J. Geophys. 42, 191 – 200, 1976

Journal of Geophysics

Computation of Reflection Coefficients for Layered Media *

R. Kind

Geophysikalisches Institut, Universität Karlsruhe, Hertzstr. 16, Bau 42, D-7500 Karlsruhe 21, Federal Republic of Germany

Abstract. A fast computer program of the Thomson-Haskell matrix formalism is presented for the computation of the $P-SV$ reflection coefficients R_{pp}, R_{ps}, R_{ss} and R_{sp} for layered solid media. A matrix formalism and a computer program are also derived for the computation of P reflection coefficients for layered liquid media and of SH reflection coefficients for layered solid media.

Key words: Theoretical seismograms – Thomson-Haskell matrix formalism – Reflection coefficients.

Introduction

The reflectivity method of computing theoretical seismograms (Fuchs and Müller, 1971) is now a more often used tool for the interpretation of data in explosion seismology as well as in earthquake seismology. Although this method has great advantages, it suffers from rather long computer times. This is especially cumbersome if complete seismograms for the whole earth are computed (Müller and Kind, 1976). Therefore increasing the speed of the computations is still a desirable aim. The central part of the reflectivity method is the calculation of plane body waves in a layered medium. This problem is similar to the problem of computing dispersion of surface waves in such a medium. Efficient computer programs for the latter have been published by Schwab and Knopoff (1972). In the present paper a fast program is presented of the Thomson-Haskell matrix formalism for the computation of $P-SV$ reflection coefficients. A computer program for sound wave reflection coefficients for a layered liquid is also presented. Because of the equivalence of sound waves in a liquid and of SH waves in a solid, which was established by Satô (1954), this program can also be used for SH reflection coefficients.

* Contribution No. 136, Geophysical Institute, University of Karlsruhe

Computation of $P-SV$ Reflection Coefficients

We consider monochromatic plane waves propagating in a medium consisting of a number of parallel, solid, homogeneous, isotropic and ideal elastic layers between two halfspaces. A potential vector is defined (see e.g. Dunkin (1965)) for each of the n different media

$$\Phi_i=(\varphi_i^-,\psi_i^-,\varphi_i^+,\psi_i^+),\qquad i=1,n \tag{1}$$

where φ_i^+, φ_i^- and ψ_i^+, ψ_i^- are the P wave and SV wave potentials, respectively, corresponding to waves travelling in positive or negative z direction. The application of the boundary conditions yields a relation between the potential vectors of the lower and upper halfspace:

$$\Phi_n=M\Phi_1 \tag{2}$$

where M is the Haskell matrix. It is the product of the matrix of the lower half-space T_n, the $n-2$ layer matrices G_i, and the matrix of the upper halfspace T_1:

$$M=T_n\cdot G_{n-1}\ldots G_2\cdot T_1. \tag{3}$$

The elements of all these matrices are given by Fuchs (1968). Equation (2), however, cannot be used directly for numerical computations due to an intrinsic loss-of-precision problem. The delta matrix extension and the reduced delta matrix extension (Pestel and Leckie, 1963; Dunkin, 1965; Watson, 1970) were developed to overcome this problem. The 6×6 delta matrix of the 4×4 Haskell matrix is obtained by computing all possible 2×2 subdeterminants of the 4×4 matrix. The reduced delta matrix extension allows to work with 5×5 matrices instead of the original 6×6 matrices, due to symmetry in the elements.

Červený (1974) has calculated the reflection coefficients from (2) in terms of the elements $\hat{M}_{ij}$ of the delta matrix $\hat{M}$ of the Haskell matrix M:

$$\begin{aligned} R_{pp}&=\hat{M}_{14}/\hat{M}_{11}, & R_{ps}&=-\hat{M}_{12}/\hat{M}_{11}\\ R_{ss}&=-\hat{M}_{13}/\hat{M}_{11}, & R_{sp}&=\hat{M}_{15}/\hat{M}_{11}. \end{aligned} \tag{4}$$

There exists a very important multiplication rule in delta matrix theory: the delta matrix of a product matrix is equal to the product of the delta matrices of the individual factor matrices. Therefore the $\hat{M}_{ij}$ can be computed by multiplication of the delta matrices $\hat{T}_n$, $\hat{G}_i$ and $\hat{T}_1$ of T_n, G_i and T_1 (see Eq. (3)). This solves the loss-of-precision problem. Only the first row $\hat{M}_{1,i}\,(i=1,5)$ of $\hat{M}$ is needed to compute (4). To obtain this row, one has to perform a matrix multiplication of the symbolic form

$$(1\times 5)_n\cdot(5\times 5)_{n-1}\cdot\cdots\cdot(5\times 5)_2\cdot(6\times 5)_1 \tag{5}$$

where $(1\times 5)_n$ stands for the first row of the reduced delta matrix $\hat{T}_n$, the $(5\times 5)_i$ represent the reduced delta matrices $\hat{G}_i$, and $(6\times 5)_1$ represents the required elements of the delta matrix $\hat{T}_1$ (which is not reduced). The elements of the delta matrices have been given by Fuchs (1968) and Kind and Müller (1975). They will be given in the following, some in a rearranged form, more suitable for computers.

The reduced delta matrix extension (Watson, 1970) uses the equality of the following elements:

$$(\hat{T}_n)_{13}=(\hat{T}_n)_{14},\quad (\hat{G}_i)_{13}=(\hat{G}_i)_{14},\quad (\hat{G}_i)_{23}=(\hat{G}_i)_{14},\quad (\hat{G}_i)_{53}=(\hat{G}_i)_{54},$$
$$(\hat{G}_i)_{63}=(\hat{G}_i)_{64},\quad (\hat{G}_i)_{31}=(\hat{G}_i)_{41},\quad (\hat{G}_i)_{32}=(\hat{G}_i)_{42},\quad (\hat{G}_i)_{35}=(\hat{G}_i)_{45},$$
$$(\hat{G}_i)_{36}=(\hat{G}_i)_{46},\quad \text{and}\quad (\hat{G}_i)_{44}=(\hat{G}_i)_{34}=(\hat{G}_i)_{43}=(\hat{G}_i)_{33}-1.$$

From this follows that in the product of the first row of $\hat{T}_n$ and $\hat{G}_i$ the element $(\hat{T}_n)_{14}$ may be omitted if $(\hat{T}_n)_{13}$ is multiplied by 2 and if the 4th row and column of $\hat{G}_i$ is omitted and 0.5 is subtracted from $(\hat{G}_i)_{33}$. The element $(\hat{G}_i)_{33}$ is already replaced by $(\hat{G}_i)_{33}-0.5$ in (7). The 3rd element of the (1×5) matrix in (5) must be multiplied by 2 in each multiplication step. In the delta matrix $\hat{T}_1$ only the first, 3rd and 4th columns have equal elements in their 3rd and 4th row, which allows the application of the reduced delta matrix extension only for these columns.

We have in the i-th medium:

$\alpha_i = P$ velocity
$\beta_i = S$ velocity
$\rho_i =$ density
$d_i =$ layer thickness
(not defined in the two halfspaces)

$\omega =$ angular frequency
$c =$ horizontal phase velocity
$k=\omega/c$ wave number
$j =$ imaginary unit
$\mu_i=\beta_i^2\,\rho_i$
$l_i=2k^2-\omega^2/\beta_i^2$

$$v_i=\begin{cases}\sqrt{c^2/\alpha_i^2-1}, & c\geqq\alpha_i\\ -j\sqrt{1-c^2/\alpha_i^2}, & c<\alpha_i\end{cases}\qquad v_i'=\begin{cases}\sqrt{c^2/\beta_i^2-1} & c\geqq\beta_i\\ -j\sqrt{1-c^2/\beta_i^2}, & c<\beta_i.\end{cases}$$

The elements of $\hat{T}_n$ are:

$$(\hat{T}_n)_{11}=-\frac{\beta_n^4\,\rho_n}{2\omega^2}(4k^2\,v_n\,v_n'+l_n^2)$$

$$(\hat{T}_n)_{12}=j/2\,v_n$$

$$(\hat{T}_n)_{13}=-\frac{j\,\beta_n^2}{2\omega c}(l_n+2\,v_n\,v_n')$$

$$(\hat{T}_n)_{15}=-j/2\,v_n'$$

$$(\hat{T}_n)_{16}=-\frac{1}{2\rho_n\,\omega^2}(v_n\,v_n'+k^2)$$

For the elements of the layer delta matrix $\hat{G}_i$ the following abbreviations are introduced:

$$\begin{aligned}
&\gamma_i=-2\beta_i^2/c^2, && W_i=\sin P_i/v_i, && e_1=\cos P_i\cdot\cos Q_i, && r_1=c\,\omega\,\rho_i,\\
&P_i=k\,v_i\,d_i, && Y_i=\sin Q_i/v_i', && e_2=1-e_1, && r_2=1/r_1,\\
&Q_i=k\,v_i'\,d_i, && X_i=\sin P_i\,v_i, && e_3=W_i\,Y_i, && r_3=r_1\,\gamma_i,\\
&\gamma_2=\gamma_i+1, && Z_i=\sin Q_i\,v_i, && e_4=X_i\,Z_i, && r_4=r_1\,\gamma_2,\\
& && && e_5=W_i\cos Q_i, && f_1=e_2+e_3,\\
& && && e_6=Y_i\cos P_i, && f_2=f_1\,r_2.
\end{aligned}\tag{6}$$

Then, the elements of $\hat{G}_i$ are:

$$(\hat{G}_i)_{16} = -r_2(f_2 + (e_2 + e_4)\, r_2)$$
$$g_{13} = -r_3(\hat{G}_i)_{16} + f_2$$
$$(\hat{G}_i)_{13} = j\, g_{13} = (\hat{G}_i)_{36}$$
$$f_3 = \gamma_i f_1 + e_3$$
$$f_4 = r_3\, g_{13} + f_3$$
$$g_{31} = r_3\, f_4 + f_3\, r_4$$
$$(\hat{G}_i)_{31} = j\, g_{31} = (\hat{G}_i)_{63}$$
$$(\hat{G}_i)_{11} = e_1 - f_4 = (\hat{G}_i)_{66}$$
$$(\hat{G}_i)_{33} = f_4 + 0.5$$
$$(\hat{G}_i)_{61} = -r_3\, g_{31} - r_4(e_3\, r_4 + f_3\, r_3)$$
$$(\hat{G}_i)_{15} = -r_2(e_5 + Z_i \cos P_i) = (\hat{G}_i)_{26}$$
$$g_{23} = -r_3(\hat{G}_i)_{15} + e_5$$
$$(\hat{G}_i)_{23} = j\, g_{23} = (\hat{G}_i)_{35}$$
$$(\hat{G}_i)_{21} = -r_3\, g_{23} - r_4\, e_5 = (\hat{G}_i)_{65}$$
$$(\hat{G}_i)_{12} = r_2(e_6 + X_i \cos Q_i) = (\hat{G}_i)_{56}$$
$$g_{32} = -r_3(\hat{G}_i)_{12} - e_6$$
$$(\hat{G}_i)_{32} = j\, g_{32} = (\hat{G}_i)_{53}$$
$$(\hat{G}_i)_{51} = -r_3\, g_{32} + r_4\, e_6 = (\hat{G}_i)_{62}$$
$$(\hat{G}_i)_{22} = e_1 = (\hat{G}_i)_{55}$$
$$(\hat{G}_i)_{25} = Z_i\, W_i$$
$$(\hat{G}_i)_{52} = X_i\, Y_i. \qquad (7)$$

The required elements of $\hat{T}_1$ are:

$$(\hat{T}_1)_{11} = -k^2 - v_1\, v_1'$$
$$(\hat{T}_1)_{21} = -j\rho_1\, v_1'\, \omega^2$$
$$(\hat{T}_1)_{31} = -j\mu_1\, k(l_1 + 2 v_1\, v_1')$$
$$(\hat{T}_1)_{51} = j\rho_1\, v_1\, \omega^2$$
$$(\hat{T}_1)_{61} = -\mu_1^2(l_1^2 + 4k^2\, v_1'\, v_1)$$
$$(\hat{T}_1)_{12} = 2k\, v_1, \qquad (\hat{T}_1)_{15} = 2k\, v_1'$$
$$(\hat{T}_1)_{22} = (\hat{T}_1)_{52} = 0, \qquad (\hat{T}_1)_{25} = (\hat{T}_1)_{55} = 0$$
$$(\hat{T}_1)_{32} = j4\mu_1\, k^2\, v_1, \qquad (\hat{T}_1)_{35} = j2\mu_1\, l_1\, v_1'$$
$$(\hat{T}_1)_{42} = j2\mu_1\, l_1\, v_1, \qquad (\hat{T}_1)_{45} = j4\mu_1\, k^2\, v_1'$$
$$(\hat{T}_1)_{62} = 4\mu_1^2\, l_1\, k\, v_1, \qquad (\hat{T}_1)_{65} = 4\mu_1^2\, l_1\, k\, v_1'$$
$$(\hat{T}_1)_{14} = k^2 - v_1\, v_1' = -(\hat{T}_1)_{13}$$
$$(\hat{T}_1)_{24} = -(\hat{T}_1)_{21}, \qquad (\hat{T}_1)_{23} = j v_1'\, \mu_1(2k^2 - l_1)$$
$$(\hat{T}_1)_{34} = -jk\, \mu_1(l_1 - 2v_1\, v_1') = -(\hat{T}_1)_{33}$$
$$(\hat{T}_1)_{54} = (\hat{T}_1)_{51}, \qquad (\hat{T}_1)_{53} = j v_1\, \mu_1(2k^2 - l_1)$$
$$(\hat{T}_1)_{64} = \mu_1^2(l_1^2 - 4k^2\, v_1\, v_1') = -(\hat{T}_1)_{63}. \qquad (8)$$

The time consuming innermost loop in the computer program contains essentially the construction of the layer matrix $\hat{G}_i$ from (7) and the matrix multiplication (5). Setting up the elements of $\hat{G}_i$ according to (7) requires about three times less operations than in the version of Fuchs (1968). In general the matrix $\hat{T}_n$ is complex. The elements of $\hat{G}_i$ are either real or imaginary. In (5) we have to multiply a (1×5)

complex matrix with a (5×5) real or imaginary matrix, if we do the multiplication from the left to the right. This means 50 multiplications with each step, if the complex multiplication is separated into real and imaginary part. Fuchs (1968) multiplied the (6×6) matrices $\hat{G}_i$ first, which means 216 multiplications with each step. The so far probably fastest program for the layered media problem is due to Schwab and Knopoff (1972). They have in their innermost loop about half as many operations as in the comparable part of the present version. However, their program is real, which is sufficient for Rayleigh wave dispersion computations. For theoretical seismograms, however, the complex version is required. The FORTRAN program for the computation of $P-SV$ reflection coefficients is shown in Appendix 1. A normalization process is contained in the innermost loop of the program in order to avoid overflow problems (see Schwab and Knopoff (1972)). The normalization is not always required in every layer. In some cases a few percent of computer time may be saved by omitting the normalization.

Computation of Reflection Coefficients of Sound Waves in a Liquid and of *SH* Waves in a Solid

Satô (1954) has established the equivalence of *SH* waves and sound waves in a liquid. The reflection coefficients in both problems are identical if the following correspondence is used:

$$V_s \; (= S \text{ velocity in the solid}) \leftrightarrow V_p \; (= \text{velocity in the liquid})$$

and

$$V_s^2 \cdot \rho_s \; (\rho_s = \text{density in the solid}) \leftrightarrow 1/\rho_p \; (\rho_p = \text{density in the liquid}).$$

Therefore, after a density transformation, the same computer program can be used for both problems.

In the following a matrix formalism for a layered liquid medium will be derived, following lecture notes by Gerhard Müller. The potential in the i-th medium is

$$\Phi_i = \exp[j(\omega t - kx)] \cdot [A_i \exp(-jk v_i(z-z_i)) + B_i \exp(jk v_i(z-z_i))]$$

with the same denotations as in the previous section and the depth of the i-th boundary z_i.

At the boundaries $z = z_i$ we have

$$\frac{\partial \Phi_{i+1}}{\partial z} = \frac{\partial \Phi_i}{\partial z} \quad \text{and} \quad \rho_{i+1} \frac{\partial^2 \Phi_{i+1}}{\partial t^2} = \rho_i \frac{\partial^2 \Phi_i}{\partial t^2}.$$

From this follows

$$\begin{pmatrix} A_{i+1} \\ B_{i+1} \end{pmatrix} = m_i \begin{pmatrix} A_i \\ B_i \end{pmatrix}$$

where

$$m_i = \begin{pmatrix} l_i \rho_{i+1} + l_{i+1} \rho_i & (-l_i \rho_{i+1} + l_{i+1} \rho_i) \exp(2 j l_i d_i) \\ -l_i \rho_{i+1} + l_{i+1} \rho_i & (-l_i \rho_{i+1} + l_{i+1} \rho_i) \exp(2 j l_i d_i) \end{pmatrix} \tag{9}$$

and $d_i = z_{i+1} - z_i$, $d_1 = 0$ and $l_i = k v_i$. The exponential term containing x and t and a factor $\exp(-j l_i d_i)/(2 l_{i+1} \rho_{i+1})$ common to all elements of m_i, have been omitted. Repeated application of the same formalism yields

$$\begin{pmatrix} A_n \\ B_n \end{pmatrix} = m_{n-1} \cdot m_{n-2} \ldots m_2 \cdot m_1 \begin{pmatrix} A_1 \\ B_1 \end{pmatrix} = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} \begin{pmatrix} A_1 \\ B_1 \end{pmatrix}.$$

From this follows the reflection coefficient $(B_n = 0)$:

$$R_{pp} = \frac{B_1}{A_1} = -\frac{M_{21}}{M_{22}}. \tag{10}$$

We only need to perform a matrix multiplication of the symbolic form

$$(1 \times 2)_{n-1} \cdot (2 \times 2)_{n-2} \ldots (2 \times 2)_1 \tag{11}$$

in order to obtain M_{21} and M_{22}. This is similar to (5), but a difference is, that in (5) we have one matrix for each medium, whereas we have in (11) one matrix for each boundary. Computer time is saved, if the matrix multiplication is written in the following form

$$\begin{aligned} m_{21}^i &= e_1^i + e_2^i \\ m_{22}^i &= \exp(2 j l_i d_i)(e_1^i - e_2^i) \\ e_1^i &= e_1^{i+1}(m_{21}^{i+1} + m_{22}^{i+1}) \\ e_2^i &= e_2^{i+1}(m_{21}^{i+1} - m_{22}^{i+1}) \\ e_1^{i+1} &= l_{i+1} \rho_i \\ e_2^{i+1} &= l_i \rho_{i+1} \end{aligned} \tag{12}$$

for $i = n-1 \ldots 1$ and $m_{21}^n = 0$, $m_{22}^n = 1$. Successive application yields: $M_{21} = m_{21}^1$ $M_{22} = m_{22}^1$. A list of the corresponding FORTRAN program is shown in Appendix 2. It should be mentioned, that the two computer programs for a solid medium and for a liquid medium have identical output for R_{pp} for more than five digits, if 0.001 km/s is chosen for the shear velocity in the solid medium. This shows that a mixed model can be approximated with good accuracy if for the liquid layers a small shear velocity such as 0.001 km/s is taken.

Acknowledgements. This research was supported by a grant of the Deutsche Forschungsgemeinschaft. The development of the computer program was done at the computing center of the University of Karlsruhe. I wish to thank Gerhard Müller for discussions and for reading the manuscript and Karl Fuchs for reading the manuscript. The manuscript was typed by Ingrid Hörnchen.

Appendix 1. Computer program for the computation of $P-SV$ reflection coefficients for a layered solid. No provision is made in this program and in the program of Appendix 2 for zero frequency and for phase velocities exactly equal to layer velocities. These cases can easily be avoided.

```
      SUBROUTINE RECOPS(N,A,B,RHO,D,U,FREQ,RPP,RPS,RSS,RSP)
C
C     COMPUTATION OF P-SV REFLECTION COEFFICIENTS
C
C     N= NUMBER OF DIFFERENT MEDIA, STARTING ON TOP
C     A(I),B(I),RHO(I),(I=1,N)= P-VELOCITY, S-VELOCITY AND DENSITY
C     D(I),(I=2,N-1)= LAYER THICKNESS
C     U= PHASE SLOWNESS, FREQ= FREQUENCY
C     RPP,RPS,RSS,RSP= COMPLEX PP,PS,SS,SP-REFLECTION COEFFICIENTS
      DIMENSION A(N),B(N),RHO(N),D(N)
      COMPLEX T1,T2,T3,T4,T5,RPP,RPS,RSS,RSP,DET,CN,CNS,T53,T63
     A,T11,T21,T31,T51,T61,T12,T15,T32,T45,T42,T35,T62,T65,T13,T23,T33
      PI=3.14159265
      OMEG=2.*PI*FREQ
      C=1./U
      RK=OMEG*U
      N1=N-1
      COM=C*OMEG
      U2=U*U
      C2=C*C
      RK2=RK*RK
      OM2=OMEG*OMEG
C     SET MATRIX ELEMENTS OF EQUATION (6)
      S=B(N)
      P=A(N)
      RRO=RHO(N)
      S2=S*S
      P2=P*P
      ARGP=1.-C2/P2
      ARGS=1.-C2/S2
      IF(ARGP.GE.0.) CN=CMPLX(0.,-RK*SQRT(ARGP))
      IF(ARGP.LT.0.) CN=CMPLX(RK*SQRT(-ARGP),0.)
      IF(ARGS.LT.0.) CNS=CMPLX(RK*SQRT(-ARGS),0.)
      IF(ARGS.GE.0.) CNS=CMPLX(0.,-RK*SQRT(ARGS))
      RL=2.*RK2-OM2/S2
      RPP=CN*CNS
      T1=CMPLX(-S2*S2*RRO/(OM2*OM2),0.)*(CMPLX(4.*RK2,0.)*RPP+
     ACMPLX(RL*RL,0.))
      T2=CMPLX(0.,0.5)*CN
      T3=CMPLX(0.,-S2*U/(2.*OMEG))*(CMPLX(RL,0.)+RPP+RPP)
      T4=CMPLX(0.,-0.5)*CNS
      T5=CMPLX(-1./(2.*RRO*OM2),0.)*(RPP+CMPLX(RK2,0.))
      TR1=REAL(T1)
      TI1=AIMAG(T1)
      TR2=REAL(T2)
      TI2=AIMAG(T2)
      TR3=2.*REAL(T3)
      TI3=2.*AIMAG(T3)
      TR4=REAL(T4)
      TI4=AIMAG(T4)
      TR5=REAL(T5)
      TI5=AIMAG(T5)
      IF(N.LT.3) GOTO 2000
C     SET MATRIX ELEMENTS (7)
C     DO MATRIX MULTIPLICATION (5) FROM LEFT TO RIGHT
C     DO NORMALIZATION
      DO 1000 J=2,N1
      I=N-J+1
      S=B(I)
      S2=S*S
      P=A(I)
      P2=P*P
      THK=RK*D(I)
      ARGP=1.-C2/P2
      IF(ARGP.GE.0.) GOTO 190
      RA=SQRT(-ARGP)
      P=THK*RA
      SP=SIN(P)
      CP=COS(P)
      X=RA*SP
  180 ARGS=1.-C2/S2
      IF(ARGS.GE.0.) GOTO 200
      RB=SQRT(-ARGS)
      Q=THK*RB
      SQ=SIN(Q)
      CQ=COS(Q)
      Z=SQ*RB
      GOTO 210
  190 RA=-SQRT(ARGP)
      EP=0.5*EXP(THK*RA)
      EM=0.25/EP
      SP=EP-EM
```

```
      CP=EP+EM
      X=-SP*RA
      GOTO 180
200   RB=-SQRT(ARGS)
      EP=0.5*EXP(THK*RB)
      EM=0.25/EP
      SQ=EP-EM
      CQ=EP+EM
      Z=-SQ*RB
210   W=SP/RA
      Y=SQ/RB
      G1=-2.*S2*U2
      G2=G1+1.
      E1=CP*CQ
      E2=1.-E1
      E3=W*Y
      E4=X*Z
      E5=W*CQ
      E6=Y*CP
      R1=COM*RHO(I)
      R2=1./R1
      R3=R1*G1
      R4=R1*G2
      F1=E2+E3
      F2=F1*R2
      G16=-R2*(F2+(E2+E4)*R2)
      G13=-R3*G16+F2
      F3=G1*F1+E3
      F4=R3*G13+F3
      G31=R3*F4+F3*R4
      G11=E1-F4
      G33=F4+0.5
      G61=-R3*G31-R4*(E3*R4+F3*R3)
      G15=-R2*(E5+Z*CP)
      G23=-R3*G15+E5
      G21=-R3*G23-R4*E5
      G12=R2*(E6+X*CQ)
      G32=-R3*G12-E6
      G51=-R3*G32+R4*E6
      G22=E1
      G25=Z*W
      G52=X*Y
      TR11=TR1*G11+TR2*G21-TI3*G31+TR4*G51+TR5*G61
      TI11=TI1*G11+TI2*G21+TR3*G31+TI4*G51+TI5*G61
      TR22=TR1*G12+TR2*G22-TI3*G32+TR4*G52+TR5*G51
      TI22=TI1*G12+TI2*G22+TR3*G32+TI4*G52+TI5*G51
      TR33=-TI1*G13-TI2*G23+TR3*G33-TI4*G32-TI5*G31
      TI33=TR1*G13+TR2*G23+TI3*G33+TR4*G32+TR5*G31
      TR44=TR1*G15+TR2*G25-TI3*G23+TR4*G22+TR5*G21
      TI44=TI1*G15+TI2*G25+TR3*G23+TI4*G22+TI5*G21
      TR5=TR1*G16+TR2*G15-TI3*G13+TR4*G12+TR5*G11
      TI5=TI1*G16+TI2*G15+TR3*G13+TI4*G12+TI5*G11
      TR1=TR11
      TI1=TI11
      TR2=TR22
      TI2=TI22
      TR3=2.*TR33
      TI3=2.*TI33
      TR4=TR44
      TI4=TI44
      RMAX=ABS(TR5)
      IF(RMAX.LT.ABS(TI5)) RMAX=TI5
      IF(RMAX.LT.ABS(TI4)) RMAX=TI4
      IF(RMAX.LT.ABS(TI3)) RMAX=TI3
      IF(RMAX.LT.ABS(TI2)) RMAX=TI2
      IF(RMAX.LT.ABS(TI1)) RMAX=TI1
      IF(RMAX.LT.ABS(TR4)) RMAX=TR4
      IF(RMAX.LT.ABS(TR3)) RMAX=TR3
      IF(RMAX.LT.ABS(TR2)) RMAX=TR2
      IF(RMAX.LT.ABS(TR1)) RMAX=TR1
      RMAX=1./RMAX
      TR1=TR1*RMAX
      TR2=TR2*RMAX
      TR3=TR3*RMAX
      TR4=TR4*RMAX
      TR5=TR5*RMAX
      TI1=TI1*RMAX
      TI2=TI2*RMAX
      TI3=TI3*RMAX
      TI4=TI4*RMAX
      TI5=TI5*RMAX
1000  CONTINUE
2000  CONTINUE
C   SET MATRIX ELEMENTS (8)
      P=A(1)
      P2=P*P
      S=B(1)
      S2=S*S
      RRO=RHO(1)
      ARGS=1.-C2/S2
      ARGP=1.-C2/P2
```

```
      IF(ARGP.GE.0.) CN=CMPLX(0.,-RK*SQRT(ARGP))
      IF(ARGP.LT.0.) CN=CMPLX(RK*SQRT(-ARGP),0.)
      IF(ARGS.LT.0.) CNS=CMPLX(RK*SQRT(-ARGS),0.)
      IF(ARGS.GE.0.) CNS=CMPLX(0.,-RK*SQRT(ARGS))
      RM=RRO*S2
      RL=RK2+RK2-OM2/S2
      RPP=CN*CNS
      RM2=RM*RM
      RL2=RL*RL
      T11=CMPLX(-RK2,0.)
      T13=T11+RPP
      T11=T11-RPP
      T21=CMPLX(0.,RRO*OM2)
      T51=T21*CN
      T21=-T21*CNS
      T31=CMPLX(0.,-RM*RK*RL)
      RSS=CMPLX(0.,2.*RM*RK)*RPP
      T33=T31+RSS
      T31=T31-RSS
      T61=CMPLX(-RM2*RL2,0.)
      RSS=CMPLX(4.*RK2*RM2,0.)*RPP
      T63=T61+RSS
      T61=T61-RSS
      T23=CMPLX(0.,RM*(2.*RK2-RL))
      T53=T23*CN
      T23=T23*CNS
      T12=CMPLX(RK+RK,0.)
      T15=T12*CNS
      T12=T12*CN
      T32=CMPLX(0.,4.*RM*RK2)
      T45=T32*CNS
      T32=T32*CN
      T42=CMPLX(0.,2.*RM*RL)
      T35=T42*CNS
      T42=T42*CN
      T62=CMPLX(4.*RM2*RL*RK,0.)
      T65=T62*CNS
      T62=T62*CN
      T1=CMPLX(TR1,TI1)
      T2=CMPLX(TR2,TI2)
      T3=CMPLX(TR3,TI3)
      T4=CMPLX(TR4,TI4)
      T5=CMPLX(TR5,TI5)
C   DO LAST PART OF MATRIX MULTIPLICATION (5)
C   COMPUTE REFLECTION COEFFICIENTS (4)
      DET=T1*T11+T2*T21+T3*T31+T4*T51+T5*T61
      DET=CMPLX(1.,0.)/DET
      RSS=T1*T13+T2*T23+T3*T33+T4*T53+T5*T63
      RSS=-RSS*DET
      RPP=-T1*T13-T2*T21-T3*T33+T4*T51-T5*T63
      RPP=RPP*DET
      T3=T3*CMPLX(0.5,0.)
      RPS=T1*T12+T3*T32+T3*T42+T5*T62
      RPS=-RPS*DET
      RSP=T1*T15+T3*T35+T3*T45+T5*T65
      RSP=RSP*DET
      RETURN
      END
```

Appendix 2. Computer program for the computation of reflection coefficients for a layered liquid and for *SH* reflection coefficients for a layered solid.

```
      SUBROUTINE RECOPP(N,A,RHO,D,U,FREQ,RPP)
C
C   COMPUTATION OF REFLECTION COEFFICIENTS FOR A LAYERED LIQUID
C
C   N= NUMBER OF DIFFERENT MEDIA, STARTING ON TOP
C   A(I),RHO(I),(I=1,N)= VELOCITY AND DENSITY
C   D(I),(I=2,N-1)= LAYER THICKNESS
C   U= PHASE SLOWNESS, FREQ= FREQUENCY
C   RPP= COMPLEX REFLECTION COEFFICIENT
C
C   FOR COMPUTATION OF SH REFLECTION COEFFICIENTS REPLACE A BY
C   SHEAR VELOCITY B AND RHO BY 1./(B*B*RHO)
C
      DIMENSION  A(N),RHO(N),D(N)
      COMPLEX RPP,NI,NIP,ROI,ROIP,M21,M22,E2,E1,E
      D(1)=0.
      PI=3.14159265
      OMEGA=2.*PI*FREQ
      OM2=OMEGA*OMEGA
      XK=OMEGA*U
      XK2=XK*XK
      M22=CMPLX(1.,0.)
      M21=CMPLX(0.,0.)
```

```
C     DO MATRIX MULTIPLICATION (11) FROM LEFT TO RIGHT
C     USING (12) AND DO NORMALIZATION
      DO 170 J=1,N
      I=N-J+1
      ARG=OM2/(A(I)*A(I))-XK2
      IF(ARG.GT.0.) NI=CMPLX(SQRT(ARG),0.)
      IF(ARG.LE.0.) NI=CMPLX(0.,-SQRT(-ARG))
      ROI=CMPLX(RHO(I),0.)
      IF(I.EQ.N) GOTO 171
      E1=NIP*ROI
      E2=NI*ROIP
      E=CEXP(NI*CMPLX(0.,2.*D(I)))
      E1=E1*(M21+M22)
      E2=E2*(M21-M22)
      M21=E1+E2
      M22=E*(E1-E2)
      RMAX=CABS(M22)
      RM=CABS(M21)
      IF(RM.GT.RMAX) RMAX=RM
      E1=CMPLX(1./RMAX,0.)
      M22=M22*E1
      M21=M21*E1
171   NIP=NI
      ROIP=ROI
170   CONTINUE
      RPP=-M21/M22
      RETURN
      END
```

References

Červený, V.: Reflection and transmission coefficients for transition layers. Studia geophys. geodaet. **18**, 59-68, 1974

Dunkin, J.W.: Computation of modal solutions in layered media at high frequencies. Bull. Seism. Soc. Am. **55** (2), 335-358, 1965

Fuchs, K.: Das Reflexions- und Transmissionsvermögen eines geschichteten Mediums mit beliebiger Tiefenverteilung der elastischen Moduln und der Dichte für schrägen Einfall ebener Wellen. Z. Geophys. **34**, 389-413, 1968

Fuchs, K., Müller, G.: Computation of synthetic seismograms with the reflectivity method and comparison with observations. Geophys. J.R.A.S. **23**, 417-433, 1971

Kind, R., Müller, G.: Computation of *SV* waves in realistic earth models. J. Geophys. **41**, 149-172, 1975

Müller, G., Kind, R.: Observed and computed seismogram sections for the whole earth. Geophys. J.R.A.S. **44**, 699-716, 1976

Pestel, E., Leckie, F.A.: Matrix methods in elastomechanics. New York: McGraw Hill 1963

Satô, Y.: 2. Study on surface waves *X*. Equivalency of *SH*-waves and sound waves in a liquid. Bull. Earthquake Res. Inst. **32**, 7-16, 1954

Schwab, F.A., Knopoff, L.: Fast surface wave and free mode computations. Methods in computational physics, Vol. II, B.A. Bolt, ed. New York: Academic Press 1972

Watson, T.H.: Fast computation of Rayleigh wave dispersion in a layered halfspace. Bull. Seism. Soc. Am. **60**, 161-166, 1970

Received May 28, 1976