x) \). Another system of ordinary differential equations results when the space derivative is replaced by a finite difference.

\[
\ddot{u}_n(t) \approx \frac{1}{c^2} \left( \frac{u_{n+1}(t) - 2u_n(t) + u_{n-1}(t)}{\Delta^2} \right)
\]

These offer interesting possibilities for further studies.
REFERENCES


CHAPTER EIGHT

DISCUSSION

Inverse problems have stimulated much interest in recent years, since the advent of modern electronic computers. The estimation of the structure of a complicated system, formerly unattainable by analytic means or by the use of a desk calculator, is now possible.

The determination of orbits from observations, is a kind of inverse problem in celestial mechanics, going back to Newton, Laplace, Gauss and others [1,2]. Ambarzumian [3], Borg [4], and others [5-8] considered the problem of determining a linear differential equation of Sturm-Liouville type given a spectrum of eigenvalues. The estimation of scattering potentials from the phase shift has been the concern of investigators in quantum theory [9-17]. Many inverse problems have been considered [18-45], especially in the fields of astrophysics, geophysics and geology. Some computational results have already been obtained for the structure of the earth's atmosphere and crust using actual geophysical data [19, 20, 36]. Some inverse problems fall within the domain of system identification, prediction and
control [46-62], while others may be called design problems [63-66]. The common goal of all inverse problems is to determine the structure of a system which has a desired or observed characteristic output.

Computational procedures for the solution of inverse problems have been few and limited in scope. The methodologies put forth in this thesis may serve to widen the range of inverse problems which can now be solved. We formulate inverse problems as nonlinear boundary value problems, since we possess effective computational methods for solving many classes of nonlinear boundary value problems. These methods include quasilinearization, dynamic programming, invariant imbedding, and various combinations of these [67-69, 62]. A number of modifications of the basic techniques are given in Refs. 67, 70-72, describing more accurate solutions of linear algebraic equations, simultaneous calculations of successive approximations, and automatic evaluations of partial derivatives.

Much remains to be done to build a firm library of computational procedures for the solution of inverse problems. Both new and existing methods should be developed. In particular, system identification via invariant imbedding [51] appears promising.
REFERENCES


APPENDICES

THE FORTRAN PROGRAMS

The library routines mentioned in these Appendices are

**BET**

**INTS/INTM**

**MATINV**

**SIMPLX**
APPENDIX A

PROGRAMS FOR ORBIT DETERMINATION
PROGRAM A.1. PRODUCTION OF OBSERVATIONS

The complete program is listed:

MAIN program
DAUX subroutine

The following library routine is required:
INTS/INTM
$JOB 2890,3BODY*K+160,5+100,1CO*C
$JOB MAP
$IBFTC MAIN REF
COMMON ALPHA,X1,Y1.C(4),T(51)

C
C 3 BODY ORBIT DETERMINATION

C 1 READ(5,100)NPRNT,MPRNT,ALPHA,X1,Y1,DELTA
WRITE(6,90)NPRNT,MPRNT,ALPHA,X1,Y1,DELTA
READ(5,101)(C(I),I=1,4)
WRITE(6,91)(C(I),I=1,4)

C
T(2)=0.0
T(3)=DELTA
DO 2 L=4,7
2 T(L)=C(L-3)
CALL INTS(T,4,2,C,0,0,0,0,0)
THETA=ATAN2(T(6),T(4)-1.0)
SN=SIN(THETA)
CS=COS(THETA)
TN=SN/CS
WRITE(6,92)
WRITE(6,93)T(2),T(4),T(5),T(6),T(7),THETA,TN

C
DO 4 M1=1,MPRNT
DO 3 M2=1,MPRNT
3 CALL INTM
THETA=ATAN2(T(6),T(4)-1.0)
SN=SIN(THETA)
CS=COS(THETA)
TN=SN/CS
4 WRITE(6,93)T(2),T(4),T(5),T(6),T(7),THETA,TN
GO TO 1

C
100 FORMAT(2I12*4E12.8)
101 FORMAT(6E12.8)
93 FORMAT(11H12J,4E20.8)
91 FORMAT(1HC6E20.8)
92 FORMAT(///9X1HT,19X1HX,15X5HDX/0T,19X1HY,15X5HDY/0T,15X5HTHETA
1,13X7HTANGENT//)
93 FORMAT(F10.2,T6E20.5)
END

$IBFTC DAUX REF
SUBROUTINE DAUX
COMMON ALPHA,X1,Y1.C(4),T(51)

C
R=T(4)**2+T(6)**2
R=SGRT(R**3)
R1=(X1-T(4))**2+(Y1-T(6))**2
R1=SGRT(R1**3)
T(8)=T(5)
T(9)=-T(4)/R+ALPHA*(X1-T(4))/R1
T(10)=T(7)
T(11)=-T(6)/R+ALPHA*(Y1-T(6))/R1
RETURN
<table>
<thead>
<tr>
<th>SENTRY</th>
<th>MAIN</th>
<th>10</th>
<th>25</th>
<th>0.2</th>
<th>4.0</th>
<th>1.0</th>
<th>0.01</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
PROGRAM A.2. DETERMINATION OF ORBIT

The complete program is listed:

- MAIN program
- INPUT subroutine
- DAUX subroutine
- FUN1 subroutine
- FUN2 subroutine
- PDR1 subroutine
- PDR2 subroutine
- START subroutine

The following library routines are required:

- INTS/INTM
- MATINV
$I榕FC MAIN    REF
COMMON T(363),NEQ,KMAX,HGRID,NGRID(5),THETA(5),X(4,251),Y(4,251),LPHA,
1 H(5,5,251),P(5,251),A(50,50),B(50,1),X,U,Y,V,NPRNT,MPRNT,DTIME
DIMENSION PIVOT(50),INDEX(50,2),IPIVOT(50)

THREE BODY ORBIT DETERMINATION
1 CALL INPUT
DO 8 I=1,5
THET=THETA(I)
ST=SIN(THET)
CT=COS(THET)
TN=ST/CT
8 PRINT114,THET,TN
2 CALL START
K ITERATIONS
3 DO 19 K=1,KMAX
NEQ=30
4 T(2)=0,0
T(3)=HGRID
DO 5 I=4,363
5 T(I)=0.0
T(5)=1.0
T(12)=1.0
T(19)=1.0
T(26)=1.0
T(33)=1.0
N=1
X=W(I+1)
Y=W(3+1)
6 CALL INTS(T,NEQ,2,0,0,0,0,0)
L=3
DO 7 I=1,5
L=L+1
P(I,N)=T(L)
7 H(J,J+1,N)=T(L)
PRINT49,T(2),((H(J,J+1,N),I=1,5),J=1,5)
INTEGRATE OVER RANGE
10 M1=1,MPRNT
10 M2=1,NPRNT
CALL INTM
N=N+1
X=W(I+N)
Y=W(3+N)
STORE P+S AND H+S
L=3
DO 10 I=1,5
L=L+1
P(I,N)=T(L)
DO 10 J=1,5
L=L+1
10 H(J, I, N) = T(L)
   PRINT49, T(2), ((H(J, I, N), I = 1,5), J = 1,5)
11 CONTINUE
   
   
   C   COMPUTE CONSTANTS
   DO 14 I = 1,5
      N = NGRID(I)
      THET = THETA(I)
      STHET = SIN(THET)
      CTHET = COS(THET)
   DO 13 J = 1,5
      A(I, J) = H(J, I, N) * STHET - H(J, I, N) * CTHET
      B(I, J) = (1.0 - P(I, N)) * STHET + P(3, N) * CTHET
   13 CONTINUE
   PRINT114, (A(I, J), J = 1, 5), B(I, J), I = 1, 5
   CALL MATINV(A, 5, DETERM, PIVOT, INDEX, IPIVOT)
   PRINT114, (B(I, J), I = 1, 5)
   
   C   COMPUTE NEW W'S
   N = 1
   DO 20 I = 1, 4
      W(I, N) = B(I, 1)
      ALPHA = B(5, 1)
      PRINT40, K, ALPHA
      TIME = 0.0
      A1 = ATAN2(W(3, N), W(1, N) - 1.0)
      TN = W(3, N) / (W(1, N) - 1.0)
      PRINT50, TIME, (W(I, 1), I = 1, 4), A1, TN
   18 M1 = 1, NPRNT
   DO 17 M2 = 1, NPRNT
      N = N + 1
      DO 17 I = 1, 5
         W(I, N) = P(I, N)
   17 CONTINUE
   DO 17 J = 1, 5
      W(I, N) = W(I, N) + B(J, 1) * H(J, I, N)
      TIME = TIME + DTIME
      AT = ATAN2(W(3, N), W(1, N) - 1.0)
      TN = W(3, N) / (W(1, N) - 1.0)
   18 CONTINUE
   PRINT50, TIME, (W(I, N), I = 1, 4), AT, TN
   
   C   CONTINUE
   GO TO 1
   
   C   CONTINUE
   GO TO 1

   C   CONTINUE
   GO TO 1

   C   CONTINUE
   GO TO 1

   FORMAT(1H10/4C10, 9HITERATION, 13, 5X7HALPHA = , E18.6//
   1 6X4H 7, 14x1Hx, 19X2HX, 18X1H)
   2 19X2HY, 15X5HANGLE, 13X7HTANGENT)
   
   FORMAT(1HGOF9.2, 5E20.8/(10X5E20.8))
   50 FORMAT(F1C2.6E20.6)
   114 FORMAT(I10, 6E20.6)
   END

$13FTC INPUT  REF
SUBROUTINE INPUT
COMMON T(363), NEQ, KMAX, HGRID, NGRID(5), THETA(5), W(4, 251), ALPHA,
   1 H(5, 5, 25), P(5, 251), A(50, 50), B(50, 1), X, U, Y, V, NPRNT, MPRNT, DTIME
   C   READ110, NPRNT, MPRNT, KMAX
PRINT10,NPRNT,MPRNT,KMAX
READ11,HGRID,ALPHA
PRINT11,HGRID,ALPHA
F=NPRNT
DTIME=F#HGRID
READ12,(NGRID(I),THETA(I),I=1,5)
PRINT20,(NGRID(I),THETA(I),I=1,5)
110 FORMAT(6I12)
10 FORMAT(106I120)
111 FORMAT(6E12.8)
120 FORMAT(106E12.8)
20 FORMAT(120.E20.8)
RETURN
END
$DOTC DAUX REF
SUBROUTINE DAUX
C
C
COMMON T(363),NEQ,KMAX,HGRID,NGRID(5),THETA(5),W(4,251),ALPHA,
1 H(5,5,251),P(5,251),A(5,5,50),B(5,5,1),X,U,Y,V,NPRNT,MPRNT,DTIM,
2 ,IFLAG
DIMENSION XX(2),YY(2),ANS(2),PP(5),HH(5,5),PD(5),HD(5,5),PD1(3),
1 PD2(3),AA(2)
C
GO TO (10,20),IFLAG
C
10 XX(1)=T(4)
XX(2)=0.0
YY(1)=T(6)
YY(2)=0.0
AA(1)=ALPHA
AA(2)=0.0
T(8)=T(5)
CALL FUN1(XX,YY,AA,ANS)
T(9)=ANS(1)
C
T(10)=T(7)
CALL FUN2(XX,YY,AA,ANS)
T(11)=ANS(1)
RETURN
C
20 XX(1)=X
XX(2)=0.0
YY(1)=Y
YY(2)=0.0
AA(1)=ALPHA
AA(2)=0.0
L=3
DO 1 I=1,5
L=L+1
PP(I)=T(L)
DO 1 J=1,5
L=L+1
HH(J,I)=T(L)
C
DX/DT
CALL FUN1(XX,YY,AA,ANS)
CALL PDR1(XX,YY,AA,PD1)
PD(1) = PP(2)
PD(2) = ANS(1) + (PP(1) - X)*PD1(1) + (PP(3) - Y)*PD1(2)
1 + (PP(5) - ALPHA)*PD1(3)
DO 2 J = 1,5
HD(J,1) = HH(J,2)
2 HD(J,2) = HH(J,1)*PD1(1) + HH(J,3)*PD1(2) + HH(J,5)*PD1(3)
C
CALL FUN2(XX,YY,AA,ANS)
CALL PDR2(XX,YY,AA,PD2)
PD(3) = PP(4)
PD(4) = ANS(1) + (PP(1) - X)*PD2(1) + (PP(3) - Y)*PD2(2)
1 + (PP(5) - ALPHA)*PD2(3)
DO 3 J = 1,5
HD(J,3) = HH(J,4)
3 HD(J,4) = HH(J,1)*PD2(1) + HH(J,3)*PD2(2) + HH(J,5)*PD2(3).
C
PD(5) = 0.0
DO 5 J = 1,5
5 HD(J,5) = 0.0
C
DO 4 I = 1,5
L = L + 1
T(L) = PD(1)
DO 4 J = 1,5
L = L + 1
4 T(L) = HD(J,1)
RETURN
END

SUBROUTINE FUN1(XX,YY,AA,ANS)
DIMENSION XX(2), YY(2), AA(2), ANS(2)
X = XX(1)
Y = YY(1)
A = AA(1)
R13 = (X**2 + Y**2)**1.5
R23 = (X - 4.0)**2 + (Y - 1.0)**2)**1.5
ANS(1) = -X/R13 - A*(X - 4.0)/R23
RETURN
END

SUBROUTINE FUN2(XX,YY,AA,ANS)
DIMENSION XX(2), YY(2), AA(2), ANS(2)
X = XX(1)
Y = YY(1)
A = AA(1)
R13 = (X**2 + Y**2)**1.5
R23 = (X - 4.0)**2 + (Y - 1.0)**2)**1.5
ANS(1) = -Y/R13 - A*(Y - 1.0)/R23
RETURN
END

SUBROUTINE PDR1(XX,YY,AA,PD1)
DIMENSION XX(2), YY(2), AA(2), PD1(3)
RR = (X-4.0)**2 + (Y-1.0)**2
R25 = RR**2.5
X = XX(1)
Y = YY(1)
R13 = PR**1.5
A = AA(1)
R15 = RR**2.5
RR = (X-4.0)**2 + (Y-1.0)**2
R23 = RR**1.5
R25 = RR**2.5
PD1(1) = -1.0/R13 + 3.0*X**2/R15 - A/R23 + 3.0*A*(X-4.0)**2/R25
PD1(2) = 3.0*X*Y/R15 + 3.0*A*(X-4.0)*(Y-1.0)/R25
PD1(3) = -(Y-1.0)/R23
RETURN
END

SUBROUTINE POR2

DIMENSION XX(2), YY(2), AA(2), PD2(3)
X = XX(1)
Y = YY(1)
A = AA(1)
RR = X**2 + Y**2
R13 = RR**1.5
R23 = RR**1.5
RR = X**2 + Y**2
R15 = RR**2.5
PD2(1) = 3.0*X*Y/R15 + 3.0*A*(X-4.0)*(Y-1.0)/R25
PD2(2) = -1.0/R13 + 3.0*X**2/R15 - A/R23 + 3.0*A*(Y-1.0)**2/R25
PD2(3) = -(Y-1.0)/R23
RETURN
END

SUBROUTINE START

COMMON T(363), KMAX, HGRID, NGRID(5), THETA(5), W(4, 251), ALPHA(1)
COMMON H(5, 5, 251), P(5, 251), A(50, 50), B(50, 1), X, U, Y, V, NPRNT, MPRNT, DTIME
IFLAG = 1
K = 0
PRINT40, K
N = 1
TIME = 0.0
T(2) = U * 0
T(3) = HGRID
READ1100(T(I), I = 4, 7)
10 FORMAT(6E12.8)
CALL INTS5(I(I+3))
DO 3 I = 1, 4
3 W(I, I) = T(I+3)
PRINT50, TIME, (W(I, N), I = 1, 4)
DO 2 M1 = 1, MPRNT
DO 1 M2 = 1, NPRNT
N = N + 1
CALL INTM
DO 4 I=1,4
  W(I,N)=T(I+3)
1 CONTINUE
  TIME=TIME+TIME
2 PRINT50,TIME,(W(I,N)+1=I,4)
  IFLAG=2
  RETURN
C
40 FORMAT(1H0/65X 9HITERATION*13//
   1 26X4H T,14X1HX,19X2HX',18X1HY,
   2 19X2HY')
50 FORMAT(F30.2,4E20.6)
END
APPENDIX B

PROGRAMS FOR RADIATIVE TRANSFER: LAYERED MEDIA
PROGRAM B.1. DETERMINATION OF \( c \). THE THICKNESS OF THE LOWER LAYER

The complete program is listed:

MAIN program
DAUX subroutine
NONLIN subroutine
PANDH subroutine
LINEAR subroutine
OUTPUT-subroutine
ALBEDO subroutine

The following library routine is required:
INTS/INTM
$IBFTC RTINV

COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRINT,M1MAX,KMAX,DELTA,XTAU,
1 ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
2 P(2,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
3 H2(7,7,3),CONST(3),NEQ

C C PHASE I

1 READ1000,N
PRINT899
PRINT900,N
READ1U01,(RT(I),I=1,N)
PRINT901,(RT(I),I=1,N)
READ1U01,(WT(I),I=1,N)
PRINT901,(WT(I),I=1,N)
DO 2 I=1,N
WR(I)=WT(I)/RT(I)
DO 2 J=1,N
2 AR(I,J)=1.0/RT(I)+1.0/RT(J)

C 899 FORMAT(1H146X36HRADIATIVE TRANSFER - INVERSE PROBLEM /
1000 FORMAT(6E12.8)
900 FORMAT(6E12.8)
1001 FORMAT(6E20.8)
901 FORMAT(6E20.8)
READ1U00,NPRINT,M1,AX,KMAX
PRINT900,NPRINT,M1MAX,KMAX
READ1U01,DELTA
PRINT901,DELTA
READ1U01,XTAU,ZERLAM,XLAM(1),XLAM(2)
PRINT902
PRINT903,XTAU,ZERLAM,XLAM(1),XLAM(2)
902 FORMAT(1H123HPHASE I - TRUE SOLUTION /
903 FORMAT(1H0/

1 1X11HTHICKNESS =, F10.4 /
2 1X11HALBEDO(X) =, 20HA + 8*TANH(10*(X-C)) //
3 1X3HA =, E16.8, 10X3HB =, E16.8, 10X3HC =, E16.8 //)

CALL NONLIN
DO 3 I=1,N
DO 3 J=1,N
3 B2(I,J)=R2(I,J)

C C C

PHASE II

4 READ1U01,XTAU,ZERLAM,XLAM(1),XLAM(2)
K=0
PRINT904,K
PRINT903,XTAU,ZERLAM,Y,XLAM(1),XLAM(2)

C CALL NONLIN

C 904 FORMAT(1H1 13HAPPROXIMATION, 13/

C QUASILINEARIZATION ITERATIONS
C
DO 5 K1=1,KMAX
PRINT904,K1
CALL PANDH
CALL LINEAR
5 CONTINUE
C
READ1000,IGO
GO TO (1,4)IGO
END
SUBFTC DAUX
SUBROUTINE DAUX
DIMENSION V2(7,7),X(3),F(7),G(7)
COMMON N,R(7),WT(7),WR(7),AR(7,7),NPRINT,M1MAX,KMAX,DELTA,XTAU,
1 ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
2 P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
3 H2(7,7,3),CONST(3),NEQ
GO TO (1,2)*IFLAG
C
CNONLINEAR
C
1 L=3
DO 4 I=1,N
DO 3 J=1,I
L=L+1
4 V2(I,J)=T(L)
DO 5 I=1,N
DO 5 J=1,N
5 V2(I,J)=V2(J,I)
L=L+1
VLAM2=T(L)
SIG=T(2)
Y=XTAU*SIG
X(1)=ZERLAM
X(2)=XLAM(1)
X(3)=VLAM2
CALL ALBEDO(Y,X,Z)
ZLAMDA=Z
C
DO 6 I=1,N
F(I)=U.O
DO 7 K=1,N
7 F(I)=F(I) + WR(K)*V2(I,K)
6 F(I)=C.5*f(n+1.0
DO 8 I=1,N
DO 8 J=1,I
L=L+1
8 T(L)=DR
DO 9 I=1,L
L=L+1
9 T(L)=U.O
RETURN
C
C CLINEAR
C
2 SIG=T(2)
    Y=XTAU*SIG
    X(1)=ZERLAM
    X(2)=Xlam(1)
    X(3)=Xlam(2)
    CALL ALBEDO(Y,X,Z)
    ZLAMDA=Z
C
    DO 16 I=1,N
    F(I)=0.0
    DO 17 K=1,N
 17 F(I)=F(I) + WR(K)*R2(I,K)
    16 F(I)=0.5*F(I) + 1.0
C
CP'S
C
L=3
    DO 14 I=1,N
    DO 14 J=1,N
    L=L+1
 14 V2(I,J)=T(L)
    DO 15 I=1,N
    DO 15 J=1,N
 15 V2(I,J)=V2(J,I)
    L=L+1
    VLAM2=T(L)
C
    DO 10 I=1,N
    G(I)=0.0
    DO 10 K=1,N
 10 G(I)=G(I) + (V2(I,K)-R2(I,K))*WR(K)
    ARG=1.0*(Y-Xlam(2))
    XTANX=-10.0*Xlam(1)*(1.0-(TANH(ARG))*M2)
        M=3*NEQ
    DO 12 I=1,N
    DO 12 J=1,N
    FIJ=F(I)*F(J)
    CAPF=-AR(I,J)*R2(I,J) + ZLAMDA*FIJ
    T1=-AR(I,J)*(V2(I,J)-R2(I,J))
    T2=0.5*ZLAMDA*(F(I)*G(J)+F(J)*G(I))
    T3=CAPF
    T4=(VLAM2-Xlam(2))*XTANX*FIJ
        M=M+1
 12 T(M)=T1+T2+T3+T4
    DO 19 I=1,1
        M=M+1
 19 T(M)=0.0
C
CH'S
C
DO 100 K=1,N
  DO 24 I=1,N
    DO 24 J=1,I
      L=L+1
  24  V2(I,J)=T(L)
  DO 25 I=1,N
    DO 25 J=I,N
  25  V2(I,J)=V2(J,I)
    L=L+1
    VLAM2=T(L)
  DO 20 I=1,N
    G(I)=V*0
  DO 20 J=1,N
  20  G(I)=G(I)+V2(I,J)*WR(J)
  DO 22 I=1,N
    DO 22 J=1,I
      FIJ=F(I)+F(J)
      T1=-AR(I,J)*V2(I,J)
      T2=0.5*ZLAMDA*(F(I)*G(J)+F(J)*G(I))
      T3=0.0
      T4=VLAM2*XTANX*FIJ
      N=M+1
  22  T(M)=T1+T2+T3+T4
  DO 29 I=1,1
    M=M+1
  29  T(M)=0.0
  100  CONTINUE
RETURN
END
SIBFTC
NONLIN
SUBROUTINE NONLIN
COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPEN,T,M1MAX,KMAX,DELTA,XTAU,
  1 ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,121),T(1491),SIG,
   2 P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
   3 H2(7,7,3),CONST(3),NEQ
  C  NONLINEAR D.E. FOR TRUE SOLUTION OR FOR INITIAL APPROX.
  C
  IFLAG=1
  T(2)=V*0
  T(3)=DELTA
  V=1
  L1=0
  L3=3
  DO 1 I=1,N
    DO 1 J=1,I
      L1=L1+1
      L3=L3+1
    R2(I,J)=0.0
  1    R(L1+M)=R2(I,J)
  T(L3)=R2(I,J)
  L3=L3+1
2  
T(L3) = XLAM(2)

C

NEQ = (N*(N+1))/2 + 1
CALL INTS(T, NEQ, 2, 0, 0, 0, 0, 0)

C

SIG = T(2)
CALL OUTPUT

C

DO 5 M1 = 1, M1MAX
DO 4 M2 = 1, NPRNT
CALL INTM
M = M + 1
L1 = 0
L3 = 3
DO 3 I = 1, N
DO 3 J = 1, I
L1 = L1 + 1
L3 = L3 + 1
R2(I, J) = T(L3)
3 R(L1, M) = R2(I, J)
4 SIG = T(2)
5 CALL OUTPUT

C

RETURN
END

SUBROUTINE PANDH
COMMON N, RT(7), WT(7), WR(7), AR(7, 7), NPRNT, M1MAX, KMAX, DELTA, XTAU,
1 ZERLAM, XLAM(2), B2(7, 7), R2(7, 7), IFLAG, R(28, 101), T(1491), SIG,
2 P(28, 101), H(28, 3, 101), PTAU, PLAM(2), HTAU(3), HLMAM(2, 3), P2(7, 7),
3 H2(7, 7, 3), CONST(3), NEQ

IFLAG = 2
T(2) = 0.0
T(3) = DELTA
M = 1

C

L1 = 0
L3 = 3
DO 1 I = 1, N
DO 1 J = 1, I
L1 = L1 + 1
L3 = L3 + 1
P(L1, M) = 0.0
1 T(L3) = P(L1, M)
L3 = L3 + 1
PLAM(2) = 0.0
2 T(L3) = PLAM(2)

C

DO 7 K = 1, 1
L1 = 0
DO 3 I = 1, N
DO 3 J = 1, 

L1=L1+1  
L3=L3+1  
H(L1,K,M)=0.0  
3  
T(L3)=H(L1,K,M)  
C  
L3=L3+1  
6  
H(L1,K)=1.0  
7  
T(L3)=H(L1,K)  
C  
L=0  
DO 8  I=1,N  
DO 8  J=1,1  
L=L+1  
5  
R2(I,J)=R(L,M)  
DO 9  I=1,N  
DO 9  J=1,N  
9  
R2(I,J)=R2(J,I)  
C  
NEG=2*((N*(N+1))/2)  
CALL INTS(T,NEG,2,0,0,0,0,0,0)  
LMAX=(N*(N+1))/2  
PRINT52,T(2),(P(L,M),H(L1,M),L=1,LMAX)  
52  
FORMAT(1H0F9.4,5E20.8/(10X5E20.8))  
C  
DO 51  M1=1,MMAX  
DO 55  M2=1,NPRNT  
CALL INTM  
M=M+1  
C PREV. APPROX. R(I,J)  
L1=0  
DO 10  I=1,N  
DO 10  J=1,1  
L1=L1+1  
10  
R2(I,J)=R(L1,M)  
DO 11  I=1,N  
DO 11  J=1,N  
11  
R2(I,J)=R2(J,I)  
L1=0  
L3=3  
DO 12  I=1,N  
DO 12  J=1,1  
L1=L1+1  
L3=L3+1  
12  
P(L1,M)=T(L3)  
L3=L3+1  
DO 13  K=1,1  
L1=C  
DO 14  I=1,N  
DO 14  J=1,1  
L1=L1+1  
L3=L3+1  
14  
H(L1,K,M)=T(L3)  
13  
L3=L3+1  
50  
CONTINUE  
51  
PRINT52,T(2),(P(L,M),H(L1,M),L=1,LMAX)
RETURN
END

SUBROUTINE LINEAR
DIMENSION CHK1(3)
DIMENSION A(49,3),B(49),EMAT(50,5),P(50)
COMMON N,RT(7),WT(7),R(7),AK(7,7),NPRT,M1MAX,KMAX,DELTA,XTAU,
1 ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
2 F(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
3 H2(7,7,3),CONST(3),NEQ

CBOUNDARY CONDITIONS
MLAST=NPRNT*M1MAX + 1
DO 1 K=1,N
L=U
DO 2 I=1,N
DO 2 J=1,N
L=L+1
2 H2(I,J,K)=H(I,J,MLAST)
DO 1 I=1,N
DO 1 J=1,N
1 H2(I,J,K)=H2(J,I,K)
L=0
DO 3 I=1,N
DO 3 J=1,N
L=L+1
3 P2(I,J)=P(L,MLAST)
DO 4 I=1,N
DO 4 J=1,N
4 P2(I,J)=P2(J,I)

CLEAST SQUARES
DO 5 K=1,N
L=U
DO 5 I=1,N
DO 5 J=1,N
L=L+1
5 A(L,K)=H2(I,J,K)
L=0
DO 6 I=1,N
DO 6 J=1,N
L=L+1
6 B(L)=B2(I,J) - P2(I,J)

CO
LMAX=N**2
PRINT60
60 FORMAT(1HO)
DO 61 L=1,LMAX
61 PRINT82,(A(L,K),K=1,1),B(L)
CO
DO 8 I=1,N
DO 7 J=1,N
SUM=0.0
DO 9 L=1,LMAX
9 SUM=SUM + A(L,I)*A(L,J)
7 EMAT(I,J)=SUM
SUM=0
DO 10 L=1,LMAX
10 SUM=SUM + A(L,1) * B(L)
FVEC(1,1)=SUM

PRINT60
DO 81 I=1,1
81 PRINT82,(EMAT(I,J),J=1,1)*FVEC(1,1)
82 FORMAT(10X6E2.8)
FVEC(1,1)=FVEC(1,1)/EMAT(1,1)

DO 11 I=1,1
11 CONST(I)=FVEC(I,1)

XLAM(2)=CONST(1)
PRINT903,XTAU,ZERLAM,XLAM(1),XLAM(2)
903 FORMAT(1HO/
  1  1X11HTHICKNESS =, F10.4 /
  2  1X11HALBESO(X) =, 20HA + 6*TANH(10*(X-C)) //
  3  1X3HA =, E16.8, 10X3HB =, E16.8, 10X3HC =, E16.8 //)

CNEW APPROXIMATION

M=1
L=0
DO 12 I=1,N
DO 12 J=1,1
L=L+1
SUM=P(L,M)
DO 13 K=1,1
13 SUM=SUM + CONST(K)*H(L,K,M)
12 R(L,M)=SUM
L=0
DO 14 I=1,N
DO 14 J=1,1
L=L+1
14 R2(I,J)=R(L,M)
SIG=0.0
CALL OUTPUT

DO 5 M1=1,M1MAX
DO 18 M2=1,NPRNT
M=M+1
L=0
DO 15 I=1,N
DO 15 J=1,1
L=L+1
SUM=P(L,M)
DO 16 K=1,1
16 SUM=SUM + CONST(K)*H(L,K,M)
15 R(L,M)=SUM
L=0
DO 17 I=1,N
DO 17 J=1,1
L = L + 1
17 R2(I,J) = R(L,M)
18 SIG = SIG + DELTA
50 CALL OUTPUT

C
RETURN
END

$IBFC OUTPUT
SUBROUTINE OUTPUT
DIMENSION X(3)
COMMON N, RT(7), WT(7), WR(7), AR(7, 7), NPRNT, M1MAX, KMAX, DELTA, XTAU,
1 ZERLAM, XLAM(2), B2(7, 7), R2(7, 7), IFLAG, R(28, 101), T(1491), SIG,
2 P(28, 101), H(28, 3, 101), PTAU, PLAM(2), HTAU(3), HLAM(2, 3), P2(7, 7),
3 H2(7, 7, 3), CONST(3), NEQ

DO 1 I = 1, N
DO 1 J = 1, N
1 R2(I, J) = R2(J, I)
Y = XTAU * SIG
X(1) = ZERLAM
X(2) = XLAM(1)
X(3) = XLAM(2)
CALL ALBEDO(Y, X, Z)
PRINT 100, SIG, Y, Z

100 FORMAT (1HO, 7H SIGMA =, F6.2, 4X, 5HTAU =, F6.2, 4X, 8HALBEDO =, F6.2)

DO 2 J = 1, N
2 PRINT 101, J, (R2(I, J), I = 1, N)

101 FORMAT (110, 7F10.6)
RETURN
END

$IBFC ALBEDO
SUBROUTINE ALBEDO(Y, X, Z)
DIMENSION X(3)
COMMON N, RT(7), WT(7), WR(7), AR(7, 7), NPRNT, M1MAX, KMAX, DELTA, XTAU,
1 ZERLAM, XLAM(2), B2(7, 7), R2(7, 7), IFLAG, R(28, 101), T(1491), SIG,
2 P(28, 101), H(28, 3, 101), PTAU, PLAM(2), HTAU(3), HLAM(2, 3), P2(7, 7),
3 H2(7, 7, 3), CONST(3), NEQ
ARG = 10.0 * (Y - X(3))
Z = X(1) + X(2) * TANH(ARG)
RETURN
END
PROGRAM B.2. DETERMINATION OF T, THE OVERALL OPTICAL THICKNESS

The complete program is listed:

MAIN program
DAUX subroutine
NONLIN subroutine
PANDH subroutine
LINEAR subroutine
OUTPUT subroutine
ALBEDO subroutine

The following library routine is required:

INTS/INAM
SUBFTC RTINV
COMMON N,RT(7),WT(7),WR(7),AK(7,7),NPRINT,M1MAX,KMAX,DELTA,XTAU-
1 ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG-
2 P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),-
3 H2(7,7,3),COMMON NEQ

C PHASE I

1 READ1V00,N
PRINT899
RINT899,N
READ1V01,(RT(I),I=1,N)
PRINT901,(RT(I),I=1,N)
READ1V01,(WT(I),I=1,N)
PRINT901,(WT(I),I=1,N)
DO 2 I=1,N
WR(I)=WT(I)/RT(I)
DO 2 J=1,N
AR(I,J)=1.0/RT(I)+1.0/RT(J)

C FORMAT(1H146X36HRADIATIVE TRANSFER - INVERSE PROBLEM /)
1000 FORMAT(6I12)
900 FORMAT(6I20)
1001 FORMAT(6E12.8)
901 FORMAT(6E20.8)
READ1000,NPRINT,M1MAX,KMAX
PRINT900,NPRINT,M1MAX,KMAX
READ1V01,DELTA
PRINT901,DELTA
READ1V01,XTAU,ZERLAM,XLAM(1),XLAM(2)
PRINT902
PRINT903,XTAU,ZERLAM,XLAM(1),XLAM(2)
902 FORMAT(1H1H123HPHASE I - TRUE SOLUTION /)
903 FORMAT(1H0/
1 1X11HTHICKNESS =, F10.4 /
2 1X11HALBEDO(X) =, 20HA + 8*TANH(10*(X-C)) //
3 1X3HA =, E16.8, 10X3HB =, E16.8, 10X3HC =, E16.8 //)
CALL NONLIN
DO 3 I=1,N
DO 3 J=1,N
3 B2(I,J)=R2(I,J)

C PHASE II

4 READ1V01,XTAU,ZERLAM,XLAM(1),XLAM(2)
K=0
PRINT904,K
PRINT903,XTAU,ZERLAM,XLAM(1),XLAM(2)

CALL NONLIN

904 FORMAT(1H13HAPPROXIMATION, I3 /)

C QUASILINEARIZATION ITERATIONS
C
DO 5 K1=1,KMAX
PRINT904,K1
CALL PANDH
CALL LINEAR
5 CONTINUE
C
C
READ1000,IGO
GO TO (i,4),IGO
END
SUBROUTINE DAUX
DIMENSION V2(7,7),X(3),F(7),G(7)
COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
1 ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
2 P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
3 H2(7,7,3),CONST(3),NEQ
GO TO (1,2),IFLAG
C
CNONLINEAR
C
1 L=3
DO 4 I=1,N
DO 4 J=1,I
L=L+1
4 V2(I,J)=T(L)
DO 5 I=1,N
DO 5 J=I,N
5 V2(I,J)=V2(J,I)
L=L+1
VTAU=T(L)
SIG=T(2)
Y=VTAU*SIG
X(1)=ZERLAM
X(2)=XLAM(1)
X(3)=XLAM(2)
CALL ALBEDO(Y,X,Z)
ZLAMDA=Z
C
DO 6 I=1,N
F(I)=G
DO 7 K=1,N
6 F(I)=F(I) + WR(K)*V2(I,K)
7 F(I)=F(I) + 1.0
C
DO 8 I=1,N
DO 8 J=1,I
L=L+1
DR=AR(I,J)*V2(I,J) + ZLAMDA*F(I)*F(J)
8 T(L)=DR*VTAU
DO 9 I=1,1
L=L+1
9 T(L)=0.0
RETURN

C
CLINEAR
C
2 SIG=T(2)
   Y=XTAU*SIG
   X(1)=ZERLAM
   X(2)=XLAM(1)
   X(3)=XLAM(2)
   CALL ALBEDO(Y,X,Z)
   ZLAMDA=Z
   DO 16 I=1,N
   F(I)=0.0
   DO 17 K=1,N
   17 F(I)=F(I) + WR(K)*R2(I,K)
   16 F(I)=0.5*F(I) + 1.0
C
CP'S
C
   L=3
   DO 14 I=1,N
   DO 14 J=1,I
   L=L+1
   14 V2(I,J)=T(L)
   DO 15 I=1,N
   DO 15 J=1,N
   15 V2(I,J)=V2(J,I)
   L=L+1
   VTAU=T(L)
   DO 10 I=1,N
   DO 10 K=1,N
   10 G(I)=G(I) + (V2(I,K)-R2(I,K))*WR(K)
   ARG=10.0*(Y-XLAM(2))
   PARTL=10.0*SIG*XLAM(1)*(1.0-(TANH(ARG))**2)
   M=3+NE0
   DO 12 I=1,N
   DO 12 J=1,I
   FIJ=F(I)*F(J)
   CAPF=-AR(I,J)*F(I,J) * ZLAMDA*FIJ
   T1=XTAU*AR(I,J)*(V2(I,J)-R2(I,J))
   T2=XTAU*0.5*ZLAMDA*(G(I)*G(J)+F(I)*G(J))
   T3=XTAU*CAPF
   T4=(VTAU-XTAU)*(CAPF + XTAU*FIJ*PARTL)
   M=M+1
   12 T(M)=T1+T2+T3+T4
   DO 19 I=1,N
   M=M+1
   19 T(M)=0.0
C
CH'S
C
DO 100 K=1,1
   DO 24 I=1,N
   DO 24 J=1,I
       L=L+1
       V2(I,J)=T(L)
   DO 25 J=I,N
       V2(I,J)=V2(J,I)
       L=L+1
       VTAU=T(L)
   DO 20 J=1,N
       G(J)=0.0
   DO 20 J=1,N
       G(J)=G(J)+V2(J)*WR(J)
   DO 22 J=1,N
       DO 22 I=I,N
           F1J=F(I)*F(J)
           CAPF=-AR(I,J)*R2(I,J) + ZLAMDA*FIJ
           T1=-XTAU*(I,J)*V2(I,J)
           T2=XTAU*0.5*ZLAMDA*(F(I)*G(J)+F(J)*G(I))
           T3=0.0
           T4=VTAU*(CAPF + XTAU*FIJ*PARTL)
   M=M+1
   T(M)=T1+T2+T3+T4
   CONTINUE
   RETURN
END

SUBROUTINE NONLIN
COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,XMAX,DELTA,XTAU,
1 ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
2 P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
3 H2(7,7,3),CONST(3),NEQ

IFLAG=1
T(2)=U*0
T(3)=DELTA
M=1
L1=0
L3=3
DO 1 I=1,N
   DO 1 J=1,N
       L1=L1+1
       L3=L3+1
       R2(I,J)=0.0
       R(L1,M)=R2(I,J)
   T(L3)=R2(I,J)
L3 = L3 + 1

T(L3) = XTAU

NEQ = (N*(N+1))/2 + 1
CALL INTS(T,NEQ,2,0,0,0,0,0)

SIG = T(2)
CALL OUTPUT

DO 5 M1 = 1, M1MAX
DO 4 M2 = 1, NPRNT
CALL INTM
M = M + 1
L1 = 0
L3 = 3
DO 3 I = 1, N
DO 3 J = 1, I
L1 = L1 + 1
L3 = L3 + 1
R2(I, J) = T(L3)
R(L1, M) = R2(I, J)
SIG = T(2)
5 CALL OUTPUT

RETURN
END

$IGFTC LINEAR
SUBROUTINE LINEAR
DIMENSION CHKI(3)
DIMENSION A(49,3), B(49), EMAT(50,50), PIVOT(50), INDEX(50,2)
1*1 PIVOT(50) FVEC(50,1)
COMMON N, RT(7), WT(7), WR(7), AR(7,7), NPRNT, M1MAX, KMAX, DELTA, XTAU,
1 ZERLAM, XLAM(2), B2(7,7), R2(7,7), IFLAG, R(28, 101), T(1491), SIG,
2 P(28, 101), H(28, 3, 101), PTAU, PLAM(2), HTAU(3), HLAM(2, 3), P2(7, 7),
3 H2(7, 7, 3), CONST(3), NEQ

CBoundary CONDITIONS
MLAST = NPRNT*M1MAX + 1
DO 1 K = 1, 1
L = 0
DO 2 I = 1, N
DO 2 J = 1, I
L = L + 1
2 H2(I, J, K) = H(L, K, MLAST)
DO 1 I = 1, N
DO 1 J = 1, N
1 H2(I, J, K) = H2(J, I, K)
L = 0
DO 3 I = 1, N
DO 3 J = 1, I
L = L + 1
3 P2(I, J) = P(L, MLAST)
DO 4 I = 1, N
DO 4 J = 1, N
4 P2(I, J) = P2(J, I)
CLEAST SQUARES
DO 5 K=1,1
L=0
DO 5 I=1,N
DO 5 J=1,N
L=L+1
5 A(L+K)=H2(I,J,K)
L=0
DO 6 I=1,N
DO 6 J=1,N
L=L+1
6 B(L)=B2(I,J) - P2(I,J)
C
LMAX=N**2
PRINT60
60 FORMAT(1HO)
DO 61 L=1,LMAX
61 PRINT82,(A(L+K),K=1,1),B(L)
C
DO 8 I=1,1
DO 7 J=1,1
SUM=0.0
DO 9 L=1,LMAX
9 SUM=SUM + A(L,I)*A(L,J)
7 EMAT(I,J)=SUM
SUM=0.0
DO 10 L=1,LMAX
10 SUM=SUM + A(L,I)*B(L)
8 FVEC(I,1)=SUM
C
PRINT60
DO 81 I=1,1
81 PRINT82,(EMAT(I,J),J=1,1),FVEC(I,1)
82 FORMAT(10X6E20.8)
FVEC(1,1)=FVEC(1,1)/EMAT(1,1)
C
CONST(1)=FVEC(I,1)
C
XTAU =CONST(1)
PRINT903,XTAU,ZERLAM,XLAM(1),XLA::(2)
903 FORMAT(1HO/
1 1X11HTHICKNESS =, E16.8 /
2 1X11HALBEDO(X) =, 20HA + B*TANH(10*(X-C)) //
3 1X3HA =, E16.8, 10X3HC =, E16.8 //)
C
CNEW APPROXIMATION
C
M=1
L=0
DO 12 I=1,N
DO 12 J=1,1
L=L+1
SL:A=P(L,M)
DC 13 K=1,1
13 \text{SUM} = \text{SUM} + \text{CONST(K)} \times \text{H(L,K,M)}
12 R(L,M) = \text{SUM}
L = L + 1
DO 14 I = 1, N
DO 14 J = 1, I
L = L + 1
14 R2(I,J) = R(L,M)
SIG = 0.0
CALL OUTPUT
C
DO 50 M1 = 1, M1MAX
DO 18 M2 = 1, NPRNT
M = M + 1
L = L + 1
DO 15 I = 1, N
DO 15 J = 1, I
L = L + 1
15 \text{SUM} = \text{P(L,M)}
16 \text{SUM} = \text{SUM} + \text{CONST(K)} \times \text{H(L,K,M)}
15 R(L,M) = \text{SUM}
L = L + 1
DO 17 I = 1, N
DO 17 J = 1, I
L = L + 1
17 R2(I,J) = R(L,M)
18 SIG = SIG + DELTA
50 CALL OUTPUT
C
RETURN
END

$SIBFTC PANDH LIST$

SUBROUTINE PANDH
COMMON NRT(7), WT(7), WR(7), AR(7,7), NPRNT, M1MAX, KMAX, DELTA, XTAU,
1 ZERLAM, XLAM(2), B2(7,7), R2(7,7), IFLAG, R(28,101), T(1491), SIG,
2 P(26,101), H(28,3,101), PTAU, PLAM(2), HTAU(3), HLM(2,3), P2(7,7),
3 HZ(7,7,3), CONST(3), NEQ
IFLAG = 2
T(2) = 0.0
T(3) = DELTA
M = 1
C P+S
C
L1 = 0
L3 = 3
DO 1 I = 1, N
DO 1 J = 1, I
L1 = L1 + 1
L3 = L3 + 1
P(L1,M) = 0.0
1 T(L3) = P(L1,M)
L3 = L3 + 1
PTAU = 0.0
2 T(L3) = PTAU
C
C H*S
C
DO 7 K=1,N+1
L1=0
DO 3 I=1,N
DO 7 J=1,I
L1=L1+1
L3=L3+1
H(L1,K+M)=0.0
7 T(L3)=H(L1+K+M)
C
L3=L3+1
6 HTAU(K)=1.0
7 T(L3)=HTAU(K)
C
L=0
DO 8 I=1,N
DO 8 J=1,I
L=L+1
8 R2(I,J)=R(L,M)
DO 9 I=1,N
DO 9 J=1,N
9 R2(I,J)=R2(J,I)
C
NEQ=2*((N*N+1)/2 + 1)
CALL INTS(T*NEQ,2,0,0,0,0,0,0)
LMAX=(N*(N+1))/2
C
DO 51 M1=1,M1MAX
DO 50 M2=1,NPRNT
CALL INTM
M=M+1
C PREV.*APPROX. R(I,J)
L1=0
DO 10 I=1,N
DO 10 J=1,I
L1=L1+1
10 R2(I,J)=R(L1,M)
DO 11 I=1,N
DO 11 J=1,N
11 R2(I,J)=R2(J,I)
L1=0
L3=3
DO 12 I=1,N
DO 12 J=1,I
L1=L1+1
L3=L3+1
12 P(L1,M)=T(L3)
L3=L3+1
DO 13 K=-1,1
L1=0
DO 14 I=1,N
DO 14 J=1,I
L1=L1+1
L3=L3+1
$IBFTC$ OUTPUT

SUBROUTINE OUTPUT
DIMENSION X(3)
COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRINT,M1MAX,KMAX,DELTA,XTAU,
1 ZERLAM,XLAM(2),B(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
2 P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
3 H2(7,7,3),CONST(3),NEQ
DO 1 I=1,N
DO 1 J=1,N
1 R2(I,J)=R2(J,I)
Y=XTAU*SIG
X(1)=ZERLAM
X(2)=XLAM(1)
X(3)=XLAM(2)
CALL ALBEDO(Y,X,Z)
PRINT100, SIG,Y,Z
100 FORMAT(1H0 7HSIGMA =,F6.2, 4X5HTAU =, F6.2, 4X8HALBEDO =,F6.2/
DO 2 J=1,N
2 PRINT101,J,(R2(I,J),I=1,N)
101 FORMAT(1I0, 7F10.6)
RETURN
END$IBFTC$

ALBEDO

SUBROUTINE ALBEDO(Y,X,Z)
DIMENSION X(3)
COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRINT,M1MAX,KMAX,DELTA,XTAU,
1 ZERLAM,XLAM(2),B(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
2 P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
3 H2(7,7,3),CONST(3),NEQ
ARG=0.0*(Y-X(3))
Z=X(1) + X(2)*TANH(ARG)
RETURN
END

The complete program is listed:

- MAIN program
- DAUX subroutine
- NONLIN subroutine
- PANDH subroutine
- LINEAR subroutine
- OUTPUT subroutine
- ALBEDO subroutine

The following library routines are required:

- MATINV
- INTS/INTM
$JOB 2609*STRAT3,HK0160*5*0,20,P
$PAUSE
$IBJOB STRAT2 MAP
$IBFTC RTINV
COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
1 XLAM(3), B2(/7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
2 P(28,101),H(28,3,101),PLAM(3),HLAM(3,3),P2(7,7),
3 H2(7,7,3),CONST(3),NEQ
C
C
1 READ1000,N
PRINT899
PRINT900,N
READ1001,(RT(I),I=1,N)
PRINT901,(RT(I),I=1,N)
READ1001,(WT(I),I=1,N)
PRINT901,(WT(I),I=1,N)
DO 2 I=1,N
WR(I)=WT(I)/RT(I)
DO 2 J=1,N
2 AR(I,J)= 1.0/RT(I) + 1.0/RT(J)
C
899 FORMAT(1H146X36HRADIATIVE TRANSFER - INVERSE PROBLEM / )
1000 FORMAT(6I12)
900 FORMAT(6I12)
100 FORMAT(6I12)
90 FORMAT(6E12.8)
90D FORMAT(6E20.8)
D1000,NPRNT,M1MAX,KMAX
PRINT900,NPRNT,M1MAX,KMAX
READ1001,DELTA
PRINT1,DELTA
READ1U,X,XTAU,XLAM(I),I=1,3
PRINT902
PRINT903,XTAU,XLAM(I),I=1,3
902 FORMAT(1H123PHASE I - TRUE SOLUTION / )
903 FORMAT(1H0/
1 1X11HTHICKNESS =, E10.4 /
2 1X11HALBEDO(X) =, 20HA + B*TANH(10*(X-C)) //
3 1X3HA =, E16.8, 10X3HB =, E16.8, 10X3HC =, E16.8 //)
CALL NONLIN
DO 3 I=1,N
DO 3 J=1,N
3 B2(I,J)=R2(I,J)
C
C
C
C
4 READ1U,X,XTAU,XLAM(I),I=1,3
K=1
PRINT904,K
PRINT903,XTAU,XLAM(I),I=1,3
C
CALL NONLIN
C
904 FORMAT(1H1, 13HAPPROXIMATION, 13/ )
C
C QUASILINEARIZATION ITERATIONS
C
DO 5 K1=1,KMAX
PRINT904,K1
CALL PANDH
CALL LINEAR
5 CONTINUE
C
C READ1000,IGO
GO TO (1,4)*IGO
END
$IBFCT DAUX LIST
SUBROUTINE DAUX
DIMENSION V2(7,7),X(3),F(7),G(7)
1 *VLAM(3)
COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRINT,M1MAX,KMAX,DELTA,XTAU,
1 XLAM(3),B2(7,7),R2(7,7),IFLAG,R28,101,T(1491),SIG,
2 P(28,101),H(28,3,101),PLAM(3),HLAM(3,3),P2(7,7),
3 H2(7,7,3),CONST(3),NEQ
GO TO (1,2)*IFLAG
C
C LINEAR
L=3
DO 4 I=1,N
DO 4 J=1,I
L=L+1
4 V2(I,J)=T(L)
DO 5 I=1,N
DO 5 J=I,N
5 V2(I,J)=V2(J,I)
DO 51 I=1,3
L=L+1
51 VLAM(I)=T(L)
SIG=T(2)
Y=XTAU*SIG
DO 52 I=1,3
52 X(I)=VLAM(I)
CALL ALBEDO(Y,X,Z)
ZLAMDA=Z
C
DO 6 I=1,N
F(I)=U*0
DO 7 K=1,N
6 F(I)=F(I) + WR(K)*V2(I,K)
7 F(I)=U*5*F(I) + 1*0
C
DO 8 I=1,N
DO 8 J=1,I
L=L+1
DR=-AR(I,J)*V2(I,J) + ZLAMDA*F(I)*F(J)
T(L)=DR
DO 9 I=1,3
L=L+1
9 T(L)=v.0
RETURN

CLINEAR

C
2 SIG=T(2)
Y=XTAU*SIG
DO 21 I=1,3
21 X(I)=X[LAM(I)]
CALL ALBEDO(Y,X,Z)
Z[LAMDA]=Z
C
DO 16 I=1,N
F(I)=0.0
DO 17 K=1,N
17 F(I)=F(I) + WR(K)*R2(I,K)
16 F(I)=0.5*F(I) + 1.0

CP'S

C
L=3
DO 14 I=1,N
DO 14 J=1,I
L=L+1
14 V2(I,J)=T(L)
DO 15 I=1,N
DO 15 J=1,N
15 V2(I,J)=V2(J,I)
DO 18 I=1,3
L=L+1
18 VLAM(I)=T(L)
C
DO 10 I=1,N
G(I)=0.0
DO 10 K=1,N
10 G(I)=G(I) + (V2(I,K)-R2(I,K))*WR(K)
ARG=10.0*(Y-X[LAM(3)])
TARG=TANH(ARG)
XTANX=-10.0*Z[LAM(2)]*(1.0-TARG**2)
M=3+NEQ
DO 12 I=1,N
DO 12 J=1,I
FIJ=F(I)*F(J)
CAPF=-AR(I,J)*R2(I,J) + Z[LAMDA]*FIJ
T1=CAPF
T2=-AR(I,J)*(V2(I,J)-R2(I,J))
1 + 0.5*Z[LAMDA]*(F(I)*G(J) + F(J)*G(I))
T3=(VLAM(I)-X[LAM(1)])*FIJ
T4=(VLAM(2)-X[LAM(2)])*TARG*FIJ
T5=(VLAM(3)-X[LAM(3)])*XTANX*FIJ
M=M+1
DO 20 I=1,N
T(I)=G(I) + V2(I,J)*WR(J)

DO 22 I=1,N
DO 22 J=1,I
FIJ=F(I)*F(J)
T1=0.0
T2=-AR(I,J)*V2(I,J) + 0.5*ZLAMDA*(F(I)*G(J) + F(J)*G(I))
T3=VLAM(1)*FIJ
T4=VLAM(2)*TARG*FIJ
T5=VLAM(3)*XTANX*FIJ
M=M+1

T(M)=T1+T2+T3+T4+T5

DO 29 I=1,N
M=M+1

T(M)=0.0

RETURN
END

$IBFTC NONLIN
SUBROUTINE NONLIN
COMMON N,RT(7),WR(7),AR(7,7),NPRINT,MAX,MAX,DELTA,XTAU,
XLAM(3),
SIG,PLAM(3),HLAM(3,3),P2(7,7),NEQ

IFLAG=1
T(2)=0.0
T(3)=DELTA
M=1
L1=0
L3=3
DO 1 I=1,N
DO 1 J=1,I
L1=L1+1
L3=L3+1
R2(I,J)=0.0
R(L1,M)=R2(I,J)
1 T(L3)=R2(I,J)
DO 2 I=1,J
L3=L3+1
2 T(L3)=XLAM(I)
C
NEQ=(N*(N+1))/2 + 3
CALL INTS(T,NEQ,2,0,0,0,0,0,0)
C
SIG=T(2)
CALL OUTPUT
C
DO 5 M1=1,M1MAX
DO 4 M2=1,NPRNT
CALL INTM
M=M+1
L1=0
L3=3
DO 3 I=1,N
DO 3 J=1,I
L1=L1+1
L3=L3+1
R2(I,J)=T(L3)
3 R(L1,M)=R2(I,J)
4 SIG=T(2)
5 CALL OUTPUT
C
RETURN
END

$IBTFC PANDH
SUBROUTINE PANDH
COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,MMAX,DELTAX,XTAU,
XLAM(3)*,B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491)*,SIG,
P(28,101),H(28,3,101),PLAM(3),HLAM(3,3),P2(7,7)*,
H2(7,7,3),CONST(3)*,NEQ
IFLAG=2
T(2)=0.0
T(3)=DELTA
M=1
C P'S
C
L1=0
L3=3
DO 1 I=1,N
DO 1 J=1,I
L1=L1+1
L3=L3+1
P(L1,M)=0.0
1 T(L3)=P(L1,M)
DO 2 I=1,3
L3=L3+1
PLAM(I)=0.0
2 T(L3)=PLAM(I)

C H'S

DO 7 K=1,3
L1=0
DO 3 I=1,N
DO 3 J=1,I
L1=L1+1
L3=L3+1
H(L1*K*M)=0.0
3 T(L3)=H(L1,K,M)

C

DO 7 I=1,3
L3=L3+1
HLAM(I,K)=0.0
IF(I-K)7,6,7
6 HLAM(I,K)=1.0
7 T(L3)=HLAM(I,K)

C

L=0
DO 8 I=1,N
DO 8 J=1,I
L=L+1
8 R2(I,J)=k(L,M)
DO 9 I=1,N
DO 9 J=1,N
9 R2(I,J)=R2(J,I)

C

NEQ=4*{(N*(N+1))/2 + 3)
CALL INTS(T,NEQ,2,0,0,0,0,0,0)
MAX=({N*(N+1))/2
PRINT52,T(2),(P(L,M),H(L,1,M),L=1,LMAX)
52 FORMAT(1H0F9.4,5E20.8/(10X5E2G.8))

C

DO 51 M1=1,M1MAX
DO 50 M2=1,NPRNT
CALL INTM
M=M+1
CPREV APPROX R(I,J)

L1=0
DO 10 I=1,N
DO 10 J=1,I
L1=L1+1
10 R2(I,J)=R(L1,M)
DO 11 I=1,N
DO 11 J=1,N
11 R2(I,J)=R2(J,I)
L1=0
L3=3
DO 12 I=1,N
DO 12 J=1,I
$\text{10}\text{ FTC LINEAR}
\text{SUBROUTINE LINEAR}
\text{DIMENSION CHK1(3)}
\text{DIMENSION A(49,3),B(49),EMAT(50,50)}, \text{PIVOT(50),INDEX(50,2)}
\text{1,PIVOT(50),FVEC(50,1)}
\text{COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,}\text{1}
\text{XLAM(3)}, \text{2,B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,}\text{2}
\text{P(28,101),H(28,3,101),PLAM(3),HLAM(3,3),P2(7,7),}\text{3}
\text{H2(7,7,3),CONST(3),NEQ}
\text{CBOUNDARY CONDITIONS}
\text{MLAST=NPRNT*M1MAX + 1}
\text{DO 1 K=1,3}
\text{L=0}
\text{DO 2 I=1,N}
\text{DO 2 J=1,I}
\text{L=L+1}
\text{2 H2(I,J,K)=H(L,K,MLAST)}
\text{DC 1 I=1,N}
\text{DO 1 J=I,N}
\text{1 H2(I,J,K)=H2(J,I,K)}
\text{L=0}
\text{DO 3 I=1,N}
\text{DO 3 J=1,I}
\text{L=L+1}
\text{3 P2(I,J)=P(L,MLAST)}
\text{DO 4 I=1,N}
\text{DO 4 J=I,N}
\text{4 P2(I,J)=P2(J,I)}
\text{CLEAST SQUARES}
\text{DO 5 K=1,3}
\text{L=0}
\text{DO 5 I=1,N}
\text{DO 5 J=I,N}
\text{L=L+1}
\text{5 A(I,K)=H2(I,J,K)}
\text{L=0}
\text{DO 6 I=1,N}
\text{DO 6 J=I,N}
\text{L=L+1}
6  B(L)=B2(I,J) - P2(I,J)

C  LMAX=N*2
C
60  FORMAT(1HO)
61  DO 61 L=1,LMAX
61  PRINT82*(A(L*K)*K=1,3)*B(L)

C
DO 8 I=1,3
DO 7 J=1,3
SUM=0.0
DO 9 L=1,LMAX
SUM=SUM + A(L,J)*A(L,J)
7  EMAT(I,J)=SUM
SUM=0.0
DO 10 L=1,LMAX
10  SUM=SUM + A(L,I)*B(L)
8  FVEC(I,1)=SUM

C
PRINT60
DO 81 I=1,3
81  PRINT82*(EMAT(I,J),J=1,3),FVEC(I,1)
92  FORMAT(10X6E20,8)

C
CALL MATINV(EMAT,3,FVEC,1,DETERM,PIVOT,INDEX,IPIVOT)

C
DO 11 I=1,3
11  CONST(I)=FVEC(I,1)

C
DO 20 I=1,3
20  XLAM(I)=CONST(I)
PRINT903,XTAU,(XLAM(I),I=1,3)
903  FORMAT(1HO/
1  1X11HTHICKNESS =, E16.8 /
2  1X11HALBEDO(X) =, 20HA + B*TANH(10*(X-C)) //
3  1X3HA =, E16.8, 10X3HB =, E16.8, 10X3HC =, E16.8 //)

C
CNEW APPROXIMATION

C
M=1
L=0
DO 12 I=1,N
DO 12 J=1,I
L=L+1
SUM=P(L,M)
DO 13 K=1,3
13  SUM =SUM + CONST(K)*H(L,K,M)
12  R(L,M)=SUM
L=0
DO 14 I=1,N
DO 14 J=1,I
L=L+1
14  R2(I,J)=R(L,M)
SIG=0.0
CALL OUTPUT
C
DO 50 M1=1,M1MAX
DO 18 M2=1,NPRNT
M=M+1
L=0
DO 15 I=1,N
DO 15 J=1,I
L=L+1
SUM=P(L,M)
DO 16 K=1,3
SUM=SUM + CONSTU)H(L,K,M)
15 R(L,M)=SUM
L=0
DO 17 I=1,N
DO 17 J=1,I
L=L+1
17 R2(I,J)=R(L,M)
18 SIG=SIG + DELTA
50 CALL OUTPUT
C
RETURN
END
$IBFTC OUTPUT
SUBROUTINE OUTPUT
DIMENSION X(3)
COMMON N,RT(7),WT(7),WR(7),R(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
1 XLAM(3), B2(7,7), R2(7,7), IFLAG,R(28,101),T(1491),SIG,
2 P(28,101), H(28,3,101), PLAM(3), HLAM(3,3), P2(7,7),
3 H2(7,7,3), CONST(3), NEQ
DO 1 I=1,N
DO 1 J=1,N
1 R2(I,J)=R2(J,I)
Y=XTAU*SIG
DO 3 I=1,3
3 X(I)=XLAM(I)
CALL ALBEDO(Y,X,Z)
PRINT100, SIG,Y,Z
100 FORMAT(10, 7HSIGMA =, F6.2, 4X5HTAU =, F6.2, 4X8HALBEDO =, F6.2)
DO 2 J=1,N
2 PRINT101,J, (R2(I,J)+I=1,N)
101 FORMAT(!10, 7F10,6)
RETURN
END
$IBFTC ALBEDO
SUBROUTINE ALBEDO(Y,X,Z)
DIMENSION X(3)
COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
1 XLAM(3), B2(7,7), R2(7,7), IFLAG,R(28,101),T(1491),SIG,
2 P(28,101), H(28,3,101), PLAM(3), HLAM(3,3), P2(7,7),
3 H2(7,7,3), CONST(3), NEQ
ARG=10,0*(Y-X(3))
Z=X(1) + X(2)*TANH(ARG)
RETURN
END
$ENTRY RTINV
<table>
<thead>
<tr>
<th>7</th>
<th>25446046E-01</th>
<th>112923441E-00</th>
<th>29707742E-00</th>
<th>500000000E</th>
<th>070292258E-00</th>
<th>0087076559E</th>
<th>00</th>
</tr>
</thead>
<tbody>
<tr>
<td>97455396E</td>
<td>00</td>
<td>64742484E</td>
<td>01</td>
<td>13985269E</td>
<td>0019091502E</td>
<td>0020897958E</td>
<td>00</td>
</tr>
<tr>
<td>64742484E-01</td>
<td>10</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td>0.3</td>
<td>0.1</td>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>0.6</td>
<td>0.9</td>
<td>0.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$\text{SYS}$
APPENDIX C

PROGRAMS FOR RADIATIVE TRANSFER:

NOISY OBSERVATIONS
PROGRAM C.1. **MANY ACCURATE OBSERVATIONS FOR THE DETERMINATION OF ALBEDO**

The complete program is listed:

- MAIN program
- DAUX subroutine
- ALBEDO subroutine
- PANDH subroutine
- LINEAR subroutine
- NONLIN subroutine
- OUTPUT subroutine

The following library routines are required:

- MATINV
- INTS/INTM
SUBROUTINE RTINV
    COMMON N,RT(7),WT(7),R(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
    ZERLAM,XLAM(2),B2(7,7),R2(7,7),FLAG,R(28,101),T(1491),SIG,
    P(128,101),H(128,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
    H2(7,7,3),CONST(3),NEQ

    PHASE I

1  READ1000,N
   PRINT899
   PRINT900,N
   READ1001,(RT(I),I=1,N)
   PRINT901,(RT(I),I=1,N)
   READ1002,(WT(I),I=1,N)
   PRINT903,(WT(I),I=1,N)
   DO 1 I=1,N
       WR(I)=WT(I)/RT(I)
   DO 2 J=1,N
2  AR(I,J)=1.0/RT(I)+1.0/RT(J)

899 FORMAT(1H146X36HRADIATIVE TRANSFER - INVERSE PROBLEM /)
1  47X33HKNOWN QUADRATIC ALBEDO FUNCTION /
2  47X27HUNKOWN THICKNESS OF MEDIUM //)
1000 FORMAT(6I12)
900 FORMAT(6I20)
1001 FORMAT(6E12.8)
901 FORMAT(6E20.8)
   READ1000,PRNT,M1MAX,KMAX
   PRINT900,PRNT,M1MAX,KMAX
   READ1000,DELTA
   PRINT901,DELTA
   READ1002,XTAU,ZERLAM,XLAM(1),XLAM(2)
   PRINT902
   PRINT903,XTAU,ZERLAM,XLAM(1),XLAM(2)
902 FORMAT(1H123HAPPROXIMATE SOLUTION /)
903 FORMAT(1H40/
1  1X11HTHICKNESS = , F10.4 /
2  1X11HALED0(X) = , F6.2,2H **, F6.2,3HX **, F6.2,4HX**2 //)
   CALL NONLIN
   DO 3 I=1,N
   DO 3 J=1,N
3  B2(I,J)=R2(I,J)

    PHASE II

4  READ1003,XTAU,ZERLAM,XLAM(1),XLAM(2)
   K=0
   PRINT904,K
   PRINT903,XTAU,ZERLAM,XLAM(1),XLAM(2)
   CALL NONLIN

904 FORMAT(1H13HAPPROXIMATION, I3/)
C
C QUASILINEARIZATION ITERATION...

DO 5 K1=1,KMAX
PRINT904,K1
CALL PANDH
CALL LINEAR
5 CONTINUE

C
C REAO1000,IGO
GO TO (1,4),IGO
END

SUBROUTINE DAUX
DIMENSION V2(7,7),X(3),F(7),G(7)
COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
1 ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R!28,101,T(1491),SIG,
2 P2(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
3 H2(7,7,3),CONST(3)*NEQ
GO TO (1,2),IFLAG
C
CNONLINEAR
C
1 L=3
DO 4 I=1,N
DO 4 J=1,I
L=L+1
4 V2(I,J)=T(L)
DO 5 I=1,N
DO 5 J=I,N
5 V2(I,J)=V2(J,I)
L=L+1
VTAU=T(L)
L=L+1
VLAM1=T(L)
L=L+1
VLAM2=T(L)
SIG=T(2)
Y=XTAU*SIG
X(1)=ZERLAM
X(2)=VLAM1
X(3)=VLAM2
CALL ALBEDO(Y,X,Z)
ZLAMDA=Z
C
DO 6 I=1,N
F(1)=0.0
DO 7 K=1,N
7 F(I)=F(I) + WR(K)*V2(I,K)
6 F(I)=0.5*F(I) + 1.0
C
DO 8 I=1,N
DO 8 J=1,I
L=L+1
DR=-AR(I,J)*V2(I,J) + ZLAMDA*F(I)*F(J)

T(L)=DR*VTAU
DO 9 I=1,3
   L=L+1
9   T(L)=0.0
RETURN

CLINEAR

SIG=T(2)
Y=XTAU*SIG
X(1)=ZERLAM
X(2)=XLAM(1)
X(3)=XLAM(2)
CALL ALBEDO(Y,X,Z)
ZLAMDA=Z

DO 16 I=1,N
   F(I)=0.0
DO 17 K=1,N
   F(I)=F(I) + WR(K)*R2(I,K)
17   F(I)=0.5*F(I) + 1.0

DO 14 I=1,N
   DO 14 J=1,I
      L=L+1
14      V2(I,J)=T(L)
   DO 15 I=1,N
      DO 15 J=I,N
      L=L+1
15      V2(I,J)=V2(J,I)
   DO 16 I=1,N
   DO 16 J=I,N
      L=L+1
VTAU=T(L)
L=L+1
VLAM1=T(L)
L=L+1
VLAM2=T(L)

DO 10 I=1,N
   DO 10 J=1,N
      G(I)=G(I) + (V2(I,K)-R2(I,K))*WR(K)
      M=3+NEQ
   DO 12 J=1,N
10      G(I)=G(I) + (V2(I,K)-R2(I,K))*WR(K)
      M=3+NEQ
   DO 12 J=1,N
12      G(I)=G(I) + (V2(I,K)-R2(I,K))*WR(K)
      M=3+NEQ
      F1J=F(I)*F(J)
      CAPF=-AR(I,J)*R2(I,J) + ZLAMDA*F1J
      T1=-XTAU*AR(I,J)*(V2(I,J)-R2(I,J))
      T2=0.5*XTAU*ZLAMDA*(F(I)*G(J)+F(J)*G(I))
      T3=XTAU*CAPF
      T4=(VTAU-XTAU)*(XLAM(1)*Y+2.0*XLAM(2)*Y**2)*F1J
      PROD=XTAU*Y*F1J
T5=(VLAM1-XLAM(1))*PRCD
T6=(VLAM2-XLAM(2))*PRCD*Y
M=M+1
12 T(M)=T1+T2+T3+T4+T5+T6
   DO 19 I=1,3
   M=M+1
19 T(M)=0.0
C
C  CH'S
C
   DO 100 K=1,3
C
   DO 24 I=1,N
   DO 24 J=1,1
   L=L+1
24   V2(I,J)=T(L)
   DO 25 I=1,N
   DO 25 J=1,N
25   V2(I,J)=V2(J,I)
   L=L+1
   VTAU=T(L)
   L=L+1
   VLAM1=T(L)
   L=L+1
   VLAM2=T(L)
C
   DO 20 I=1,N
20   G(I)=G(I)+V2(I,J)*WR(J)
   DO 22 I=1,N
   DO 22 J=1,I
   FIJ=F(I)*F(J)
   CAPF=-AR(I,J)*R2(I,J)+ZLAMDA*FIJ
   T1=XTAU*AR(I,J)*V2(I,J)
   T2=0.5*XTAU*ZLAMDA*(F(I)*G(J)+F(J)*G(I))
   T3=VTAU*CAPF
   T4=VTAU*(XLAM(1)*Y+2.0*XLAM(2)*Y**2)*FIJ
   PROD=XTAU*Y*FIJ
   T5=VLAM1*PROD
   T6=VLAM2*PROD*Y
   M=M+1
22   T(M)=T1+T2+T3+T4+T5+T6
C
   DO 29 I=1,3
   M=M+1
29   T(M)=0.0
100   CONTINUE
   RETURN
END
$IBFTC ALBEDO
SUBROUTINE ALBEDO(Y,X,Z)
DIMENSION X(3)
COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
$\Sigma_{BFTC}$ PANDH

SUBROUTINE PANDH

COMMON N,RT(7),WT(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
1 ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
2 P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
3 H2(7,7,3),CONST(3),NEQ

Z=X(1) + X(2)*Y + X(3)*Y**2

RETURN

END

C P'S
C
L1=0
L3=3
DO 1 I=1,N
DO 1 J=1,I
L1=L1+1
L3=L3+1
P(L1,M)=0.0
1 T(L3)=P(L1,M)
L3=L3+1
PTAU=0.0
T(L3)=PTAU
DO 2 I=1,2
L3=L3+1
PLAM(I)=0.0
2 T(L3)=PLAM(I)
C
C H'S
C
DO 7 K=1,3
L1=0
DO 3 I=1,N
DO 3 J=1,I
L1=L1+1
L3=L3+1
H(L1,K,M)=0.0
3 T(L3)=H(L1,K,M)
1 3+1
H(L1,K)=0.0
IF(K-1)=5,4,5
4 HTAU(K)=1.0
5 T(L3)=HTAU(K)
DO 7 I=1,2
L3=L3+1
HLAM(I,K)=0.0
IF(K-1)=7,6,7
6 HLAM(I,K)=1.0
7 T(L3)=HLAM(I+K)

C

I=0
DO 8 I=1,N
DO 8 J=1,I
L=L+1
8 R2(I,J)=R(L+M)
DO 9 I=1,N
DO 9 J=I,N
9 R2(I,J)=R2(J,I)

: NEQ=4*((N*(N+1))/2 + 3)
CALL INTS(T,NEQ,2,0,0,0,0,0)

C

DO 50 M1=1,M1MAX
DO 50 M2=1,NPRNT
CALL INTM
M=M+1
CPREV.APPROX. R(I,J)
L1=C
DO 10 I=1,N
DO 10 J=1,I
L1=L1+1
10 R2(I,J)=R(L1+M)
DO 11 I=1,N
DO 11 J=I,N
11 R2(I,J)=R2(J,I)
L1=C
L3=3
DO 12 I=1,N
DO 12 J=1,I
L1=L1+1
L3=L3+1
12 P(L1+M), L3
L3=L3+3
DO 13 K=1,3
L1=C
DO 14 I=1,N
DO 14 J=1,I
L1=L1+1
L3=L3+1
14 H(L1+K,M)=T(L3)
13 L3=L3+3
50 CONTINUE
RETURN
END
SIBFTC LINEAR
SUBROUTINE LINEAR
DIMENSION CHKI(3)
DIMENSION A(49,3),B(49),EMAT(50,50),
PIVOT(50),INDEX(50,2)
1,PIVOT(50),FVEC(50,1)
COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
1 ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
2 P(28,101),H(28,3,101),PTAU,P0(2),PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
3 H2(7,7,3),CONST(3),NEQ
BOUNDARY CONDITIONS

1. H2(I, J, K) = H2(J, I, K)
3. P2(I, J) = P(L, MLAST)
4. P2(I, J) = P2(J, I)

LEAST SQUARES

1. C
2. LMAX = N**2

C

1. PRINT60
2. DO 61 L = 1, LMAX
3. PRINT62, (A(L, K), K = 1, 3), B(L)

C

1. SUM = 0
2. DO 9 SUM = SUM + A(L, I) * A(L, J)
3. EMAT(I, J) = SUM
4. SUM = 0
5. DO 10 SUM = SUM + A(L, I) * B(L)
6. FVEC(I, 1) = SUM

C

1. PRINT60
2. DO 81 I = 1, 3
3. PRINT82, (EMAT(I, J), J = 1, 3), FVEC(I, 1)
4. FORMAT(10X6E20.8)

C

SAVE FOR CHECKING
DO 83 I=1,3
DO 84 J=1,3
83 A(I,J)=EMAT(I,J)
84 B(I)=FVEC(I)

CALL MATINV(EMAT,3,FVEC,1,DETERK,PIVOT,INDEX,IPIVOT)

DO 11 I=1,3
11 CONST(I)=FVEC(I)

C**-CK MATRIX INVERSE

PRINT60
DO 71 I=1,3
DO 70 J=1,3
CHKI(J)=0.0
DO 70 L=1,3
70 CHKI(J)=CHKI(J) + EMAT(I,L)*A(L,J)
71 PRINT82,(CHKI(J),J=1,3)

DO 72 J=1,3
CHKI(J)=0.0
DO 72 L=1,3
72 CHKI(J)=CHKI(J) + EMAT(J,L)*B(L)
PRINT82,(CHKI(J),J=1,3)

XTAU=CONST(1)
XLAM(1)=CONST(2)
XLAM(2)=CONST(3)
PRINT903,XTAU,ZERLAM,XLAM(1),XLAM(2)
903 FORMAT(1H0/
1 1X11THICKNESS =, F10.4 /
2 1X12HALBEDO(x) =, F6.2, 17n + C1*x + C2*x**2, 
3 2X3HC1=*, E18.6, 2X3HC2=*, E18.6//)

CNEW APPROXIMATION

M=1
L=0
DO 12 I=1,N
DO 12 J=1,I
L=L+1
SUM=P(L,M)
DO 13 K=1,3
13 SUM =SUM + CONST(K)*H(L,K,M)
12 R(L,M)=SUM
L=0
DO 14 I=1,N
DO 14 J=1,I
L=L+1
14 R2(I,J)=R(L,M)
SIG=0.0
CALL OUTPUT
C
DO 50 M1=1,M1MAX
DO 18 M2=1,NPRNT
M=M+1
L=0
DO 15 I=1,N
DO 15 J=1,I
L=L+1
SUM=P(L,M)
DO 16 K=1,3
16 SUM=SUM + CONST(K)*H(L,K,M)
15 R(L,M)=SUM
L=0
DO 17 I=1,N
DO 17 J=1,I
L=L+1
17 R2(I,J)=R(L,M)
18 SIG=SIG + DELTA
50 CALL OUTPUT
C
RETURN
END
$IBFTC NONLIN
SUBROUTINE NONLIN
COMMON N,RT(7),W(7),R(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
1 ZERLAM,XLAM(2),DI2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
2 P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
3 H2(7,7,3),CONST(3),NEQ
C
IFLAG=1
T(2)=0.0
T(3)=DELTA
M=1
L1=0
L3=3
DO 1 I=1,N
DO 1 J=1,I
L1=L1+1
L3=L3+1
R2(I,J)=0.0
R(L1,M)=R2(I,J)
1 T(L3)=R2(I,J)
L3=L3+1
T(L3)=XTAU
DO 2 I=1,2
L3=L3+1
2 T(L3)=XLAM(I)
C
NEQ=(N*(N+1))/2 + 3
CALL INTS(T,NEQ,2,0,0,0,0,0,0)
C
SIG=T(2)
CALL OUTPUT
C
DO 5 M1=1,M1MAX
DO 4 M2=1,NPRNT
CALL INTM
M=M+1
L1=0
L3=3
DO 3 I=1,N
DO 3 J=1,I
L1=L1+1
L3=L3+1
R2(I,J)=T(L3)
3 R(L1,M)=R2(I,J)
4 SIG=T(2)
5 CALL OUTPUT
C
RETURN
END
SIBFTC OUTPUT
SUBROUTINE OUTPUT
DIMENSION X(3)
COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
1 ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
2 P(2E+101),H(28,3,101),PTAU,XLAM(2),HTAU(3),XLAM(2,3),P2(7,7),
3 H2(7,7,3),CONST(1),NEQ
DO 1 I=1,N
DO 1 J=1,N
1 R2(I,J)=R2(J,I)
Y=XTAU*SIG
X(1)=ZERLAM
X(2)=XLAM(1)
X(3)=XLAM(2)
CALL ALBEDO(Y,X,Z)
PRINT100, SIG,Y,Z
100 FORMAT(1H0 7HSIGMA =,F6.2, 4X5HTAU =, F6.2, 4X8HALBEDO =,F6.2/)
DO 2 J=1,N
2 PRINT101,J,(R2(I,J)-1*1)
101 FORMAT(1H0, 7F10.6)
RETURN
END
$ENTRY RT SUB
7
25446046E-01 101292344E-00 29707742E-00 00000000E-00 0292258E-00 097076559E 00
9745396E-00
64742484E-01 019091502E-00 20967959E-00 001091502E-00 013985269E-00
64742484E-01
1 0,01 0,5 2,0 -2,0
1,0 0,5 2,0 -2,0
544
PROGRAM C.2. OBSERVATIONS FOR ONLY ONE ANGLE OF INCIDENCE, FOR THE DETERMINATION OF ALBEDO

A partial program is listed:

MAIN program
LINEAR subroutine

The following subroutines are required from Program C.1:

DAUX subroutine
ALBEDO subroutine
PANDH subroutine
NONLIN subroutine
OUTPUT subroutine

The following routines are required:

MATINV
INTS/INTM
COMMON NRT(7),RT(7),R(7),T(1491),SIG,PI(28,101),P2(7,7),
   H(28,3,101),PLAM(2),HTAU(3),XLAM(2,3),P2(7,7),
   CONST(3),NEQ,NINC,JINC(7),NOBS

C PHASE I

1 READ1000,N
   PRINT999
   PRINT900,N
   READ1001,RT(I),I=1,N
   PRINT901,RT(I),I=1,N
   READ1001,WT(I),I=1,N
   PRINT901,WT(I),I=1,N
   DO 2 I=1,N
   WR(I)=WT(I)/RT(I)
   DO 2 J=1,N
   AR(I,J)=1.0/RT(I) + 1.0/RT(J)

C 699 FORMAT(1H146X36HRADIATIVE TRANSFER - INVERSE PROBLEM /
    1 47X33UNKNOWN QUADRATIC ALBEDO FUNCTION /
    2 47X27UNKNOWN THICKNESS OF MEDIUM //)

1000 FORMAT(6I12)
900 FORMAT(6I12)
1001 FORMAT(6E12.8)
901 FORMAT(6E20.8)
   READ1000,NPRNT,M1MAX,KMAX
   PRINT900,NPRNT,M1MAX,KMAX
   READ1001,DELTA
   PRINT901,DELTA
   READ1001,XTAU,ZERLAM,XLAM(1),XLAM(2)
   PRINT902
   PRINT903,XTAU,ZERLAM,XLAM(1),XLAM(2)
902 FORMAT(1H123H PHASE I - TRUE SOLUTION /)
903 FORMAT(1HC/)
   1 1X11THICKNESS = F10.4 /
   2 1X11HALBEDO(X) = F6.2*2M + F6.2*3HX + F6.2*4HX*2)
   CALL NONLIN
   DO 3 I=1,N
   DO 3 J=1,N
   3 B2(I,J)=R2(I,J)

C PHASE II

4 READ1001,XTAU,ZERLAM,XLAM(1),XLAM(2)
   K=0
   PRINT904,K
   PRINT903,XTAU,ZERLAM,XLAM(1),XLAM(2)

C READ1000,NINC
   PRINT900,NINC
   READ1000,(JINC(I),I=1,NINC)
PRINT900,(JINC(1),I=1,NINC)

NOBS=NINC*N
PRINT900,NOBS

CALL NONLIN

904 FORMAT(1H1,13HAPPROXIMATION,13/)

QUASILINEARIZATION ITERATIONS

DO 5 K1=1,KMAX
PRINT904,K1
CALL PANDH
CALL LINEAR
5 CONTINUE

READ1000,IGO
GO TO (1,4),IGO
END

$IBFC LINEAR LIST
SUBROUTINE LINEAR
DIMENSION CHKIK(3)
DIMENSION A(49,3),B(49),EMAT(5,50),PIVOT(50),INDEX(50,2)
1,IPIVOT(50),FVEC(50,1)
COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
1 ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
2 P(26,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
3 H2(7,7,3),CONST(3),NEQ
4 ,NINC,JINC(7),NOBS

CBOUNDARY CONDITIONS
MLAST=NPRNT*M1MAX+1
DO 1 K=1,3
L=G
DO 2 I=1,N
DO 2 J=1,I
L=L+1
2 H2(I,J,K)=H(L,K,MLAST)
DO 1 I=1,N
DO 1 J=1,I
1 H2(I,J,K)=H2(J,I,K)
L=U
DO 3 I=1,N
DO 3 J=1,I
L=L+1
3 P2(I,J)=P(L,MLAST)
DO 4 I=1,N
DO 4 J=1,N
4 P2(I,J)=P2(J,1)
CLEAST SQUARES
DO 5 K=1,3
L=0
DO 5 IN=1,NINC
   I=JINC(IN)
   DO 5 J=1,N
   L=L+1
5   A(L,K)=H2(I,J,K)
   L=0
   DO 6 IN=1,NINC
      I=JINC(IN)
      DO 6 J=1,N
      L=L+1
6   B(L)=B2(I,J) - P2(I,J)
C
   LMAX=N**2
PRINT60
60 FORMAT(1HO)
   DO 61 L=1,NOBS
61 PRINT82,(A(L,K),K=1,3),B(L)
C
   DO 8 I=1,3
      DO 7 J=1,3
      SUM=0.0
      DO 9 L=1,NOBS
9      SUM=SUM + A(L,I)*A(L,J)
7      EMAT(I,J)=SUM
      SUM=0.0
   DO 10 L=1,NOBS
10     SUM=SUM + A(L,I)*B(L)
   FVEC(I,1)=SUM
C
PRINT60
   DO 81 I=1,3
81     PRINT82,(EMAT(I,J),J=1,3),FVEC(I,1)
82    FORMAT(10X6E20.8)
C
C    SAVE FOR CHECKING
C
   DO 83 I=1,3
      DO 84 J=1,3
84     A(I,J)=EMAT(I,J)
83     B(I)=FVEC(I,1)
C
C    CALL MATINV(EMAT,3,FVEC,1,DETERM,PIVOT,INDEX,IPIVOT)
C
   DO 11 I=1,3
11    CONST(I)=FVEC(I,1)
C
C    CHECK MATRIX INVERSE
C
PRINT60
   DO 71 I=1,3
      DO 70 J=1,3
70     CHK1(J)=0.0
71    DO 70 L=1,3
70     CHK1(J)=CHK1(J) + EMAT(I,L)*A(L,J)
71 PRINT62,(CHKI(J),J=1,3)
C
DO 72 J=1,3
CHKI(J)=0.0
DO 72 L=1,3
72 CHKI(J)=CHKI(J) + EMAT(J,L)*b(L)
PRINT62,(CHKI(J),J=1,3)
C
XTAU=CONST(1)
XLAM(1)=CONST(2)
XLAM(2)=CONST(3)
PRINT903,XTAU,ZERLATM,XLAM(1),XLAM(2)
903 FORMAT(1H0,
1 1X11HTHICKNESS =, E18.6 /
2 1X12HALBED0(X, = , F6.2*, 17H + C1*X + C2*X**2),
3 2X3HC1= , E18.6, 2X3HC2= , E18.6//)
C
CNEW APPROXIMATION
C
M=1
L=0
DO 12 I=1,N
DO 12 J=1,I
L=L+1
SUM=P(L,M)
DO 13 K=1,3
13 SUM =SUM + CONST(K)*H(L,K,M)
12 R(L,M)=SUM
L=0
DO 14 I=1,N
DO 14 J=1,I
L=L+1
14 R2(I,J)=R(L,M)
SIG=0.0
CALL OUTPUT
C
DO 50 M1=1,M1MAX
DO 18 M2=1,NPRNT
V=M+1
L=0
DO 15 I=1,N
DO 15 J=1,I
L=L+1
SUM=P(L,M)
DO 16 K=1,3
16 SUM=SUM + CONST(K)*H(L,K,M)
15 R(L,M)=SUM
L=0
DO 17 I=1,N
DO 17 J=1,I
L=L+1
17 R2(I,J)=R(L,M)
18 SIG=SIG + DELTA
50 CALL OUTPUT
C
PROGRAM C.3. ERRORS IN THE OBSERVATIONS FOR THE DETERMINATION OF ALBEDO

A partial program is listed:

MAIN program

The following subroutines are required from Program C.1:

DAUX subroutine
ALBEDO subroutine
PANDH subroutine
NONLIN subroutine
OUTPUT subroutine

The following subroutine is required from Program C.2:

LINEAR subroutine

The following library routines are required:

MATINV
INTS/INTM
$JOBJOBT RTINV2 MAP
$1BJTC RTINV LIST

DIMENSION DERR(7,7), C2(7,7)
COMMON N,RT(7,T(7), WR(7), AR(7), PRINT, M1MAX, KMAX,*DELTA, XTAU,
1 ZERLAM, XLAM(2), P2(7,7), R2(7,7), IFLAG, R(28,101), T(1491), SIG,
2 P(28,101), H(28,3,101), PTAU, PLAM(2), HTAU(3), HLAM(2,3), P2(7,7),
3 H2(7,7,3), CONST(3), NEQ
4 NINC, JINC(7), NOBS

PHASE I

1 READ1000, N
PRINT899
PRINT900, N
READ1001, (RT(I), I=1,N)
PRINT901, (RT(I), I=1,N)
READ1001, (WT(I), I=1,N)
PRINT901, (WT(I), I=1,N)
DO 2 I=1,N
WR(I) = WT(I) / RT(I)
DO 2 J=1,N
AR(I,J) = 1.0 / RT(I) + 1.0 / RT(J)

PHASE II

DO 3 I=1,N
DO 3 J=1,N
B2(I,J) = R2(I,J)

CALL NONLIN
DO 3 I=1,N
DO 3 J=1,N

B2(I,J) = R2(I,J)

FORMAT(1H146X36HRADIATIVE TRANSFER - INVERSE PROBLEM / / 1
47X33HUNKNOWN QUADRATIC ALBEDO FUNCTION / / 2
47X27HUNKNOWN THICKNESS OF MEDIUM //)

FORMAT(6E12.8)
FORMAT(6E20.8)
READ1003, NPRNT, M1MAX, KMAX
PRINT900, NPRNT, M1MAX, KMAX
READ1003, DELTA
PRINT901, DELTA
READ1001, XTAU, ZERLAM, XLAM(1), XLAM(2)
PRINT902
PRINT903, XTAU, ZERLAM, XLAM(1), XLAM(2)

FORMAT(1H123PHASE I - TRUE SOLUTION //)
FORMAT(1HG7/
1 1X11THICKNESS = F10.4 //
2 1X11HALBEDO(X) = F6.2,2X ** F6.2,3HX +, F6.2,4HX**2 //)
CALL NONLIN
DO 3 I=1,N
DO 3 J=1,N
3 B2(I,J) = R2(I,J)

FORMAT(1HG7/
1 1X11THICKNESS = F10.4 //
2 1X11HALBEDO(X) = F6.2,2X ** F6.2,3HX +, F6.2,4HX**2 //)
CALL NONLIN
DO 3 I=1,N
DO 3 J=1,N
3 B2(I,J) = R2(I,J)
READ1000, NINC

PRINT900, NINC
READ1000, (JINC(I), I=1, NINC)
PRINT900, (JINC(I), I=1, NINC)
NOBS=NINC*N
PRINT900, NOBS

READ ERRORS AS DECIMALS

DO 6 I=1, NINC
READ1001, (DERR(I, J), J=1, N)
6 PRINT901, (DERR(I, J), J=1, N)

STORE CORRECT OBSERVATIONS

DO 7 I=1, N
DO 7 J=1, N
7 C2(I, J)=B2(I, J)

CORRUPT OBSERVATIONS

PRINT100
DO 81 IN=1, NINC
I=JINC(IN)
DO 8 J=1, N
8 B2(I, J)=B2(I, J)*(1.0+DERR(IN, J))
81 PRINT101, I, (B2(I, J), J=1, N)
100 FORMAT(1H0)
101 FORMAT(I10, 7F10.6)

CALL NONLIN

9 4 FORMAT(1H1, 13HAPPROXIMATION, 13F)

QUASILINEARIZATION ITERATIONS

DO 5 K1=1, KMAX
PRINT904, K1
CALL PANDH
CALL LINEAR
5 CONTINUE

RESTORE CORRECT OBSERVATIONS

DO 9 I=1, N
DO 9 J=1, N
9 C2(I, J)=C2(I, J)

GO TO 4
END
<table>
<thead>
<tr>
<th>SENTRY</th>
<th>RTINV</th>
</tr>
</thead>
<tbody>
<tr>
<td>25446046E-0112923441E-0029707742E-006992E-0007292258E 0087076559E 00</td>
<td>032</td>
</tr>
<tr>
<td>97455396E 00</td>
<td></td>
</tr>
<tr>
<td>64742484E-0113985265E-0019091502E-0029E+7548E-0019091502E-0013985269E-00</td>
<td>00 2</td>
</tr>
<tr>
<td>64742484E-01</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>0.1</td>
<td>0.5</td>
</tr>
<tr>
<td>1.0</td>
<td>0.5</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>0.007740</td>
<td></td>
</tr>
</tbody>
</table>

$\text{SIBSYS}$
PROGRAM C.4. MINIMAX CRITERION FOR THE DETERMINATION OF ALBEDO

A partial program is listed:

LINEAR subroutine

The MAIN program is required from Program C.2.

The following subroutines are required from Program C.1:

DAUX subroutine
ALBEDO subroutine
PANDH subroutine
NONLIN subroutine
OUTPUT subroutine

The following library routines are required:

MATINV
INTS/JNTM
SUBROUTINE LINEAR

DIMENSION INFIX(8), AS(22, 35), BS(22), TOL(4), KOUT(7), ERR(8), JH(22),
XS(22), PS(22), YS(22), K3(35), ES(22, 22), ZS(35), A(7, 3), B(7),
COMMON N, RT(7), WT(7), WR(7), AR(7, 7), RPRNT, M1MAX, KM1AX, DELTA, XTAU,
1 ZERLAM, XILAM(2), b2(7, 7), R2(7, 7), IFLAG, R(28, 101), T(1491), SIG,
2 P(28, 101), H(28, 3, 101), PTAU, PLAM(2), HTAU(3), HLAM(2, 3), P2(7, 7),
3 H2(7, 7, 3), CONST(3), NEQ
JH, NINC, JINC(7), NOBS

USE LINEAR PROGRAMMING TO MINIMIZE MAXIMUM DEVIATION

CBOUNDARY CONDITIONS

MLAST = NPRNT*M1MAX + 1
DO 1 K = 1, 3
L = 0
DO 2 I = 1, N
DO 2 J = 1, I
L = L + 1
2 H2(I, J, K) = H(L, K, MLAST)
DO 1 I = 1, N
DO 1 J = 1, N
1 H2(I, J, K) = H2(J, I, K)
L = 1
DO 3 I = 1, N
DO 3 J = 1, I
L = L + 1
3 P2(I, J) = P(L, MLAST)
DO 4 I = 1, N
DO 4 J = 1, N
4 P2(I, J) = P2(J, I)

ZERO ALL AS, BS

DO 7 I = 1, 22
BS(I) = 0.0
DO 7 J = 1, 35
1 AS(I, J) = 0.0

COLUMNS 1 - 6

IN = JINC(1)
DO 8 I = 2, 8
DO 8 K = 1, 3
J1 = 2*K - 1
AS(I, J1) = H2(IN, I-1, K) / B2(IN, I-1)
J2 = J1 + 1
8 AS(I, J2) = -AS(I, J1)
DO 9 I = 9, 15
J1 = I - 7
DO 9 J = 1, C
9 AS(I, J) = -AS(I1, J)
DO 6 I=2,8
  6 BS(I)=1.0 - P2(IN+I-1)/32(IN+I-1)
DO 66 I=9,15
  66 BS(I)=-BS(I-7)
C
C COLUMNS 7
C
AS(1,7)=1.0
DO 10 I=16,22
  10 AS(I,7)=-1.0
DO 91 I=1,22
  91 PRINT94,I,(AS(I,J)+J=1,7)+BS(I)
  94 FORMAT(1H04X15,8E15.6)
  90 FORMAT(1H04X15,7E15.6/10X7E15.6))
C
C COLUMNS 8 - 28
C
DO 11 J=8,28
  11 I=J-6
  12 AS(I,J)=1.0
DO 92 I=1,22
  92 PRINT90,I,(AS(I,J)+J=8,28
C
C COLUMNS 29 - 35
C
DO 19 J=29,35
  19 L=J-28
  12 L+1
  13 L+8
  14 L+15
  19 AS(I+J)=-1.0
  19 AS(I+J)=-1.0
  19 AS(I+J)=-1.0
DO 93 I=1,22
  93 PRINT90,I,(AS(I,J)+J=29,35)
C
C INPUT TO SIMPLX (RANDOM LIBRARY ROUTINE W0C09)
C
INIFIX(1)=4
INIFIX(2)=35
INIFIX(3)=22
INIFIX(4)=22
INIFIX(5)=2
INIFIX(6)=1
INIFIX(7)=100
INIFIX(8)=0
TOL(1)=1.0E-5
TOL(2)=1.0E-5
TOL(3)=1.0E-3
TOL(4)=1.0E-10
PRM=0.0
C
C SIMPLX
C
CALL SIMPLX(INIFIX,AS,BS,TOL,PRM,KOUT,ERR,JH,XS,PS,YS,KB,ES)
OUTPUT FROM SIMPLX

IF(KOUT(1)-3)20,21,20

PRINT60,KOUT(1),KOUT(2)
60 FORMAT(5X6I20)
CALL EXIT

PRINT60,(KOUT(I),I=1,7)
PRINT61,(ERR(I),I=1,4)
61 FORMAT(5X6E20.6)
PRINT60,(JH(I),I=1,22)
PRINT61,(XS(I),I=1,22)
PRINT61,(KB(I),I=1,35)

FIND Z'S

MF=INFIX(5)
M=INFIX(4)
DO 22 I=MF,M
J=JH(I)
22 CONTINUE

PRINT62,(J,2S(J),J=1,35)
62 FORMAT(5X12E20.6)

DO 24 I=1,3
I1=2*I-1
I2=I1+1
24 CONTINUE

PRINT63,2S(7)
63 FORMAT(1H04X20HMAXIMUM DEVIATION = + E15.6)

XTAU=CONST(1)
X1AM(1)=CONST(2)
X1AM(2)=CONST(3)
PRINT903,XTAU,ZERLAM,X1AM(1),X1AM(2)
903 FORMAT(1HO/
1 1X11HTHICKNESS = + E18.6 /
2 1X12HALBEDO(X) = + F6.2, 17H + C1*X + C2*X**2,
3 2X3HC1 = + E18.6, 2X3HC2 = + E18.6//)

CNEW APPROXIMATION

M=1
L=U
DO 12 I=1,N
DO 12 J=1,I
L=L+1
```
SUM=P(L,M)
DO 13 K=1,3
13 SUM = SUM + CONST(K)*H(L,K,M)
R(L,M)=SUM
L=0
DO 14 I=1,N
DO 14 J=1,I
L=L+1
14 R2(I,J)=R(I,J)
SIG=0.0
CALL OUTPUT
C
DO 50 M1=1,M1MAX
DO 18 M2=1,NPRNT
M=M+1
L=0
DO 15 I=1,N
DO 15 J=1,I
L=L+1
SUM=P(L,M)
DO 16 K=1,3
16 SUM = SUM + CONST(K)*H(L,K,M)
R(L,M)=SUM
L=0
DO 17 I=1,N
DO 17 J=1,I
L=L+1
17 R2(I,J)=R(I,J)
SIG=SIG + DELTA
50 CALL OUTPUT
C
RETURN
END
```
PROGRAM C.5. DESIGN OF A SLAB

A partial program is listed:

MAIN program

The following subroutines are required from Program C.1:

DAUX subroutine
ALBEDO subroutine
PANDH subroutine
NONLIN subroutine
OUTPUT subroutine

The following subroutine is required from Program C.2:

LINEAR subroutine

The following library routines are required:

MATINV
INTS/INTM
$JOB
2609
$JOB RTINV
$FTC RTINV LIST

COMMON N,RT(I),WT(I),AR(I),PRNT,M1MAX,KMAX,DELTA,XTAU,
1 ZERLAM,XLAM(1),B2(7),R2(7),FLAG,R(28,101),T(1491),SIG,
2 P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
3 H2(7,7,3),CONST(3),NEQ
4 NINC,JINC(7),NOBS

PHASE I

1 READ100,N
PRINT899
PRINT900,N
READ1-01,(RT(I),I=1,N)
PRINT901,(RT(I),I=1,N)
READ1001,(WT(I),I=1,N)
PRINT901,(WT(I),I=1,N)
DO 2 I=1,N
WR(I)=WT(I)/RT(I)
DO 2 J=1,N
2 AR(I,J)=1.0/RT(I) + 1.0/RT(J)

999 FORMAT(1H146X36HRADEITIVE TRANSFER - INVERSE PROBLEM /
1 47X33HUNKNOWN QUADRATIC ALBEDO FUNCTION /
2 47X27HUNKNOWN THICKNESS OF MEDIUM //)
1000 FORMAT(6I12)
200 FORMAT(6I20)
1001 FORMAT(6E12.8)
901 FORMAT(6E20.8)
READ100,NPRNT,M1MAX,KMAX
PRINT900,NPRNT,M1MAX,KMAX
READ1001,DELTA
PRINT901,DELTA
READ1001,XTAU,ZERLAM,XLAM(1),XLAM(2)
PRINT902
PRINT903,XTAU,ZERLAM,XLAM(1),XLAM(2)
902 FORMAT(1H123HPHASE I - TRUE SOLUTION //)
903 FORMAT(1H0/)
1 1X11HTHICKNESS = F10.4 /
2 1X11HALBEO(0) = F6.2*2H ++ F6.2*3HX ++ F6.2*4HX**2 //

PHASE II

4 READ1001,XTAU,ZERLAM,XLAM(1),XLAM(2)
K=0
PRINT904,K
PRINT903,XTAU,ZERLAM,XLAM(1),XLAM(2)

READ1000,NINC
PRINT900,NINC
READ1000,(JINC(I),I=1,NINC)
PRINT900,(JINC(I),I=1,NINC)
NOBS=NINC*N
PRINT900,NOBS

C
DO 6 I=1,NINC
   J=JINC(I)
   READ1U01*(B2(J,K),K=1,N)
6 PRINT901*(B2(J,K),K=1,N)
C
CALL NONLIN
C
904 FORMAT(1H1, 13HAPPROXIMATION, 13/)
C
     QUASILINEARIZATION ITERATIONS
     DO 5 K1=1,KMAX
      PRINT904,K1
      CALL PANDH
      CALL LINEAR
     5 CONTINUE
C
C
READ1U00,IGO
GO TO (1,4),IGO
END
ENTRY RTINV
     7
25446046E-01 12923441E-00 29707742E-00 500000000E 00 70292258E 00 087076559E 00 32
9745396E 00
64742484E-01 13985269E-00 19091502E-00 20897958E-00 19091502E-00 13985269E-00 00 2
64742484E-01
10 10 5
1.0 0.5 2.0 -2.0
1.0 0.5 2.0 -2.0
1.0 1.7
0.28 0.144 0.333 0.505 0.621 0.689
$16SYS ENDJOB
APPENDIX D

PROGRAM FOR RADIATIVE TRANSFER:

ANISOTROPIC SCATTERING
PROGRAM D.I. PROGRAM FOR THE CALCULATION OF REFLECTED INTENSITIES

The complete program is listed:

MAIN program
LGNDRP subroutine
CTAU subroutine
DAUX subroutine
DCTNRY subroutine
SSTART subroutine
OUTPUT subroutine

The following library routine is required:
INTS/INTM
### RADIATIVE TRANSFER MAIN PROGRAM

**COMMON**
- T(7263), S(11,11,10), G(11,11,10), ZINT(20,10,10),
- 1 P(11,11,10), PK(11,11,10), PSI(11,11,10), XL(10,10), W(10,10),
- 2 FAC(22), FACT(22,22), SGN(22), DEL(11), GDEL(11), C(11), C(11,11,11),
- 3 A(10,10), DELPHI(20), THETA(10),
- 4 LMD(10), NQUAD, MMAX, NFLAG, XFLAG, NPRINT,
- 5 DELTAU, OMEGA, QLBEDO, NPHI, FLUX, MPRNT

**VARIABLES**
- TAU (OPTICAL THICKNESS, IN MEAN FREE PATHS)
- S(M,K,L) (M-TH SCATTERING COMPONENT FOR MU=XL(K,NQUAD) AND MU-ZERO = XL(L,NQUAD))
- XL(K,NQUAD) (<K-TH ROOT OF NQUAD-DEGREE LEGENDRE POLYNOMIAL)
- WT(K,NQUAD) (CORRESPONDING CHRISTOFFEL WEIGHT)
- P(M,I,K) (I-TH DEGREE, (M-1)TH ORDER ASSOCIATED LEGENDRE FUNCTION EVALUATED AT X=XL(K,NQUAD))
- ZINT(J,K,L) (SCATTERED INTENSITY FOR MU=XL(K,NQUAD), MU-ZERO = XL(L,NQUAD), AND DELPHI(J))
- DELPHI(J) (J-TH AZIMUTH ANGLE, NPHI ANGLES ARE INPUT DEGREES)
- THETA(K) (POLAR ANGLE OF OUTPUT, THETA(K) = ARC COSINE(MU), WHERE MU=XL(K,NQUAD))
- C(I) (I-TH FOURIER COEFFICIENT IN EXPANSION OF PHASE FUNCTION)
- OMEGA (ALBEDO OF SINGLE SCATTERING)
- QLBEDO (ALBEDO OF EARTH'S SURFACE)

**CONSTANTS**
- NQUAD (DEGREE OF GAUSSIAN QUADRATURE)
- MM-1 (DEGREE OF FOURIER EXPANSION)
- N1 (NUMBER OF DIFFERENTIAL EQUATIONS)
  - NEG=MM*1*NQUAD*(NQUAD+1)/2
- NPRINT (NUMBER OF INTEGRATIONS PER PRINT INTERVAL)
- TAUONE (INITIAL TAU)
- TAUWO (FINAL TAU)
- DELTAU (INTEGRATING GRID SIZE)
- FLUX (INCIDENT FLUX / PI)

**FLAGS**
- NFLAG=1 (NQUAD AND MM FOR THIS PROBLEM ARE NOT THE SAME)

---

**RADIATIVE TRANSFER**

**DIFFUSE REFLECTION FROM A TWO-DIMENSIONAL FLAT LAYER**

**INTEGRATION OF SCATTERING COEFFICIENTS S(M,K,L)**

**INTENSITY IS COMPUTED FROM THE S COEFFICIENTS**

**MEANINGS**
- AMSC290 C TAU OPTICAL THICKNESS, IN MEAN FREE PATHS.
- ANISO300 C INTENSITY IS COMPUTED FROM THE S COEFFICIENTS.
- AMSC290 C TAU OPTICAL THICKNESS, IN MEAN FREE PATHS.
- ANISO300 C INTENSITY IS COMPUTED FROM THE S COEFFICIENTS.
- AMSC290 C TAU OPTICAL THICKNESS, IN MEAN FREE PATHS.
- ANISO300 C INTENSITY IS COMPUTED FROM THE S COEFFICIENTS.
- AMSC290 C TAU OPTICAL THICKNESS, IN MEAN FREE PATHS.
- ANISO300 C INTENSITY IS COMPUTED FROM THE S COEFFICIENTS.
- AMSC290 C TAU OPTICAL THICKNESS, IN MEAN FREE PATHS.
- ANISO300 C INTENSITY IS COMPUTED FROM THE S COEFFICIENTS.
AS IN THE PREVIOUS PROBLEM, SO THAT CERTAIN VARIABLES MUST BE EVALUATED AGAIN.

NFLAG = 2  OTHERWISE

KFLAG = 1  C(I) ARE ALL CONSTANT
KFLAG = 2  C(I) ARE FUNCTIONS OF TAU

LFLAG = 1  OMEGA IS CONSTANT
LFLAG = 2  OMEGA IS A FUNCTION OF TAU

DICTIONARY ARRAYS
FACT, SIGN, DEL

INPUT LEGENDRE ROOTS AND CHRISTOFFEL *EIGHTS
SET UP A DICTIONARY OF CONSTANT COEFFICIENTS

CALL DCTNRY

INPUT PROBLEM CONSTANTS
9 READ 1003, NQUAD, MMAX, NFLAG, KFLAG, LFLAG, NPRNT, N1, MPRNT
IF (NQUAD-2) 9999, 9999, 10
10 READ 1004, TAUONE, TAUTAO, DELTAU, OMEGA, QLBEDO
READ 1004, (C(I), I = 1, MMAX)
NEQ = MMAX * (NQUAD * (NQUAD + 1)) / 2
READ 1006, FLUX, NPHI, (DELPHI(J), J = 1, NPHI)
PRINT 2001, NQUAD, MMAX, N1, NEQ, NPRNT
PRINT 2002, TAUONE, TAUTAO, DELTAU, OMEGA, QLBEDO
PRINT 2003, (C(I), I = 1, MMAX)
PRINT 2004, FLUX, NPHI, (DELPHI(J), J = 1, NPHI)

GO TO (16, 17) * NFLAG

NEW ASSOCIATED LEGENDRE POLYNOMIALS AND OTHER VARIABLES

16 CALL LGNDRP
DO 18 J = 1, NQUAD
  AX = wT(J, NQUAD) / XL(J, NQUAD)
DO 18 M = 1, MMAX
DO 18 I = M, MMAX
18 PW(M+1, J) = P(M, I, J) * AX

DO 19 K = 1, NQUAD
XI = 1.0 / XL(K, NQUAD)
DO 19 L = 1, K
  A(L, K) = XI + 1.0 / XL(L, NQUAD)
19 A(K+1, L) = A(L, K)

GO TO (129, 21) * KFLAG

THE COEFFICIENTS, C(I), AND THUS K, ARE CONSTANTS

20 DO 22 M = 1, MMAX
  DO 22 I = M, MMAX
    J = 1 + 1
    JT = 1 + 1
    JT = M + 2
22  CK(I*M)=C(I)*FACT(JC*JT)*SGN(J)
C
C  INITIAL INTEGRATING STEP
21  CALL SSTART
C
20  CALL INTS(T*NEG,N1,0,0,0,0,0)
C
700  J=3
710  DO 25 M=1,MMAX
720  DO 25 K=1,NQUAD
730  DO 25 L=1,K
740     J=J+1
750     S(M*K*L)=T(J)
760     Q(M*K*L)=S(M*K*L)
770  25  S(M,L,K)=S(M,K,L)
C
800  COMPUTE INTENSITIES AND OUTPUT
810  CALL OUTPUT
820  C
830  C  GENERAL INTEGRATING STEPS
840  DO 31 M=1,MPRINT
850  27  DO 26 N=1,NPRINT
860  CALL INTM
870  C
880  J=3
890  DO 26 M=1,MMAX
900  DO 26 K=1,NQUAD
910  DO 26 L=1,K
920     J=J+1
930     S(M*K*L)=T(J)
940     S(M,L,K)=S(M,K,L)
950  26  CALL OUTPUT
960  C
970  28  DO 29 M=1,MMAX
980  DO 29 K=1,NQUAD
990  DO 29 L=1,K
1000     G=S(M,K,L)-G(M,K,L)
1010  IF(ABS(G)<.000000 129,29,30
1020  29  CONTINUE
1030  GO TO 9
1040  30  DO 31 M=1,MMAX
1050  DO 31 K=1,NQUAD
1060  DO 31 L=1,K
1070     Q(M*K*L)=S(M,K,L)
1080  31  CALL OUTPUT
1090  C
1100  999  CALL EXIT
1110  C
1400  1003  FORMAT(12I6)
1401  1004  FORMAT(6E12.6)
1402  1005  FORMAT(1E10.6,5E10.6/7E10.6)
1403  2001  FORMAT(1H149X18RADIA/TRANSFER///
1404  26X14*46H=POINT GAUSSIAN QUADRATURE
1405  26X14*46H=TERM EXPANSION OF PHASE FUNCTION
1406  30X* 7H NFLAG=,11, 9H* KFLAG=,11, 9H* LFLAG=,11*1H* /...
1407  26X14*46H INTEGRATIONS PER PRINT INTERVAL
1408  26X14*46H=INTEGRATION OPTION WORD
1409  / ANIS1210
1410  / ANIS1220
1411  / ANIS1230
1412  / ANIS1240
1413  / ANIS1250
1414  / ANIS1270
1415  / ANIS1280
1416  / ANIS1290
1417  / ANIS1300
1418  / ANIS1310
1419  / ANIS1320
1420  / ANIS1330
1421  / ANIS1340
1422  / ANIS1350
1423  / ANIS1360
1424  / ANIS1370
1425  / ANIS1380
1426  / ANIS1390
1427  / ANIS1400
1428  / ANIS1410
1429  / ANIS1430
1430  / ANIS1440
1431  / ANIS1450
1432  / ANIS1460
1433  / ANIS1470
1434  / ANIS1480
1435  / ANIS1490
1436  / ANIS1500
1437  / ANIS1510
1438  / ANIS1520
1439  / ANIS1530
1440  / ANIS1540
1441  / ANIS1550
1442  / ANIS1560
1443  / ANIS1570
1444  / ANIS1580
1445  / ANIS1590
1446  / ANIS1600
1447  / ANIS1610
1448  / ANIS1620
1449  / ANIS1630
1450  / ANIS1640
1451  / ANIS1650
1452  / ANIS1660
1453  / ANIS1670
1454  / ANIS1680
1455  / ANIS1690
1456  / ANIS1700
1457  / ANIS1710
1458  / ANIS1720
1459  / ANIS1730
1460  / ANIS1740
1461  / ANIS1750
1462  / ANIS1760

26XU.46H-PALESTINE/GAUSSIAN
26X14,46H-TERMS EXPANSION OF PHASE FUNCTION
30X, 7H NFLAG=,11, 9H* KFLAG=,11, 9H* LFLAG=,11*1H* /...
26X14*46H INTEGRATIONS PER PRINT INTERVAL
26X14*46H=INTEGRATION OPTION WORD

1440 FORMAT(1E10.6,5E10.6/7E10.6)
6 26X14.46M DIFFERENTIAL EQUATIONS / 7 26X14.46M XPRNT(/)

20-2 FORMAT(15X26M) INTEGRATION RANGE IN TAU IS F9.4/ 3M TO F9.4/
1 30X18M GRID SIZE IS F7.4/
2 30X18M ALBEDO OF SINGLE SCATTERING IS F7.4/
3 30X29M ALBEDO OF EARTH'S SURFACE IS F7.4/

20-3 FORMAT(33X6F9.4)
20-4 FORMAT(33X17M) INCIDENT FLUX IS F9.4/
1 26X14.46M DELTA PHI ANGLES ARE /
2 (33X6F9.4)
END

**LGF** LUNDRP REF

**SUBROUTINE** LUNDRP

**CASSCF**

**COMMON** (X(150),S(11,10,14),A(11,10)),LINT(20,10,10)
1 P(11,11,14) P(11,11,10) P(11,11,14) XL(10,10,10) LINT(10,10,10)
2 FACT(22) FACT(22) SGNHDEL(1) ODEL(1) C(11,11,11)
3 A(10,10,JDELPH(I,20))
4 YN=TNQOD(4) YN=TNQOD(4) YN=TNQOD(4)
5 DELTAO=O须DAO YN=TNQOD(4) YN=TNQOD(4)

**DO 10** N=1,1+10
X=X(X+1)

**DO** C=1,1+10
P(I+1,K)=1.0

**DO** C=1,1+10
P(I+2,K)=X

**IF** (WMAX-4) 10,1,10

**DO** C=1,1+10
R(I+K)=T(N-X*P(I+1,N-1,K))-T(N-X*(P(I+1,N-2,K)))/FN

**DO** C=1,1+10

**DO** C=1,1+10
R(N)=N-V

**DO** C=1,1+10
R(N+K)=T(N-X*P(I+1,N-1,K)-S(N-P(M,N-2,K))/RN

**DO** C=1,1+10

**DO** C=1,1+10
\[ V_1 = 2 \cdot N + 1 \]
\[ M_2 = N + 1 \]
\[ P(N, N + K) = (C \cdot 5 \cdot P(2, 2, K)) \cdot N \cdot \text{FACT}(M_1, M_2) \]

\[ M = 2 \]
\[ N = 1 \]
\[ P(N, N + K) = (C \cdot 5 \cdot P(2, 2, K)) \cdot N \cdot \text{FACT}(M_1, M_2) \]

C
\[ \text{CONTINUE} \]
C
\[ \text{CONTINUE} \]
RETURN
END

*IBFIC CTAU
SUBROUTINE CTAU
RETURN
END

*IBFIC DAUX
SUBROUTINE DAUX
RETURN
END

COMMON T(7263), S(11,10,10), U(11,10,10), LNT(20,10,10),
1 P(11,11,10), PX(11,11,10), PSI(11,11,10), XL(10,10), AT(10,10),
2 FACT(22), FACT(22), SGN(22), ODEL(11), ODEL(11), C(11), CK(11,11),
3 A(11,10), DELPHI(20), THETA(10),
4 MMN, NJ, NAU, MMAX, NFLAG, KFLAG, LFLAG, NPRINT, NTRAONE, TAU, TAUK,
5 DELTAU, UMEGA, GLSDEO, NEU, PHI, FLUX, NPRINT

CALL CTAU
GO TO (1,2,3)
2 DC 22 MM = 1, MMAX
DO 22 1 = M + 1, MMAX
JU = 1 - M + 1
JT = 1 + M - 1
J = I + M - 2 + 1
22 CK(1, M) = C(I) \cdot \text{FACT}(JJ, JT) \cdot \text{SGN}(J)
C
1 CALL ALBEDO
3 L = 3
DO 14 M = 1, MMAX
DO 14 K = 1, NQUAD
DO 14 J = 1, K
L = L + 1
S(M, K, J) = T(L)
14 S(M, K, J) = S(M, K, J)
DO 5 M = 1, MMAX
DO 5 K = 1, NQUAD
SUM = CM
DO 5 J = 1, NQUAD
5 SUM = SUM + S(M, K, J) \cdot Pn(M, 1, J)
J = J + M - 2 + 1
5 PSI(M, 1, K) = Pn(M, 1, K) \cdot C \cdot SGN(J) \cdot \text{SUO}/\text{DEL}(M)
C
DO 7 M = 1, MMAX
7 ODEL(M) = OMEGA*DEL(M)
C
J = N - O + 3
DO 6 M = 1, MMAX
DO 6 K = 1, NQUAD
DO 6 L = 1, K
SUBROUTINE DCTNRY

COMMON T(7263),S(11*10+1),C(11*10+10),U(20*10+10), M(11*20+10)


2. SGN(22),DEL(11),DEL(11),DEL(11),DEL(11),DEL(11),DEL(11),DEL(11),DEL(11), DEL(11),DEL(11),DEL(11)

3. A(11*10+1),DELPHI(20),THETA(10)

4. M=M+1

5. DELTA,J,CMEG,QLSC,NEPH,FLUX,MPRN

INPUT ROOTS AND WEIGHTS

DO 1 I=2,10
1 READ 100,N1,(*T(J,N),J=1,N)

DO 2 I=2,10
2 READ 100,N1,(*W(J,N),J=1,N)

100 FORMAT(112/(6E12.9))

SET UP DICTIONARY

SINGLE FACTORIALS

FAC(1)=1,J
FAC(2)=1,J
FAC(3)=2,J

DO 3 J=4,22
3 FJ=J-1
3 FAC(J)=FJ*FAC(J-1)

DOUBLE FACTORIALS

DO 4 J=1,22
4 FACT(J)={-1)**(J+1)

DO 5 M=1,11
5 L=J+1
5 MJ=MOD(J,2)+1
5 GO TO (6,7,MJ)
6 SGN(L)=1,J
6 GO TO 5
7 SGN(L)=-1,J
7 CONTINUE

DELTA,J,CMEG,QLSC,NEPH,FLUX,MPRN

DC 9 I=M-MAX
DC 8 M=M+11
8 DEL(M)=2.*
C
RETURN
DCTA0600
END
DCTA0620
SSTA0010
SSTA0030
SUBROUTINE SSTART
COMMON T(7263),S(11,10,10),Q(11,10,10),ZINT(20,10,10),
1 P(11,11,1),P*(11,11,10),PSI(11,11,10),XL(10,10),WT(10,10),
2 FAC(22),FACT(22,2),SGN(22),DEL111,ODEL111,C(11),CK(11,11),
3 A(1,10),DELPHI(20),THETA(10)
4 MMON,NGUAD,MVX,NFLAG,KFLAG,LFLAG,MPRT,N1,TAUONE,TAUTWO,
5 DELTAU,Omega,QLBEDD,NEQ,NPHI,FLUX,MPRT
DC 23 I=1,7263
T(1)=Q
T(2)=TAUONE
T(3)=DELTAN
DC 24 I=1,MVX
DO 24 J=1,NGUAD
DO 24 K=1,NGUAD
24 S(I,J,K)=0.
C
RETURN
C
SUBROUTINE OUTPUT
DIMENSION CMD(20,11)
COMMON T(7263),S(11,10,10),Q(11,10,10),ZINT(20,10,10),
1 P(11,11,1),P*(11,11,10),PSI(11,11,10),XL(10,10),WT(10,10),
2 FAC(22),FACT(22,2),SGN(22),DEL111,ODEL111,C(11),CK(11,11),
3 A(1,10),DELPHI(20),THETA(10)
4 MMON,NGUAD,MVX,NFLAG,KFLAG,LFLAG,MPRT,N1,TAUONE,TAUTWO,
5 DELTAU,Omega,QLBEDD,NEQ,NPHI,FLUX,MPRT
C
IF (LFLAG-1)1.6
STORE ANGLES AND COSINES OF M DELTA PHI
C
1 QFLUX=0.25*FLUX
DO 3 K=1,NGUAD
CSTHET=XL(K,NGUAD)
3 THETA(K)=ARCS(CSTHET) *57.2957795
C
DC 5 J=1,NPHI
C
DELPHI(J)=DELPHI(J)*1.74532526E-01
DO 4 M=1,MMAX
FM=M-1
FMDFM*DELPHI(J)
4 CMD(J,M)=COS(FMD)
PRINT1051,J,M,CMD(J,M)
1051 FORMAT(2I5,16.8)
5 CONTINUE
LFLAG=2

6 CALL ALBEDO
TAU=T(2)

OUTPUT S
PRINT10J,OMEGA,ALBEDO,TAU
DO 10 I=1,NQUAD
PUNCH2JO+1
PRINT131,1
DO 10 M=1,MMAX
MM=M-1
PRINT105,M,M,S(M,J,1),J=1,NQUAD)
105 FORMAT(3X12.1F10.6)
10 PUNCH2J1, (S(M,J,1),J=1,NQUAD)
200 FORMAT(3I12)
201 FORMAT(6E12.8)

OUTPUT I
PRINT 106,OMEGA,ALBEDO,TAU,(K,K=1,NQUAD)
DO 16 J=1,NPHI
DO 16 K=1,NQUAD
DO 16 L=1,K
SUM=0.0
DO 14 M=1,MMAX
SUM=SUM+CMD(J,M)*S(M,K,L)
14 SUM=SUM*3FLUX
ZINT(J,L,K)=SUM/XL(L,NQUAD)
16 ZINT(J,K,L)=SUM/XL(K,NQUAD)

DO 20 L=1,NQUAD
J=1
PRINT 131,J,L, (ZINT(J,K,L),K=1,NQUAD)
19 IF (NPHI=1)20U+20+19
20 J=2,NPHI
20 PRINT 131,J,L, (ZINT(J,K,L),K=1,NQUAD)
20 CONTINUE

100 FORMAT(1H124X29HSCATTERING COEFFICIENTS, S(M) /
23X7HOMEQA = F5 2 5H Q = F5 2 5H Z = F5 2
10X56H FOR THE FOLLOWING POLAR ANGLES OF INCIDENCE AND REFLECTION
OUTP0560
3 )
OUTP0570
101 FORMAT(1H026X5HANGLE,4X7HDEGREES,4X6HCOSINE/(128X12,F12.2+F11.4)) OUTP0580
102 FORMAT(1H0/2X8HINCIDENT,2X21HREFLECTED POLAR ANGLE/3X5HANGLE/
1 (2X1011)) OUTP0600
103 FORMAT(1H0/2X12,10F10.6) OUTP0610
104 FORMAT(1H0/3X5THNOTE. EACH FIGURE ABOVE CORRESPONDS TO AN INCIDEN
1 IT POLAR ANGLE, A/4X6HREFLECTED POLAR ANGLE, AND A TERM IN THE EXPON
2 SION OF THE S FUNCTION. // OUTP0640
3 3X6BHEACH FIGURE ON THE NEXT PAGE CORRESPONDS TO AN INCIDENT POLAR
4 ANGLE, /4X55HA REFLECTED POLAR ANGLE, AND A CHANGE IN AZIMUTH AN
5 GLE.)
OUTP0670
106 FORMAT(1H126X24HSCATTERED INTENSITIES. 1 /
1 23X7HOMEQA = F5 2 5H Q = F5 2 5H Z = F5 2 /
2 1H0/1X1011)) OUTP0690
108 FORMAT(5X10F10.6) RETURN OUTP0700
END OUTP0720
OUTP0730
ENTRY MAIN
2 21132486E-0078867914E 00
021
3 11270166E-0050000000E 0088729834E 00
022
4 69431845E-0130000948E-0066999052E 0093056816E 00
023
5 46910081E-0123076534E-00500C 0000E 0076923466E 0095306992E 00
024
6 33765245E-0116939531E-003606904E-006193960E 0083060469E 0096623476E 00
025
7 25446046E-0112923441E-0029707742E-0050000000E 0070292258E 0087076559E 00
026
8 97455396E 00 OUTP0800
027
9 19855071E-0110166676E-0023723390E-004828268E-0059171732E 0076276620E 00
028
0 8933324E-0098014493E 00
029
1 15919883E-0181894445E-00119331428E-0033787329E-0050000000E 0066212671E 00
030
2 06668372E-0091801555E 0098408012E 00
031
3 13046738E-0167468315E-00116029522E-0026330231E-0042556283E-0057443717E 00
032
4 71669759E 0083970478E 0093253168E 0098695327E 00
033
5 0000000E 0085000000E 00
034
6 27777778E-0044444444E-0027777778E-00
035
7 17392742E-0037607257E-0032607257E-0017392742E-00
036
8 11846343E-0023931433E-0028444444E-0073931433E-0011846343E-00
037
9 85662244E-0118038078E-0023959569E-0023959569E-0018038078E-0085662244E-01
038
10 64742484E-0113952698E-00190915028E-002897958E-00190915028E-0013985269E-00
039
11 64742484E-01
040
END
<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.0614270E-01</td>
<td>1.11119051E-01</td>
<td>1.5685332E-01</td>
<td>1.0134189E-00</td>
<td>0.18134189E-00</td>
<td>0.15685332E-00</td>
</tr>
<tr>
<td>1.1119051E-01</td>
<td>1.05061427E-01</td>
<td>1.0134189E-00</td>
<td>0.18134189E-00</td>
<td>0.15685332E-00</td>
<td>0.15511968E-00</td>
</tr>
<tr>
<td>0.04637194E-01</td>
<td>0.190324079E-01</td>
<td>0.11303535E-01</td>
<td>0.15617353E-01</td>
<td>0.16511968E-00</td>
<td>0.15617353E-00</td>
</tr>
<tr>
<td>0.15305335E-01</td>
<td>0.090324079E-01</td>
<td>0.146637194E-01</td>
<td>0.04637194E-01</td>
<td>0.04725674E-01</td>
<td>0.04725674E-01</td>
</tr>
<tr>
<td>3.358669E-01</td>
<td>0.174725674E-01</td>
<td>0.119954317E-01</td>
<td>0.04725674E-01</td>
<td>0.1333355669E-01</td>
<td>0.04725674E-01</td>
</tr>
<tr>
<td>1.3463335E-01</td>
<td>0.1954317E-01</td>
<td>0.074725674E-01</td>
<td>0.01333355669E-01</td>
<td>0.04725674E-01</td>
<td>0.04725674E-01</td>
</tr>
<tr>
<td>0.0</td>
<td>1.0</td>
<td>1.0</td>
<td>2.0</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>3.0</td>
<td>1.0</td>
<td>90.0</td>
<td>180.0</td>
<td>2.0</td>
</tr>
</tbody>
</table>
APPENDIX E

PROGRAMS FOR NEUTRON TRANSPORT
PROGRAM E.1. PRODUCTION OF INTERNAL MEASUREMENTS

The complete program is listed:

MAIN program

DAUX subroutine

The following library routine is required:

INTS/INTM
$JOB 2609, DYNNUK, 0160, 5, 100, 100, C
$JOB DRM, MAP
$IBM C MAIN REF
DIMENSION NPNCH(20), Z(300)
COMMON T(27), A(20), X(20), IFLAG, AA, U(300), V(300), NSLABS

1 READ(5, 100) NPRINT, MPRINT, NSLABS, NGRIDS, NOS
WRITE(6, 90) NPRINT, MPRINT, NSLABS, NGRIDS, NOS
READ(5, 100)(NPCH(I), I=1, NOS)
WRITE(6, 90)(NPCH(I), I=1, NOS)
READ(5, 101) DELTA, AA
WRITE(6, 91) DELTA, AA
READ(5, 101)(X(I), I=1, NSLABS)
WRITE(6, 91)(X(I), I=1, NSLABS)

C REFLECTION COEFFICIENT
RC = SIN(AA) / COS(AA)

C U AND V FLUXES
T(2) = -DELTA
T(3) = -DELTA
T(4) = RC
T(5) = 1.0
CALL INTS(T, 2, 2, 0, 0, 0, 0, 0, 0)
WRITE(6, 94)
WRITE(6, 95) T(2), T(4), T(5)

C N=0
DO 5 I=1, NSLABS
DO 5 J=1, NGRIDS
CALL INTM
WRITE(6, 95) T(2), T(4), T(5)
N = N + 1
U(N) = T(4)
Z(N) = T(2)

C PUNCH U AND V OBSERVATIONS
PRINT 97
DO 6 M=1, NSLABS
DO 6 I=1, NOS
N = (M-1)*NGRIDS + NPNCH(I)
PUNCH 96, Z(N), U(N)
PRINT 96, Z(N), U(N)

GO TO 1
C
100 FORMAT(6I12)
101 FORMAT(5E12.8)
90 FORMAT(1H06I20)
91 FORMAT(1H06E20.8)
92 FORMAT(/19X1HX, 16X 4HR(X), 11X1HA/)
93 FORMAT(F20.4, E20.8, F12.4)
94 FORMAT(/19X1HX, 16X 4HU(X), 16X4HV(X), 11X1HA/)
95 FORMAT(F20.4, E20.8, F12.4)
96 FORMAT(F12.2, E12.8)
97 FORMAT(//)
SUBROUTINE DAX
COMMON T(20), A(20), X(20), IFLAG, A(100), V(100), NSLABS
C
4 T(6) = AA * T(5)
T(7) = -AA * T(4)
RETURN
END

ENTRY MAIN
  1   100 10 10 3
  2  10  8
0.01 0.2 0.3 0.4 0.5 0.6
0.1 0.2 0.3 0.4 0.5 0.6
0.7 0.8 0.9 1.0
PROGRAM E.2. TWO DIMENSIONAL DYNAMIC PROGRAMMING FOR THE
DETERMINATION OF ABSORPTION COEFFICIENTS

The complete program is listed:

MAIN program
INTERP subroutine
DAUX subroutine
INTR subroutine

The following library routines are required:

BET
INTS/INTM
$JOB 2890,DPNT1,K0160,100,10,1C^,C
$IBJOB MAP
$IBCFTC MAIN REF
COMMON T(27),AA,NA,A(51),DA,NC,C(151),DC,NE,E(51),DF,NSLABS,B(51),
1 NOS,IGRID(I100),NGRDSB,MOB,(100),NTOBS,Z(100),W(100),DELTA,MINT,
2 F(51,51),S(51),H(51,51),U(100)

C
1 READ(5,100)NA,A(1),DA
WRITE(6,90)NA,A(1),DA
DO 2 I=2,NA
2 A(I)=A(I-1)+DA

C
READ(5,100)NC,C(1),DC
WRITE(6,90)NC,C(1),DC
DO 3 I=2,NC
3 C(I)=C(I-1)+DC

C
READ(5,100)NE,E(1),DE
WRITE(6,90)NE,E(1),DE
DO 4 I=2,NE
4 E(I)=E(I-1)+DE

C
READ(5,100)NSLABS,(B(I),I=1,NSLABS)
WRITE(6,91)NSLABS,(B(I),I=1,NSLABS)

C
READ(5,101)NOS,(IGRID(I),I=1,NOS)
WRITE(6,91)NOS,(IGRID(I),I=1,NOS)

C
READ(5,101)NGRDSB
WRITE(6,91)NGRDSB
N=0
DO 5 I=1,NSLABS
5 N=N+1
NOS=1
MOBS(N)=(I-1)*NGRDSB+IGRID(J)
NTOBS=NOS*NSLABS
WRITE(6,91)MOBS(I),I=1,NTOBS

C
READ(5,102)(Z(I),W(I),I=1,NTOBS)
WRITE(6,92)(Z(I),W(I),I=1,NTOBS)

C
READ(5,102)DELTA
WRITE(6,92)DELTA

C
MINT=NGRDSB

C
STAGE 1

C
NSTAGE=1
WRITE(6,93)NSTAGE

C
DO 10 I=1,NC
DO 10 J=1,NE
AA=ATAN2(C(I),E(J))/B(1)
F(I,J)=0.0
DO 6 K=1,NOS
6 F(I,J)=F(I,J)+(SIN(AA*Z(K))-W(K))**2
CP=0.0
EP=0.0
10 WRITE(6,94)C(I),E(J),AA,CP,EP,F(I,J)
DO 50 NSTAGE=2,NSLABS
WRITE(6,93) NSTAGE
DO 40 IC=1,NC
DO 40 JE=1,NE
C
DO 30 IA=1,NA
BX=B(NSTAGE)
AA=A(IA)
T(2)=BX
T(3)=-DELTA
T(4)=C(IC)
T(5)=E(JF)
CALL INTS(T,2,2,0,0,0,0,0,0,0,0)
DO 20 M=1,MINT
CALL INTM
20 U(M)=T(4)
D=0.0
J=(NSTAGE-1)*NOS
DO 21 I=1,NOS
M=IGRID(I)
J=J+1
21 D=D + (U(M)-W(J))**2
CP=T(4)
EP=T(5)
CALL INTERP(CP,EP,FI)
30 S(IA)=D+FI
C
C
MIN S OVER A
MINA=1
SMIN=1.0E+20
DO 31 IA=2,NA
I=IA
IF(S(I)-SMIN)32,31,31
32 SMIN=S(I)
MINA=I
AA=A(I)
31 CONTINUEF
H(IC,JE)=SMIN
C
DO 50 IC=1,NC
DO 50 JE=1,NE
50 F(IC,JE)=H(IC,JE)
C
GO TO 1
C
100 FORMAT(112,5E12.8/(6E12.8))
90 FORMAT(1H0120,5E20.8/(6E20.8))
101 FORMAT(6112)
91 FORMAT(1H06120)
102 FORMAT(2E12.8)
92 FORMAT(1H06E20.8)
93 FORMAT(1H1 9HSTAGE N =, 13//18X2HC1,18X2HC2,19X1HA,17X3HC1','
+1 17X3HC2',12X8HF(C1,C2))
94 FORMAT(6E20.8)
END
$IBFTC INTERP REF
SUBROUTINE INTERP(X,Y,ANS)
COMMON T(27),AA,NA,A(51),DA,NC,C(101),DC,NE,E(51),DE,NSLABS,B(51),NOS,IGRID(100),NGRDSB,MOBS(100),NTOBS,Z(100),W(100),DELA,MINT,
1 F(51,51),S(51),H(51,51)
C TWO-DIM. INTERPOLATION
C FIND I1,I2, I.E., X1, X2
DO 1 I=2,NC
I1=I
I2=I-1
X1=C(I1)
X2=C(I2)
IF(BET(X1,X,X2,MM))1,2,2
1 CONTINUE
ANS=1.0E+20
RETURN
C C
C FIND J1,J2, I.E., Y1, Y2
2 DO 11 J=2,NE
J1=J
J2=J-1
Y1=E(J1)
Y2=E(J2)
IF(BET(Y1,Y,Y2,MM))11,12,12
11 CONTINUE
ANS=1.0E+20
RETURN
C C
C FIND F(X,Y1)=G1
12 F1=F(I1,J1)
F2=F(I2,J1)
DX=X2-X1
D=X-X1
CALL INTR(F1,F2,DX,D,G1)
C C
C FIND F(X,Y2)=G2
F1=F(I1,J2)
F2=F(I2,J2)
CALL INTR(F1,F2,DX,D,G2)
C C
C FIND F(X,Y)=ANS
DY=Y2-Y1
D=Y-Y1
CALL INTR(G1,G2,DY,D,ANS)
RETURN
END
SIBFTC DAUX REF
SUBROUTINE DAUX
COMMON T(27),AA,NA,A(51),DA,NC,C(101),DC,NE,E(51),DE,NSLABS,B(51),NOS,IGRID(100),NGRDSB,MOBS(100),NTOBS,Z(100),W(100),DELA,MINT,
1 F(51,51),S(51),H(51,51)
C T(6)=AA*T(5)
T(7)=-AA*T(4)
RETURN
FND
SIBFTC INTR REF
SUBROUTINE INTR(F1,F2,DX,D,G)
C ONE-DIM. INTERPOLATION
G=F1+(F2-F1)*D/DX
RETURN
PROGRAM E.3. ONE DIMENSIONAL DYNAMIC PROGRAMMING FOR THE
DETERMINATION OF ABSORPTION COEFFICIENTS

The complete program is listed:

MAIN program
DAUX subroutine
SUBREF subroutine
SHIFT subroutine
SUBNLV subroutine
SUBDF subroutine

The following library routines are required:

BET
INTS/INTM
SUBFC MAIN

COMMON T(5), NA, A(10), DA, NC, C(1001), DC, NSLABS, B(10), DR, NOS,
1 IGGRID(50), NGRDSB, DELTA, MOBS, NTOBS, A(200), W(200), MINT,
2 IFLAG, AA, CP, EP, CPA, EPA, R(1001), F(1001), NSTAGE, RBIG, TBIG,
3 RP, PO(1001), ALPHA, SMIN, I, MIN, FO(1001)

INPUT

1 READ(5,100)NA, A(1), DA
WRITE(6,90)NA, A(1), DA
DO 2 I=2, NA

2 A(I)=A(I-1)+DA

READ(5,100)NC, C(1), DC
WRITE(6,90)NC, C(1), DC
DO 3 I=2, NC

3 C(I)=C(I-1)+DC

READ(5,100)NSLABS, B(1), DB
WRITE(6,90)NSLABS, B(1), DB
DO 4 I=2, NSLABS

4 B(I)=B(I-1)+DB

READ(5,101)NOS, (IGRID(I), I=1, NOS)
WRITE(6,91)NOS, (IGRID(I), I=1, NOS)

READ(5,100)NGRDSB, DELTA, ALPHA
WRITE(6,90)NGRDSB, DELTA, ALPHA

N=0
DO 5 I=1, NSLABS
DO 5 J=1, NOS

5 MOBS(N)=(I-1)*NGRDSB + IGRID(J)
NTOBS=NOS*NSLABS
WRITE(6,91)MOBS(N), N=1, NTOBS

READ(5,102)(Z(I), W(I), I=1, NTOBS)
WRITE(6,92)(Z(I), W(I), I=1, NTOBS)

MINT=NGRDSB

STAGE 1

NSTAGE=1
WRITE(6,93) NSTAGE
DO 8 IC=1, NC

8 SMIN=1.0E+00
DO 7 IA=1, IC

7 IA=1

FIND CP=V(0)

IFLAG=1
AA=AA+1

T(4)=0.0
T(5)=1.0
T(2)=0.0
T(3)=DELTA
CALL INTS(T, 2, 2, 0, 0, 0, 0, 0, 0, 0)
DO 6 M=1, MINT

6 CALL INTM
CP=C(IC)/T(5)
EP=0.0

CALL INTM
EP = (C(IC) - CP*T(7))/T(5)

COMPUTE D, F
CALL SUBDF
CONTINUE
AA = AMIN
F(IC) = SMIN
IF (F(IC) < 100.0) I6, 15, 15
NC = IC - 1
WRITE(6, 95) NC
GO TO 17

COMPUTE R(C)
CALL SUBREF
R(IC) = RP
WRITE(5, 94) C(IC), AA, CPA, EPA, R(IC), F(IC)
CALL SHIFT
CONTINUE
GO TO 1

FORMAT(I12, 5E12.8/(6E12.8))
FORMAT(1H0120, 5E20.8/(6E20.8))
FORMAT(6112)
FORMAT(1H06120)
FORMAT(2E12.8)
FORMAT(1H0620.8)
FORMAT(1H110HSTAGE N =, I3//19X1HC, 19X1HA, 18X2HCP, 18X2HEP,
16X4HR(C), 16X4HF(C)//)
FORMAT(2F20.6, 4E20.6)
FORMAT(1X18HNUMBER OF STATES =, 13)
END

SUBROUTINE DAUX
COMMON T(51), NA, A(10), DA, NC, C(1001), DC, NSLABS, B(10), DB, NOS,
GRID(50), NGRDSB, DELTA, MOBS:200, NTOLS, Z(200), W(200), MINT,
IFLAG, AA, CP, EP, CPA, EPA, R(1001), F(1001), NSAGE, RBIG, IBIG,
RP, RO(1001), ALPHA, SMIN, AMIN, FO(1001)

GO TO (1, 2, 3, 4), IFLAG

TRANSPORT Eqs. FOR U, V

T(6) = AA*T(5)
T(7) = -AA*T(4)
RETURN

FOR P, H

T(8) = AA*T(5)
T(9) = -AA*T(4)
T(10) = AA*T(7)
T(11) = -AA*T(6)
RETURN

REFLECTION

T(5) = AA*(1.0 + T(4)**2)
RETURN

AND TRANSMISSION
$\textsc{subref}$

SUBROUTINE SUBREF
COMMON 1(51), NA, A(10), DA, NC, C(1001), DC, NSLABS, B(10), DB, NOS,
1 IGRID(50), NGRDSB, DELTA, MOBS(200), NI0BS, N200, N2U0, NMIN,
2 IFLAG, AA, CP, EP, CPA, EPA, R(1001), F(1001), NSTAGE, RBIG, TBIG,
3 RP, RO(1001), ALPHA, SMIN, AMIN, FO(1001)

C
C

1 IFLAG = 3
2 T(2) = 0
3 T(3) = DELTA
4 T(4) = RP
5 CALL INTS(T, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0)
6 DO 1 M = 1, MINT
7 1 CALL INTM
8 RP = T(4)
9 RETURN
10 END

$\textsc{shift}$

SUBROUTINE SHIFT
COMMON 1(51), NA, A(10), DA, NC, C(1001), DC, NSLABS, B(10), DB, NOS,
1 IGRID(50), NGRDSB, DELTA, MOBS(200), NI0BS, N200, N2U0, NMIN,
2 IFLAG, AA, CP, EP, CPA, EPA, R(1001), F(1001), NSTAGE, RBIG, TBIG,
3 RP, RO(1001), ALPHA, SMIN, AMIN, FO(1001)

C
C

DO 1 I = 1, NC
2 RO(I) = R(I)
3 IF(I) = F(I)
4 RETURN
5 END

$\textsc{subnlv}$

SUBROUTINE SUBNLV(CC, IS)
COMMON 1(51), NA, A(10), DA, NC, C(1001), DC, NSLABS, B(10), DB, NOS,
1 IGRID(50), NGRDSB, DELTA, MOBS(200), NI0BS, N200, N2U0, NMIN,
2 IFLAG, AA, CP, EP, CPA, EPA, R(1001), F(1001), NSTAGE, RBIG, TBIG,
3 RP, RO(1001), ALPHA, SMIN, AMIN, FO(1001)

C
C

IS = 1
1 J = 1
2 Z1 = RBIG*C(J)*RO(J)
3 Z2 = C(J) - CC*TBIG
4 DIF1 = Z1 - Z2
5 IF(DIF1)1, 2, 3
6 CP = C(J)
7 RP = RO(J)
8 RETURN
9 DIF1 IS NEG.
10 DO 11 J = 2, NC
11 J2 = J
12 J1 = J - 1
13 Z1 = RBIG*C(J)*RO(J)
14 Z2 = C(J) - CC*TBIG
15 DIF2 = Z1 - Z2
16 IF(DIF2)10, 2, 12
10  DIF1 = DIF2
11  CONTINUE
12  GO TO 13
13  CP = C(J1) + DIF1*DC/(DIF1-DIF2)
14  RP = RO(J1) + (CP-C(J1))*(RO(J2)-RO(J1))/DC
15  RETURN
16  DIF1 IS POS.
17  DIF1 = -DIF1
18  DO 21 J = 2, NC
19  J2 = J
20  J1 = J - 1
21  Z1 = RBIG*C(J)*RO(J)
22  Z2 = C(J) - CC*TBIG
23  DIF2 = Z2 - Z1
24  IF (DIF2) 20, 2, 12
25  DIF1 = DIF2
26  CONTINUE
27  IS = 0
28  RETURN
29  END
30  $IBFTC SUBDF REF
31  SUBROUTINE SUBDF
32  DIMENSION U(20)
33  COMMON T(5), NA, D, NC, CP, Z(100), RO, M, NOS, RC, NC, T (200),
34  IGRID(50), NGRDSB, DELTA, MOBS(200), NTORS, Z(200), K(200), MINT,
35  IFLAG, AA, CP, EP, CPA, EPA, R(1001), F(1001), NSTAGE, RBIG, TBIG
36  IFLAG = 1
37  T(2) = 0.0
38  T(3) = DELTA
39  T(4) = CP
40  CALL INTS(T, 2, 2, 0, 0, 0, 0, 0, 0)
41  DO 1 M = 1, MINT
42  CALL INTM
43  U(M) = T(4)
44  D = 0.0
45  J = (NSTAGE - 1)*NOS
46  DO 2 I = 1, NOS
47  M = IGRID(I)
48  J = J + 1
49  2  D = D + (U(M) - W(J))**2
50  IF (NSTAGE - 1) 3, 3, 6
51  S = 0.0*ALPHA
52  10 IF (S - SMIN) 4, 5, 5
53  SMIN = 5
54  AMIN = AA
55  CPA = CP
56  EPA = EP
57  RETURN
58  INTERPOLATE FOR F(N-1)
59  DO 7 I = 2, NC
60  11 = I - 1
COMPUTE D, F

7 CALL SUBRF
   AA=AMIN
   F(IC)=SMIN

COMPUTE R(IC)

   RP=0.0
   CALL SUBREF
   R(IC)=RP
   WRITE(6,94)C(IC),AA,CPA,EPA,R(IC),F(IC)

8 CALL SHIFT

GENERAL STAGE

DO 13 N=2,NSLABS
   NSTAGE=N
   WRITE(6,93)NSTAGE
   DC '12 IC=1,NC
   SMIN=1.0E+20
   AMIN=0.0
   CPA=0.0
   EPA=0.0
   R(IC)=0.0
   F(IC)=0.0

10 CALL SUBNLV(CC,IS)

DO 11 IA=1,NA

FIND RBIG, TBIG
   AA=A(IA)
   IFLAG=4
   T(2)=0.0
   T(3)=DELTA
   T(4)=0.0
   T(5)=1.0
   CALL INTS(T,2,0,0,0,0,0)
   DO 9 M=1,MINT

9 CALL INTM
   RBIG=T(4)
   TBIG=T(5)

FIND CP=V(N-1), RP=R(V)

CC=C(IC)
   CALL SUBNLV(CC,IS)
   IF(IS)11,11,14

FIND EP=U(N-1)

14 IFLAG=2
   T(2)=0.0
   T(3)=DELTA
   T(4)=1.0
   T(5)=0.0
   T(6)=0.0
   T(7)=1.0
   CALL INTS(T,4,2,0,0,0,0,0,0,0,0,0)
   DO 10 M=1,MINT
I2=1
X1=C(11)
X2=C(12)
IF(SLT(X1,CP +X2,MM))7,8,8
7 CONTINUE
   S=1.0 E+10
   GO TO 10
C 8 F1=FO(11)
   F2=FO(12)
   DX=X2-X1
   G=CP -X1
   FX=F1 + (F2-F1)*G/DX
   S=D::FX
   GO TO 10
END
APPENDIX F

PROGRAMS FOR WAVE PROPAGATION:

MEASUREMENTS OF TRANSIENTS
The complete program is listed:

MAIN program
LAPLAC subroutine
DAUX subroutine
IN1TL subroutine
PANDH subroutine
LINEAR subroutine
NEXT subroutine
OUTPUT subroutine

The following library routines are required:

MATINV
INTS/INTM
$\text{16FIC MAIN LIST} - 273 -$

\text{VIBRATING STRING - LAPLACE TRANSFORMS}$

\text{COMMON T(2511),NT,RT(9),WT(9),UOBS1(9),FORCE(9),UOBS(9),FORCT(9)}

1 .NPRINT,MPRNT,NP1,NTWO,KMAX,LFLAG,N21,NEQ
2 .DELTA,TENSX,A,UPREV(18),UOBSX(9)
3 .TT(9),U18(18,101),P(19),H(19,10),C(10)
4 .MX,FORCTX(9),ATRUE,CSPEED

\text{INPUT}$

1 RE. D100,NT,NPRINT,MPRNT,KMAX
  PRINT90,NT,NPRINT,MPRNT,KMAX
  READ1-1,(RT(I),I=1,NT)
  PRINT91,(RT(I),I=1,NT)
  READ1-1,(WT(I),I=1,NT)
  PRINT92,(WT(I),I=1,NT)
  READ1-1,DELTA,TENSX,A,ATRUE
  PRINT93,DELTA,TENSX,A,ATRUE
  READ1-1,(UOBS1(I),I=1,NT)
  PRINT94,(UOBS1(I),I=1,NT)
  READ1-1,(FORCE(I),I=1,NT)
  PRINT95,(FORCE(I),I=1,NT)
  NP1=NT+1
  NTWO=2*NT
  DO 11 I=1,NT
  11 U(I,1)=0.0
  READ1-1,(U(I,1),I=NP1,NTWO)
  PRINT96,(U(I,1),I=NP1,NTWO)

\text{PRODUCE TRANSFORMS OF OBSERVATIONS}$

2 CALL 'APLAC

\text{GENERATE INITIAL APPROXIMATION}$

3 CALL INITL

\text{SUCCESSIVE APPROXIMATIONS}$

4 DO 5 K=1,KMAX
  PRINT97,K
  CALL PANDH
  CALL LINEAR
5 CALL NEXT
  GO TO 1

\text{FORMAT(6I12)}

\text{FORMAT(6E12.8)}

\text{FORMAT(1H14X,4HNT =,13/5X,7HNPRNT =,13/5X,7HMPRNT =,13/}
  \text{5X,6HKMAX =,13)}

\text{FORMAT(1H04X,5HROOTS/(5X6E20.8))}

\text{FORMAT(1H04X,7HWEIGHTS/(5X6E20.8))}
```fortran
93 FORMAT(1H04X7HDELTA = E16.8,E16.8,
5X08HTENSN = E16.8,
4X7HTENSN = E16.8)
94 FORMAT(1H04X12HOBSERVATIONS/(5X6E20.8))
95 FORMAT(1H04X12HFORCE F(T) /(5X6E20.8))
96 FORMAT(1H04X12HOBSERVATIONS/(5X6E20.8))
97 FORMAT(1H14X13HAPPROXIMATION, 13/) END
$IBFTC LAPLAC LIST
SUBROUTINE LAPLAC
COMMON T(2511), NT, RT(9), W(9), OBS1(9), FORCE(9), OBS(9), FORC(9)
1 *NPRNT, MPRT, NP1, NTWO, KMAX, LFLAG, N21, NEG
2 *DELTA, TENSNA, UPRERV(18), OBSX(9)
3 *TT(9), U(18, 401), P(19), H(19, 10), C(10)
4 *MX, PCR(9), ATRUE, CSPEED
C THE TIMES
C DO 1 I=1,NT
1 TT(I)=-ALOG(RT(I))
C THE TRANSFORMS
C DO 2 IS=1,NT
2 UOBST(IS)=0.0
FORC(IS)=0.0
JJ=IS-1
DO 2 I=1,NT
RW=WT(I)*(RT(I)*JJ)
UOBST(IS)=UOBST(IS)+UOBST1*RW
2 FORC(IS)=FORC(IS)+FORC1*RW
C PRINT10
10 FORMAT(1H04X13X1HT, 11X 9HUOBS(1,T), 16X 4HF(T),
14X H15, 6X14HUOBSTRANS(1,S), 16X 4HF(S) )
DO 3 I=1, NT
3 PRINT11, TT(I), OBS1(I), FORCE(I), OBS(I), FORC(I)
11 FORMAT(5XGL, 6. 2E20.8, 10X, 15, 2E20.8)
C EXACT TRANSFORMS OF OBSERVATIONS
CSPEED=SQRT(ATRUE)
COVERT=CSPEED/TENSN
DO 6 IS=1, NT
S=IS
FORC(1)=1.0/S
6 UOBSTX(IS)=TANH(S/CSPEED)*FORC(IS)*COVERT/S
PRINT98, (UOBSTX(IS), IS=1, NT)
98 FORMAT(1H04X32HEXACT TRANSFORMS OF OBSERVATIONS/(5X6E20.8))
PRINT99, (FORC(1), IS=1, NT)
99 FORMAT(1H04X25HEXACT TRANSFORMS OF FORCE /(5X6E20.8)) RETURN
END
$IBFTC DAUX LIST
SUBROUTINE DAUX
COMMON T(2511), NT, RT(9), W(9), OBS1(9), FORCE(9), OBS(9), FORC(9)
1 *NPRNT, MPRT, NP1, NTWO, KMAX, LFLAG, N21, NEG
2 *DELTA, TENSNA, UPRERV(18), OBSX(9)
3 *TT(9), U(18, 401), P(19), H(19, 10), C(10)
```
4  *MX*FORCTX(9),*ATRUE,*CSPEED
DIMENSION V(N21)
GO TO (100,200,300),LFLAG

C
C
C
100 L=3
   DO 1 IS=1,N21
        L=L+1
   1 V(IS)=T(L)
      L=NEQ+3
      DO 2 IS=1,NT
           L=L+1
           NN=NT+IS
      2 T(L)=V(NN)
      DO 3 IS=1, NT
           L=L+1
           S=IS**2
      3 T(L)=S*V(IS)/T(NEQ+3)
           L=L+1
           T(L)=0.0
           RETURN

C
C
C
200 L=3
    DO 4 IS=1,N21
         L=L+1
    4 V(IS)=T(L)
       M=NEQ+3
       DO 5 IS=1, NT
           M=M+1
           NN=NT+IS
      5 T(M)=V(NN)
      DO 6 IS=1, NT
           M=M+1
           S=IS**2
      6 T(M)=S*(V(IS) - V(N21)*UPREV(IS)/A + UPREV(IS))/A
           M=M+1
           T(M)=0.0
    IF(LFLAG-3) 20, 300, 300
    RETURN

C
C
C
20 DO 0 J=1,NP1

C
DO 7 IS=1,N21
    L=L+1
7 V(IS)=T(L)

C
DO 8 IS=1, NT
    M=M+1
    NN=NT+IS
T(i) = 0.0
T(3) = DELTA
L = NT + 3
DO 2 IS=1,NT
J = NT + IS
L = L + 1
2 T(L) = U(J,1)
   L = L + 1
   T(L) = A

I = 1
N21 = 2*NT + 1
NEQ = N21
CALL INTS(T,NEQ,2,0,0,0,0,0,0)
MX = I
CALL OUTPUT

DO 4 M1 = 1, MPRNT
DO 3 M2 = 1, MPRNT
CALL INTM
I = I + 1
L = 3
DO 3 IS = 1, NTWO
   L = L + 1
3 U(IS,I) = T(L)
   MX = I
4 CALL OUTPUT
RETURN
END

-- 276 --
LFLAG=2
DO 1 I=1,2511
1 T(I)=0
T(3)=DELTA
NEQ=(NT+2)*N21
L=4+N21+NT
T(L)=1
DO 2 I=1,NT
L=L+N21+1
2 T(L)=1
I=1
DO 12 IS=1,NT
12 UPREV(IS)=U(IS,I)
CALL INTS(T,NEQ,2,0,0,0,0,0)
NEQ3=NEQ+3
DO 4 M1=1,NPRNT
DO 3 M2=1,NPRNT
CALL INTM
I=I+1
DO 3 IS=1,NT
3 UPREV(IS)=U(IS,I)
4 CONTINUE
C
L=3
DO 5 IS=1,N21
L=L+1
5 P(IS)=T(L)
C
DO 6 J=1,NP1
DO 6 IS=1,N21
L=L+1
6 H(IS,J)=T(L)
C
10 FORMAT(F20.6,5E20.8/(5E20.8)
RETURN
END
SIBFTC LINEAR LIST
SUBROUTINE LINEAR
COMMON T(2511),NT,RT(9),WT(9),UOBST(9),FORCE(9),UOBSTX(9),FORCT(9)
1 *NPRNT,MPRNT,NP1,NTMAX,KMAX,LFLAG,N21,NEQ
2 *DELTA,TENSN,A,UPREV(18),UOBSTX(9)
3 *TT(9),U(16,401),P(19),H(19,10),C(10)
4 *MX,FORCTX(9),ATRUE,CSPEED
DIMENSION AM(50,50),BV(50),PIVOT(50),PIVOT(50),INDEX(50,2)
C
DO 2 I=1,NT
II=NT+I
DO 1 J=1,NP1
1 AM(I,J)=H(II,J)
2 BV(I)=FORCT(I)/TENSN - P(I)
C
SUBROUTINE NEXT
COMMON T(2511),NT,RT(9),WT(9),UOBS1(9),FORCE(9),UOBST(9),FORCT(9),
1 +NPRNT,MPRNT,NP1,NT,O,KMAX,LFLAG,N21,NEQ,
2 +DELTA,TE,SMAX,UPREV(18),UOBSX(9),
3 +TT(9),U(15,401),P(19),H(19,10),C(10),
4 +MX,FORCTX(9),ATRUE,C,CSPEED

C
LFLAG=3
NEG=N21
DO 1 I=1,2511
1 T(I)=0.0
   T(3)=DELTA
   L=NT+3
   DO 2 I=1,NP1
       J=NT+1
       U(J+1)=C(I)
       L=L+1
2   T(L)=C(I)
   I=1
   CALL INTS(T,NEQ,2,C,0,0,0,0,0)
   MX=1
   CALL OUTPUT
C
   DO 4 MI=1,MPRNT
   DO 3 M2=1,MPRNT
   CALL INTM
       I=I+1
       L=3
   DO 3 IS=1,NTWO
       L=L+1
3   U(IS,I)=T(L)
   MX=I
END
SUBROUTINE OUTPUT
COMMON T(2511), NT, RT(9), WT(9), UOBST(9), FORCE(9), UOBS1(9), FORCT(9)
1 ,NPRNT, MPRNT, NP1, NTWO, KMAX, LFLAG, N21, NEG
2 ,DELTA, TENSNA, UPREV(18), UOBSTX(9)
3 ,TT(9), U18(401), P(19), H(19*10), C(1C)
4 ,MX, FORCTX(9), ATURE, CSPEED

C
I=MX
PRINT10, T(2)
10 FORMAT(1H04X 3HX =, F10.6)
PRINT11, (U(IS+1), IS=1,NT)
PRINT11, (U(IS+1), IS=NP1,NTWO)
11 FORMAT(1H04X6E20.8/(5X6E20.8))
RETURN
END
A partial program is listed:

LAPLAC subroutine

The following subroutines are required from Program F.1:

MAIN program
DAUX subroutine
INITL subroutine
PANDH subroutine
LINEAR subroutine
NEXT subroutine
OUTPUT subroutine

The following library routines are required:

MATINV
INTS/INTM
SUBROUTINE LAPLAC

COMMON T(2511), NT, RT(9), WT(9), UOBST(9), FORCE(9), UOBST(9), FORCT(9),
      TP, PRNT, NPI, NP2, KMAX, LFLAG, N21, NEG

FOF DELTA FUNCTION FORCE

THE TIMES

DO 1 I = 1, NT
1      TT(I) = ALOG(RT(I))

EXACT TRANSFORMS OF OBSERVATIONS

CSPEED = SORT(ATRUE)
COVERT = CSPEED / TENSN
DO 6 IS = 1, NT
   S = IS
   FORCTX(IS) = 1.0
6      UOBSTX(IS) = TANH(S / CSPEED) * FORCTX(IS) * COVERT / S
      PRINT98, (UOBSTX(IS), IS = 1, NT)
      PRINT99, (FORCTX(IS), IS = 1, NT)
      FORMAT(1H10.0F8.2HEXACT TRANSFORMS OF OBSERVATIONS / (5X6E20.8))
59      FORMAT(1H10.0F8.2HEXACT TRANSFORMS OF FORCE / (5X6E20.8))

THE TRANSFORMS

DO 2 IS = 1, NT
   UOBST(IS) = 0.0
   JJ = IS - 1
2      DO 1 I = 1, NT
      RW = WT(I) * (RT(I) ** JJ)
      1      UOBST(IS) = UOBST(IS) + UOBST(I) * RW

PRINT 10
10     FORMAT(1H10.0F8.2HFOF LTRANS I = I, T, 16X 4HF(I),
                14X 1MS, 6X14HFOBTRANS(I+S), 16X 4HF(S) /)
      DO 3 I = 1, NT
3      PRINT11, TT(I), UOBST(I), FORCE(I), I, UOBST(I), FORCT(I)
11     FORMAT(5XF10.6, 2E20.8, 10X, 15, 2E20.8)
RETURN
END
PROGRAM F.3. PRODUCTION OF OBSERVATIONS FOR EXAMPLE 3 — INHOMOGENEOUS MEDIUM WITH DELTA FUNCTION INPUT

The complete program is listed:

MAIN program
DAUX subroutine

The following library routines are required:
MATINV
INTS/INTM
PROGRAM #18FTC

DIMENSION C(5-ISO.0),P(50),P(50),INDEX(50,2)
COMMON T(1945),H(18.9),N2,NT,MRNT,MRNT,DELTA,A,B,U(18)

INPUT

1 READ100,NT,MRNT,MRNT
PRINT90,NT,MRNT,MRNT
READ11,DELTA,A,B
PRINT91,DELTA,A,B
?=2*NT
NEG=N2*NT

INITIALIZE

DO 2 I=1.1945

2 T(I)=.0
T(3)=DELTA
DO 3 J=1,NT
DO 4 I=1,N2

4 H(I,J)=0.0
K=NT+J

3 H(K,J)=1.0
L=3
DO 5 J=1,NT
DO 5 I=1.N2
L=L+1

5 T(L)=n(I,J)

CALL INTS(T,NEQ.2.C.0.C.0.0)
N=3=NEW+3
PRINT92,T(2),(T(I),I=4,N3)

INTEGRATE

DO 10 M1=1,MRNT
DC 9 M2=1,MRNT
CALL INTM

10 PRINT93,T(2),(T(I),I=4,N3)

LINEAR SYSTEM

L=3
DC 11 J=1,NT
DO 11 I=1,N2
L=L+1

11 H(I,J)=T(L)
DO 12 I=1,NT
II=NT+I
DO 12 J=1,NT
12 C(I,J)=H(I1,J)
DO 13 I=1,NT
13 D(I)=1.0
DO 15 I=1,NT
15 PRINT97,(C(I,J),J=1,NT)
CALL MATINV(C, NT, D, 1, DETERM, PIVOT, INDEX, IPIVOT)
PRINT94*(I, D(I) :== NT)

OBSERVATIONS UOBST(1, 5)
DO 14 I = 1, NT
  U(I) = 0
DO 14 J = 1, NT
  U(I) = U(I) + D(J)*H(I, J)
PRINT95*(I, U(I), I = 1, NT)
PUNCH96*(U(I), I = 1, NT)
GO TO 1

100 FORMAT(6I12)
101 FORMAT(6E12.8)
90 FORMAT(1H14X, 6120)
91 FORMAT(1H4X*6E20.8)
92 FORMAT(/5X3H4X = F7.3/(2X7E16.8))
93 FORMAT(1H4X3H4X = F7.3/(2X7E16.8))
94 FORMAT(/19X1H4X, 22X8HSGDF(1)/(15X15, E30.8))
95 FORMAT(/19X1H4X, 22X8HUS0ST(1)/(15X15, E30.8))
96 FORMAT(6E12.8)
97 FORMAT(2X7E16.8)
END

SUBTFC DAUX REF
SUBROUTINE DAUX
COMMON T(1945), H(18, 9), N2, NT, *PRT, *PRT, DELTA, A, B, U(16)

L = 3
DO 1 J = 1, NT
  DC 1 I = 1, N2
  L = L + 1
1  H(I, J) = T(L)
  DENOM = A + B*T(2)
DO 3 J = 1, NT
  DO 2 I = 1, NT
    L = L + 1
    I = I + 1
 2  T(L) = H(I, J)
  DO 3 I = 1, NT
    L = L + 1
    F = I**2
3  T(L) = T(L)*F
RETURN
END
PROGRAM F.4. DETERMINATION OF WAVE VELOCITY FOR EXAMPLE 3 — INHOMOGENEOUS MEDIUM WITH DELTA FUNCTION INPUT

The complete program is listed:

- MAIN program
- INITL subroutine
- PANDH subroutine
- LINEAR subroutine
- OUTPUT subroutine
- DAUX subroutine
- NEXT subroutine

The following library routines are required:

- MATINV
- INTS/INTM
VIBRATING STRING - LAPLACE TRANSFORMS - \( C^2 = A + B \times X \)

```fortran
COMMON T(251), NT, RT(9), MT(9), UOBST(9), FORCE(9), UOBST(9), FORCT(9)  
1  *NPRNT, MPRNT, NP1, NTX0, KMAX, LFLAG, N22, NP2, B, TRUE, NEQ  
2  *DELTA, TENSNA, UPREV(18), UOBSTX(9)  
3  *MX, FORCTX(9), ATRUE, CSPEED  

INPUT

1 READ1,0, NT, NPRNT, MPRNT, KMAX  
PRINT90, NT, NPRNT, MPRNT, KMAX  
READ1, 1, (RT(I), I = 1, NT)  
PRINT91, 1, (RT(I), I = 1, NT)  
READ1, 1, (MT(I), I = 1, NT)  
PRINT92, 1, (MT(I), I = 1, NT)  
READ1, 1, DELTA, TENSNA, ATRUE, B, TRUE  
PRINT93, 1, DELTA, TENSNA, ATRUE, B, TRUE  
READ1, 1, (UOBS(T(I), I = 1, NT)  
PRINT94, 1, (UOBS(T(I), I = 1, NT)  
READ1, 1, (FORC(T(I), I = 1, NT)  
PRINT95, 1, (FORC(T(I), I = 1, NT)  
NTWO = 2 * NT  
N22 = NTWO + 2  
NP1 = NT + 1  
NP2 = NT + 2  
DO 11 I = 1, NT  
11 U(I, 1) = 0, C  
READ1, 1, (U(I, 1), I = NP1, NT)  
PRINT96, 1, (U(I, 1), I = NP1, NT)  

GENERATE INITIAL APPROXIMATION

3 CALL INITL

SUCCESSIVE APPROXIMATIONS

4 DO 5 K = 1, KMAX  
PRINT97, K  
CALL PANDH  
CALL LINEAR  
5 CALL NEXT  
GO TO 1
```

```
100 FORMAT(1H14X, 4HNT =, 13/5X, 7HNPRTN =, 13/5X, 7HPRNT =, 13/5X, 5HMAX =, 13/5X)  
90 FORMAT(1H14X, 4HMAX =, 13/5X, 7HTEN =, 13/5X, 7HTRUE =, 13/5X)  
91 FORMAT(1H14X, 5HROOTS/(5X6E24.6))  
92 FORMAT(1H14X, 7HWEIGHTS/(5X6E24.6))  
93 FORMAT(1H14X, 7HDELTA =, E16.6, 5X, 7HTEN =, E16.6, 5X)  
1  5X20HINITIAL GUESS OF A =, E16.6, 5X8HTRUE A =, E16.6/  
2  5X20HINITIAL GUESS OF B =, E16.6, 5X8HTRUE B =, E16.6/  
94 FORMAT(1H14X, 12HOBOSERVATIONS,(5X6E24.6))  
95 FORMAT(1H14X, 12HFORCE F(T), (5X6E24.6))  
96 FORMAT(1H14X, 12HINITIAL GUESS OF U-PRIMED,(5X6E24.6))  
97 FORMAT(1H14X, 13HAPPROXIMATION, 13/)```
SUBROUTINE INITL
COMMONT(2511),NT,RT(9),AT(9),UOBS1(9),FORCE(9),UOBS(9),FORCT(9)
1*NPRT,MPRT,NT0,NTW0,KMAX,LFLAG,N22,NP2,BTRUE,NEQ
2*DELTA,TENSNA,UPREV(18),UOESTX(9)
3*TT(9),U(16,401),P(19),H(19,10),C(10)
4*MX,FORCTX(9),ATRUE,CSPED

INITIAL APPROXIMATION FROM NONLINEAR EQUATIONS

LFLAG=1
DO 1 I=1,2511
1 T(I)=0.0
T(3)=DELTA
L=NT+3
DO 2 IS=1,NT
J=NT+IS
L=L+1
2 T(L)=U(J,1)
L=L+1
T(L)=A
L=L+1
T(L)=B

I=1
N22=2*NT+2
NEQ=N22
CALL INTS(T,NEQ,2,0,0,0,0,0,0,0)
MX=1
CALL OUTPUT

DO 4 M1=1,NPRNT
DO 3 M2=1,NPRT
CALL INTM
I=I+1
L=3
DO 3 IS=1,NTW0
L=L+1
3 U(IS,I)=T(L)
MX=1
4 CALL OUTPUT
RETURN

SUBROUTINE PANDH
COMMONT(2511),NT,RT(9),AT(9),UOBS1(9),FORCE(9),UOBS(9),FORCT(9)
1*NPRT,MPRT,NT0,NTW0,KMAX,LFLAG,N22,NP2,BTRUE,NEQ
2*DELTA,TENSNA,UPREV(18),UOESTX(9)
3*TT(9),U(16,401),P(19),H(19,10),C(10)
4*MX,FORCTX(9),ATRUE,CSPED

LFLAG=2
DO 1 I=1,2511
1 T(I)=0.0
T(3)=DELTA
NEQ=(NT+3)*N22
L=4+N22+NT
T(L)=1.0
DO 2 I=1,NP1
2
L = L + \sqrt{22} + 1

T(L) = 1.0

I = 1

DO 12 IS = 1, NT

UPREV(IS) = U(IS, 1)

CALL INT3(T, NEQ, 2, 0, 0, 0, 0, 0)

NEQ3 = NEQ + 3

DO 4 M1 = 1, NPRNT

DO 3 M2 = 1, NPRNT

CALL INTM

I = I + 1

DO 3 IS = 1, NT

UPREV(IS) = U(IS, 1)

CONTINUE

L = 3

DO 5 IS = 1, N22

L = L + 1

5 P(IS) = T(L)

C

DO 6 J = 1, NP2

DO 6 IS = 1, N22

L = L + 1

6 H(IS, J) = T(L)

C

RETURN

END

**IBFTC**

**LINEAR LIST**

**SUBROUTINE LINEAR**

COMMON T(2511), NT, RT(9, 9), NT(9, 9), UOBST(9), FORCET(9), UOBST(9), FORCET(9)

DIMENSION AM(50, 50), BV(50), PIVOT(50), INDEX(50, 2)

C

DO 2 I = 1, NT

1 = NT + 1

DO 1 J = 1, NP2

1 AM(I, J) = H(I, J)

2 BV(I) = FORCET(I)/TENSN - P(I)

C

DO 4 I = 1, 2

1 = NT + 1

DO 3 J = 1, NP2

AM(I, J) = 0.0

DO 3 IS = 1, NT

AM(I, J) = AM(I, J) + H(IS, J) + H(IS, J)

BV(I) = 0.0

DO 4 IS = 1, NT

4 BV(I) = BV(I) + (UOBST(IS) - P(IS))*H(IS, I)

C

DO 5 I = 1, NP2

5 PRINT9*(AM(I, J), J = 1, NP2), BV(I)

9 FORMAT(1X, 1P7E18.8)

C

CALL MATINV(AM, NP2, BV, DETERM, PIVOT, INDEX, PIVOT)

C

DO 6 I = 1, NP2
6 \text{C(I)} = b \text{V(I)} \\
A = \text{C(NP1)} \\
b = \text{C(NP2)} \\
\text{PRINT9, (C(I), I=1, NP2)} \\
\text{RETURN} \\
\text{END}

$IBFTC$ \text{OUTPUT LIST} \\
\text{SUBROUTINE OUTPUT} \\
\text{COMMON T(2511), NT, RT(9), UT(9), UOBST(9), FORCE(9), UOBST(9), FORCT(9)} \\
1 \text{NPKNT, NPKRT, NP1, NTWO, KMIX, LFLAG, N22, NP2, B, BTRUE, NEQ} \\
2 \text{DELTA, TENSN, A, UPREV(18), UOBSTX(9)} \\
3 \text{T(9), U(18, 401), P(19), H(19), C(10)} \\
4 \text{MX, FORCTX(9), ATRUE, CSPEED}$

\begin{verbatim}
C C NONLINEAR C
100 L = 3 \\
DO 1 IS = 1, N22 \\
L = L + 1 \\
1 V(IS) = T(L) \\
L = NEQ + 3 \\
DO 2 IS = 1, NT \\
L = L + 1 \\
NN = NT + IS \\
2 T(L) = V(NN) \\
DO 3 IS = 1, NT \\
L = L + 1 \\
S = IS * IS \\
DENOM = V(N21) + V(N22) * T(2) \\
3 T(L) = S * V(IS) / DENOM \\
L = L + 1 \\
T(L) = U * 0 \\
L = L + 1 \\
T(L) = U * 0 \\
RETURN
C C LINEAR C
200 L = 3 \\
DO 4 IS = 1, N22 \\
L = L + 1
\end{verbatim}
4 \( V(IS) = T(L) \)
   \( M = NEQ + 3 \)
   DO 5 IS=1,NT
   \( M = M + 1 \)
   NN=NT+IS
5 \( T(M) = V(NN) \)
   DO 6 IS=1,NT
   \( M = M + 1 \)
   \( S = IS ** 2 \)
   DENOM=A+B*T(2)
6 \( T(M) = S \times (V(IS)/DENOM - (V(N21) + V(N22) \times T(2)) \times UPREV(IS)/ \)
   \( 1 \times DENOM ** 2 + UPREV(IS)/DENOM) \)
   \( M = M + 1 \)
   T(M)=U*C
   \( M = M + 1 \)
   T(M)=0.0 
C IF(LFLAG-3) 20,300,300
300 RETURN
C C C HOMOGENEOUS
C 20 DO10 J=1,NP2
C DO 7 IS=1,N22
   L=L+1
7 \( V(IS) = T(L) \)
C DO 8 IS=1,NT
   \( M = M + 1 \)
   NN=NT+IS
8 \( T(M) = V(NN) \)
C DO 9 IS=1,NT
   \( M = M + 1 \)
   \( S = IS ** 2 \)
   DENOM=A+B*T(2)
9 \( T(M) = (V(IS)/DENOM - (V(N21) + V(N22) \times T(2)) \times UPREV(IS)/ \)
   \( 1 \times DENOM ** 2 + UPREV(IS)/DENOM) \)
   \( M = M + 1 \)
   T(M)=U*C
   \( M = M + 1 \)
10 T(M)=0.0 
RETURN
END
$IBFTC NEXT REF
SUBROUTINE NEXT
COMMON T(2511),NT,RT(9),WT(9),UOBST(9),FORCE(9),UOBST(9),FORCT(9)
1 ,NPRNT,MPRNT,NP1,NP2,KMAX,LFLAG,N22,NP2,B,BTRUE,NEQ
2 ,DELTA,TENSNA,UPREV(18),UOBSTX(9)
3 ,TT(9),U(18,401),P(19),H(19,10),C(10)
4 ,MX,FORCTX(9),ATRUE,CSPEED
C
LFLAG=3
NEG=N22
DO 1 I=1,2511
1 T(1)=U*0
T(3)=DELTA
   L=NT+3
DO 2 I=1,NP2
J = N \cdot T + 1
U(J+1) = C(I)
L = L + 1

2 T(L) = C(I)
I = 1
CALL INTS(T*NEQ,2,0,0,0,0,0)
MX = 1
CALL OUTPUT

DO 4 M1 = 1, MPRNT
DO 3 M2 = 1, NPRNT
CALL INTM
I = I + 1
L = 3
DO 3 IS = 1, NTWO
L = L + 1

3 U(IS+1) = T(L)
MX = 1

4 CALL OUTPUT
RETURN
END
APPENDIX G

PROGRAMS FOR WAVE PROPAGATION:
MEASUREMENTS OF STEADY STATES
PROGRAM G.1. PRODUCTION OF REFLECTION COEFFICIENTS

The complete program is listed:

MAIN program
DAUX subroutine

The following library routine is required:
INTS/INTM
$JOB  2609, INDEX, K0160, 3, 100, 100, C
$IBJOB INDEX  MAP
$IBFTC  MAIN  LIST
C PHASE I
COMMON T(399)*N,NPRNT, MPRNT, DELTA, ZMAX, A, B, F(I), W(I), R(I), S(I),
1 ETA*
C
C INTEGRATE BACKWARDS FROM ONE TO ZERO
C
1 READ1*C, N, NPRNT, MPRNT
PRINT90*C, N, NPRNT, MPRNT
READ1*1, DELTA, ZMAX, A, B
ETA=A
PRINT91*1, DELTA, ZMAX, A, B, ETA
READ1*1, F(I), I=1, N
PRINT92*1, F(I), I=1, N
TPI=2.0*3.1415927
DO 2 I=1, N
2 T(I)=TPI*F(I)
PRINT93*1, T(I), I=1, N
C INTEGRATE
C
DO 3 I=1, 399
3 T(I)=0.0
T(2)=1.0
T(3)=DELTA
RR=0.0
DO 4 I=1, N
R(I)=RR
4 S(I)=0.0
NEQ=2*N + 1
J=NEQ + 3
L=3
DO 50 I=1, N
L=L+1
50 T(L)=R(I)
DO 41 I=1, N
L=L+1
41 T(L)=S(I)
T(J)=ETA
C
CALL INTS(T, NEQ, 2, C, 0, C, 0, 0)
CALL OUTPUT
C
DO 8 M1=1, MPRNT
DO 5 M2=1, NPRNT
5 CALL INTM
L=3
DO 6 I=1, N
L=L+1
6 R(I)=T(L)
DO 7 I=1, N
DIMENSION RV(5), SV(5)

L = 3
DO 1 I = 1, N
    L = L + 1
    RV(I) = T(L)
1    DO 2 I = 1, N
        L = L + 1
        SV(I) = T(L)
2    L = L * 1
    ETA = T(L)

ETA_P = 2.0 * ETA * (T(2) - 1.0)
ETA2 = ETA_P / ETA

DO 3 I = 1, N
    L = L + 1
    T(L) = 0.5 * ETA2 + 2.0 * ETA * W(I) * SV(I) - 0.5 * ETA2 * RV(I) + 2.0 * ETA2 * (RV(I) ** 2 - SV(I) ** 2)
3    DO 4 I = 1, N
        L = L + 1
        T(L) = -2.0 * ETA * W(I) * RV(I) - ETA2 * RV(I) * SV(I)
4    L = L + 1
    T(L) = ETA_P
5    RETURN
END

SUBROUTINE OUTPUT
PRINT 10, T(2), ETA
10 FORMAT(1X, F10.6, 5X, 7HINDEX =, E16.8)
PRINT 11
}
DO 1 I=1,N
AMP=SQRT(R(I)**2 + S(I)**2)
PHI=ATAN2(S(I),R(I))
1 PRINT12,F(I),W(I),R(I),S(I),AMP,PHI
C
11 FORMAT(1H013X11HFREQUENCY F, 7X13HANGULAR FREQ.,
1 11X9REAL PART, 11X9IMAGINARY, 11X9AMPLITUDE,15X5PHASE)
12 FORMAT(5X6E20.8)
RETURN
END
ENTRY
$ENTRY MAIN
3 10 100
-0.001 1.0 1.0 0.5
1.0 2.0 3.0
$IBSYS ENDJOB
PROGRAM G.2. DETERMINATION OF INDEX OF REFRACTION

The complete program is listed:

MAIN program
DAUX subroutine
OUTPUT subroutine

The following library routine is required:

INTS/INTM
COMMON T(435), N, NPRNT, MPRNT, KMAX, DELTA, ZMAX, A, B, ETA, F(5), W(5),
1 NTWO, N2P2, OBS(10), LFLAG, N2P1, X(7, 401), P(6, 401), H(6, 2, 401),
2 BVEC(2), AMAT(2, 2), R(5), S(5), MX, NEQ

1 READ100, N, NPRNT, MPRNT, KMAX
PRINT90, N, NPRNT, MPRNT, KMAX
READ101, DELTA, ZMAX, A, B, ETA
PRINT91, DELTA, ZMAX, A, B, ETA
READ101, (F(I), I = 1, N)
PRINT92, (F(I), I = 1, N)
TP1 = 2 * 0.3 * 1415927
DO 2 I = 1, N
2 W(I) = TPI * F(I)
PRINT93, (W(I), I = 1, N)
NTWO = 2 * N
N2P2 = NTWO + 2

OBSERVATIONS
READ101, (OBS(I), I = 1, NTWO)
PRINT94, (OBS(I), I = 1, NTWO)

INITIAL APPROXIMATION

K1 = 0
PRINT95, K1
PRINT97, A, B
N2P1 = NTWO + 1
LFLAG = 1
DO 2 I = 1, 435
3 T(I) = 0.0
T(2) = 1.0
T(3) = DELTA
DO 4 I = 1, NTWO
4 X(I, 1) = 0.0
X(N2P1, 1) = ETA
MX = 1
NEO = N2P1
J = NEO + 3
T(J) = ETA

CALL INTS(T, NEO, 2, 0, 0, 0, 0, 0, 0)

DO 8 M1 = 1, MPRNT
DO 5 M2 = 1, NPRNT
CALL INTM
MX = MX + 1
M = MX
L = 3
DO 6 I = 1, NTWO
C
C
C
SUCCESSIVE APPROXIMATIONS
C
DO 25 K1=1,KMAX
PRINT95,K1
C
PARTICULAR AND HOMOGENEOUS SOLUTIONS
C
LFLAG=2
M=1
MX=M
DO 9 I=1,NTWO
P(I+1)=0.0
DO 9 J=1,2
9 H(I+J+1)=0.0
C
NEO=3*N2P2
DO 10 I=1,435
10 T(I)=0.0
T(2)=1.0
T(3)=DELTA
L=3
DO 11 I=1,N2P2
L=L+1
11 T(L)=0.0
DO 12 J=1,2
DO 12 I=1,N2P2
L=L+1
12 T(L)=0.0
I=3+N2P2+N2P1
T(I)=1.0
J=3+NEQ
T(J)=1.0
C
DO 13 I=1,N
R(I)=X(I+1)
J=I+N
13 S(I)=X(J+1)
ETA=X(N2P1+1)
C
CALL INTS(T,NEO+2,0,0,0,0,0,0)
L3=NEQ+3
C
DO 18 M1=1,MPRNT
DO 17 M2=1,NPRNT
CALL INTM
M=M+1
BOUNDARY CONDITIONS DETERMINE NEW A, B

DO 20 I=1,2
BVEC(I)=0.0
DO 19 K=1,NTWO
19 BVEC(I)=BVEC(I) + T(K,1,M)*(BCBS(K)-P(K,M))
DO 20 J=1,2
AMAT(I,J)=0.0
DO 20 K=1,NTWO
20 AMAT(I,J)=AMAT(I,J) + T(K,1,M)*H(K,J,M)

D=AMAT(1,1)*AMAT(2,2) - AMAT(1,2)*AMAT(2,1)
A=(BVEC(1)*AMAT(2,2) - BVEC(2)*AMAT(1,2))/D
B=(BVEC(2)*AMAT(1,1) - BVEC(1)*AMAT(2,1))/D
PRINT97,A,B

NEW APPROXIMATION

M=1
MX=M
T(2)=1.0
X(N2P1,M)=A

DO 22 M1=1,MPRNT
DO 21 M2=1,NPRNT
M=M+1
MX=M
T(2)=T(2)+DELTA
X(N2P1,M)=A + B*(T(2)-1.0)**2
ETA=X(N2P1,M)
DO 21 I=1,NTWO
21 X(I,M)=P(I,M) + A*H(I,1,M) + B*H(I,2,M)
22 CALL OUTPUT
25 CONTINUE
GO TO 1

100 FORMAT(6I12)
SUBROUTINE DAUX

DIMENSION RV(5), SV(5), RPREV(5), SPRV(5), FUNR(5), FUNS(5)
COMMON T(435), N, NPRNT, XPRNT, KMAX, DELTA, ZMAX, A, B, ETA, F(5), W(5),
1. NTWO, N2P2, DOES(10), LFLAG, N2P1, X(7, 401), P(6, 401), H(6, 2, 401),
2. BVEC(2), AMAT(2, 2), R(5), S(5), MX, NEQ

C
GO TO (10, 20) * LFLAG

C NONLINEAR

10    L = 3
DO 1 I = 1, N
   L = L + 1
  1 RV(I) = T(L)
DO 2 I = 1, N
   L = L + 1
  2 SV(I) = T(L)
   L = L + 1
ETA = T(L)
ETAPRI = 2.0 * B * (T(2) - 1.0)
PR = ETAPRI / ETA

C
DO 3 I = 1, N
   L = L + 1
  3 T(L) = 0.5 * PR + 2.0 * W(I) * SV(I) * ETA - PR * (RV(I) ** 2 - SV(I) ** 2) * 0.5
DO 4 I = 1, N
   L = L + 1
  4 T(L) = -2.0 * ETA * W(I) * RV(I) - PR * RV(I) * SV(I)
   L = L + 1
T(L) = ETAPRI
RETURN

C LINEAR

20 ETA = A + B * (T(2) - 1.0) ** 2
ETAPR = 2.0 * B * (T(2) - 1.0)
PR = ETAPR / ETA
DNDA = 1.0
DNDB = (T(2) - 1.0) ** 2
DNPNDA = -ETAPR / ETA ** 2
DNPNDB = 2.0 * (T(2) - 1.0) / ETA - (ETAPR * (T(2) - 1.0) ** 2) / ETA ** 2

C CPARTICULAR

L = 3
DO 5 I=1,N
L=L+1
5 RV(I)=T(L)
DO 6 I=1,N
L=L+1
6 SV(I)=T(L)
L=L+1
ANEW=T(L)
BNEW=T(L)
DO 7 I=1,N
RPREV(I)=R(I)
SPREV(I)=S(I)
MPREV=A
BPREV=B
DO 8 I=1,N
FUNR(I)=0.5*PR + 2.0*ETA*W(I)*S(I) - (R(I)**2-S(I)**2)*PR*0.5
FUNS(I)=-2.0*ETA*W(I)*R(I) - R(I)*S(I)*PR
IFLAG=0
M=NEQ+3
C
100 IFLAG=IFLAG+1
DO 101 I=1,N
M=M+1
T(M)=FUNR(I) + (RV(I)-RPREV(I))*(R(I)*PR)
T(M)=T(M) + (SV(I)-SPREV(I))*(2.0*ETA*W(I) + S(I)*PR)
T(M)=T(M) + (ANEW-APREV)*(0.5*DNPDCA+2.0*W(I)*S(I)*DNDA)
   - 0.5*(R(I)**2-S(I)**2)*DNPDCA
101 T(M)=T(M) + (BNEW-BPREV)*(0.5*DNPCB+2.0*W(I)*S(I)*DNDDB)
   - 0.5*(R(I)**2-S(I)**2)*DNPCDB
DO 102 I=1,N
M=M+1
T(M)=FUNS(I) + (RV(I)-RPREV(I))*(-2.0*ETA*W(I)-S(I)*PR)
T(M)=T(M) + (SV(I)-SPREV(I))*(-R(I)*PR)
T(M)=T(M) + (ANEW-APREV)*(-2.0*W(I)*R(I)*DNDCA-R(I)*S(I)*DNPDCA)
102 T(M)=T(M) + (BNEW-BPREV)*(-2.0*W(I)*R(I)*DNDDB-R(I)*S(I)*DNPDDB)
   - 0.5*(R(I)**2-S(I)**2)*DNPCDB
C
IF(IFLAG-1)50,201
C
CHOMOGENEOUS
50 DO 201 J=1,2
DO 51 I=1,N
L=L+1
51 RV(I)=T(L)
DO 52 I=1,N
L=L+1
52 SV(I)=T(L)
L=L+1
ANEW=T(L)
BNEW=T(L)
DO 53 I=1,N
RPREV(I)=0.0
SPREV(I)=0.0
FUNR(I)=0.0
53 FUNS(I)=0.0
APREV=0.0
BPREV=0.0
GO TO 100
201 CONTINUE
RETURN
END

SUBROUTINE OUTPUT REF
COMMON T(435), N, NPRINT, MPRINT, KM, MAX, DELTA, ZMAX, L, B, ETA, F(5), W(5),

1. NTP0, N2P2, G0BS(10), LFLAG, N2P1, X(7, 401), P(6, 4, 1), H(6, 2, 401),
2. BVEC(2), AMAT(2, 2), R(5), S(5), MX, NEQ

C
PRINT92
DO 4 I=1,N
R(I)=X(I, MX)
J=I+N
4 S(I)=X(J, MX)
IF(MX-1)11,1,2
1 PRINT91
2 DO 3 I=1,N
AMP=SQRT(R(I)**2+S(I)**2)
PHI=ATAN2(S(I), R(I))
3 PRINT93, T(2), X(N2P1+MX), I, R(I), S(I), AMP, PHI
C
95 FORMAT(1HC4X, 3HA = , E18.8
91 FORMAT(1HC1X1HX, 6X8HINDEX(X) + 5X1HI + 11X9HREAL PART,

1. 11X9HIMAGINARY + 11X9HAMPLITUDE, 15X5HPHASE //)
93 FORMAT(F12*4, F14*6, 16*4E20*8)
92 FORMAT(1HC)
RETURN
END

ENTRY MAIN

3 20 20 5
-0.025 1.0 1.0 0.4 1.0
1.0 2.0 3.0
+1.32178E-02+323131E-03+368550E-03+148430E-01+954147E-02+589762E-02

316
An investigation of inverse problems as basic problems in science, in which physical systems are to be identified on the basis of experimental observations. These problems are especially important in astrophysics and astronomy, for their objects of investigation are frequently not observable in a direct fashion. Solar and stellar structure, for example, is estimated from the study of spectra, while the structure of a planetary atmosphere may be deduced from measurements of reflected sunlight. This Memorandum shows that a wide class of inverse problems may now be solved with high-speed computers and modern computational techniques.