27 August 1975: se

Slant Multiple Reflections

by Anupam Khanna

The following material extends Claerbout's work (Slanted Multiple Reflection Calculation-SEP 7, p. 1) to include interbed multiples.

The geometry is the same as before, and once again we assume slant stacking of the data. This can be viewed as a line array of shots which sends a plane wave traveling down at a certain angle. Because of slant stacking we can restrict our attention to a fixed angle -- waves traveling at other angles interfere destructively. Figure 1 illustrates the basic geometry and indexing.

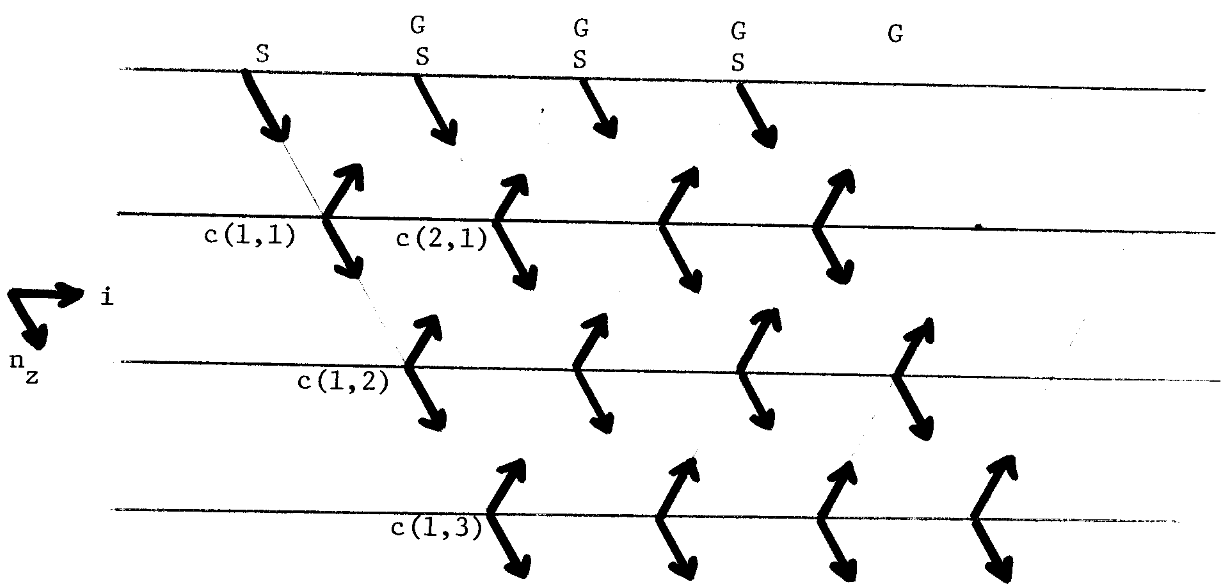

Fig. 1. Basic Geometry.

A key point to be noted is that since we consider fixed angles with the vertical there is a definite interrelationship between the x, z and t coordinates. In particular, the received signal at the discrete time instant T is wholly due to the shot at location T lags away and reflected through the T top layers. Figure 2 illustrates this for T=3 .

Another notable point is that the computation is done in the natural frame for upcoming waves.

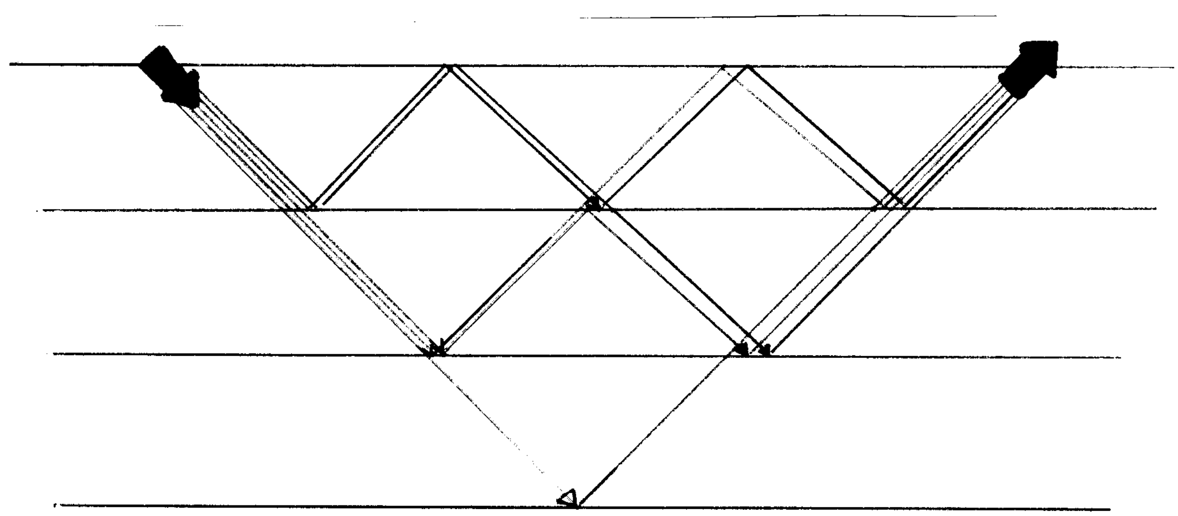

Fig. 2. Illustration for T=3. The receiver waveform at the 3rd time instant is wholly due to shot 3 unit offsets (lags) away. Waves penetrating deeper shall take a longer time to be effected back to the surface and shall travel further along x too.

Forward Problem

The algorithm illustrated in Figure 3 proceeds by identifying the various downgoing waves at different layers that contribute to the received signal at a receiver location for a particular time instant. We sum their contributions sequentially, starting from the bottom, i.e., "traveling with the upcoming wave" to determine the surface received waveform.

Concurrently we also determine the downgoing waves which shall contribute to the received signal at the next time instant at the next x-location.

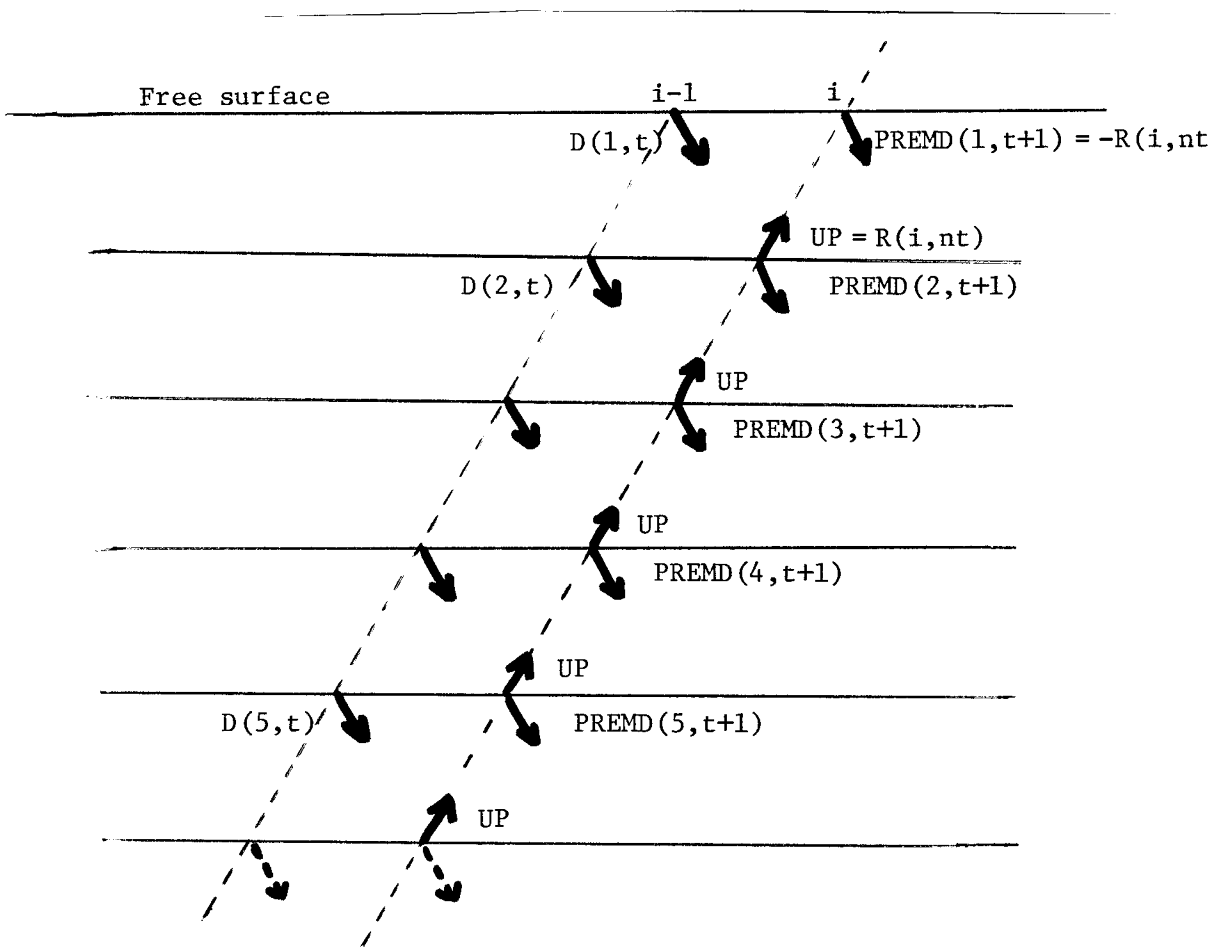

Fig. 3. The D(nz,t)'s all contribute to R(i,t) which is the final value of the 'UP' wave when it reaches the surface. The D's and c's are used to calculate UP and PREMD's starting from the bottom. At the end of the cycle the PREMD's are converted to D's for the next location.

Some Variations to the Forward Algorithm

In a considerable number of situations we are interested in modeling or generating synthetics due to only a few major events, eg. peglegs, bright spots etc. In these cases the computational effort can be reduced by a factor of (# of events) ÷ (NMAX) by slightly modifying the algorithm.

Instead of moving in steps of NZ we move in varying jumps along the upcoming wave UP from one nonzero reflection coefficient (i.e., major event) to the next higher one. This is illustrated in Figure 4.

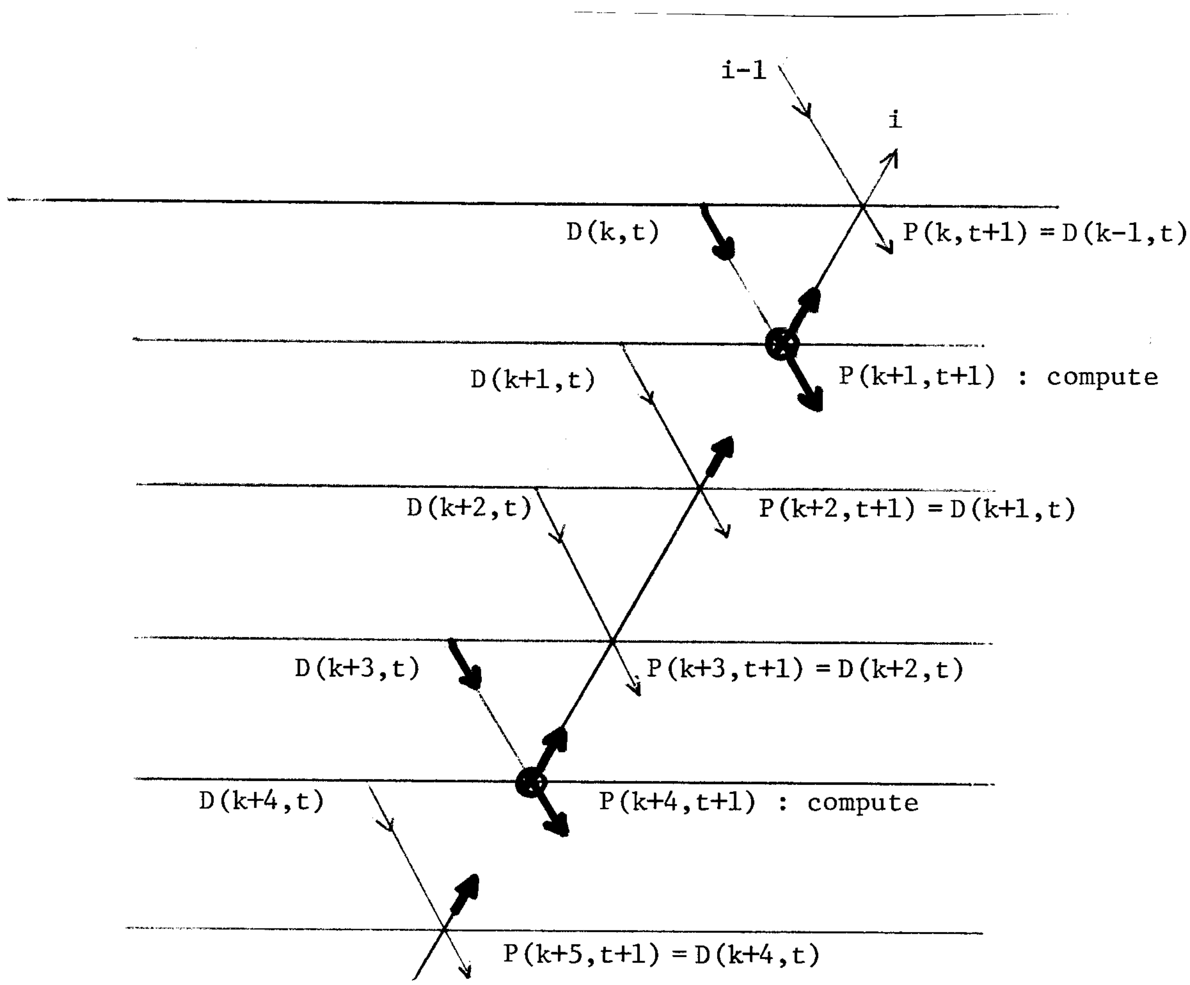

Fig. 4: ⊗ indicate nonzero c's , i.e., at nz=k, k+3 . Hence we need to compute P(nz+1,t+1) = D(nz,t)*(1.0-c(nz))-UP*c(nz) only for nz = k, k+3. For nz ≠ k, k+3 P(nz+1,t+1) = D(nz,t). A pointer L(.) is used to identify the major event locations.

The pointer for locating the nonzero c's can be either fed in directly with the data or can be generated as a part of the algorithm by simply searching along each upcoming wave path for each i . The latter method was used as illustrated in Figure 1.

Very often the sampling along the x and z (or time) directions needs to be varied corresponding to different propagation angles of the wave. In most practical situations, the wave moves at only a slight angle from the vertical and goes through several multiple reflections even before reaching the first receiver. This corresponds to coarse sampling along x .

The actual algorithm illustrated in Figure 16 is analogous to the earlier ones. Besides obvious changes in the shifting indices we also need to take care of the end effect at the top of each trace.

A single step of the algorithm is illustrated in Figure 6.

Another variation is the algorithm illustrated in Figure 17 which corresponds to the situation where the c's are sampled more densely along the x-direction than along the z-direction, i.e., we basically skip vertical layers of c's .

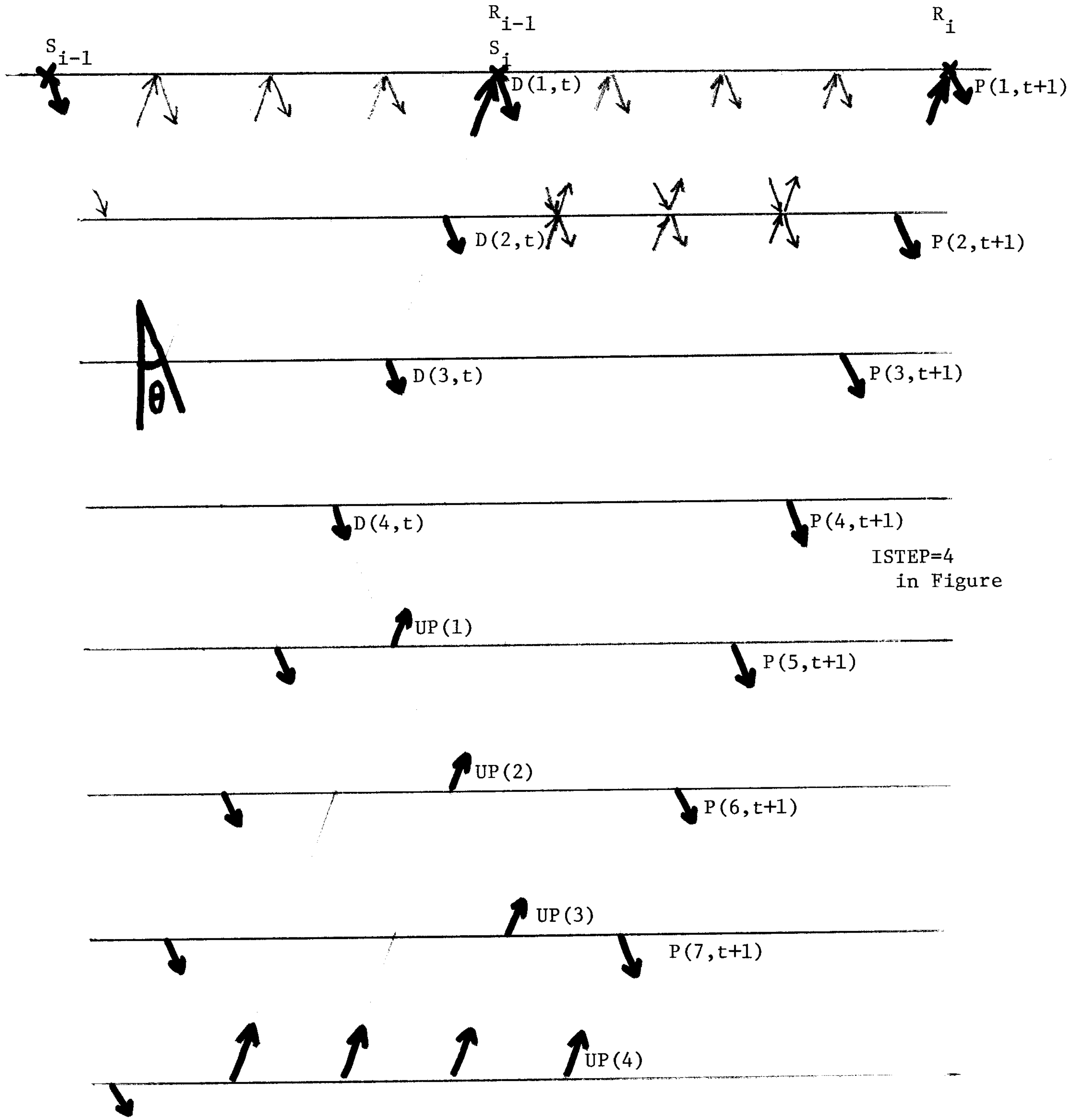

Fig. 5. Coarse sampling along x : We need to interpolate the reflection coefficients at (ISTEP) points for each depth. Also we need to move along and compute along (ISTEP) upcoming waves. Note again that the time unit now is the time it requires for the first received signal to arrive, i.e. (ISTEP)*2*travel time through a layer.

$$\text{ISTEP} = \frac{\Delta x}{\Delta z} * \frac{1.0}{2.0*\text{TAN}\theta}$$

Incrementing NZ by 1 corresponds to moving down Δz .
Incrementing I by 1 corresponds to moving down Δx .

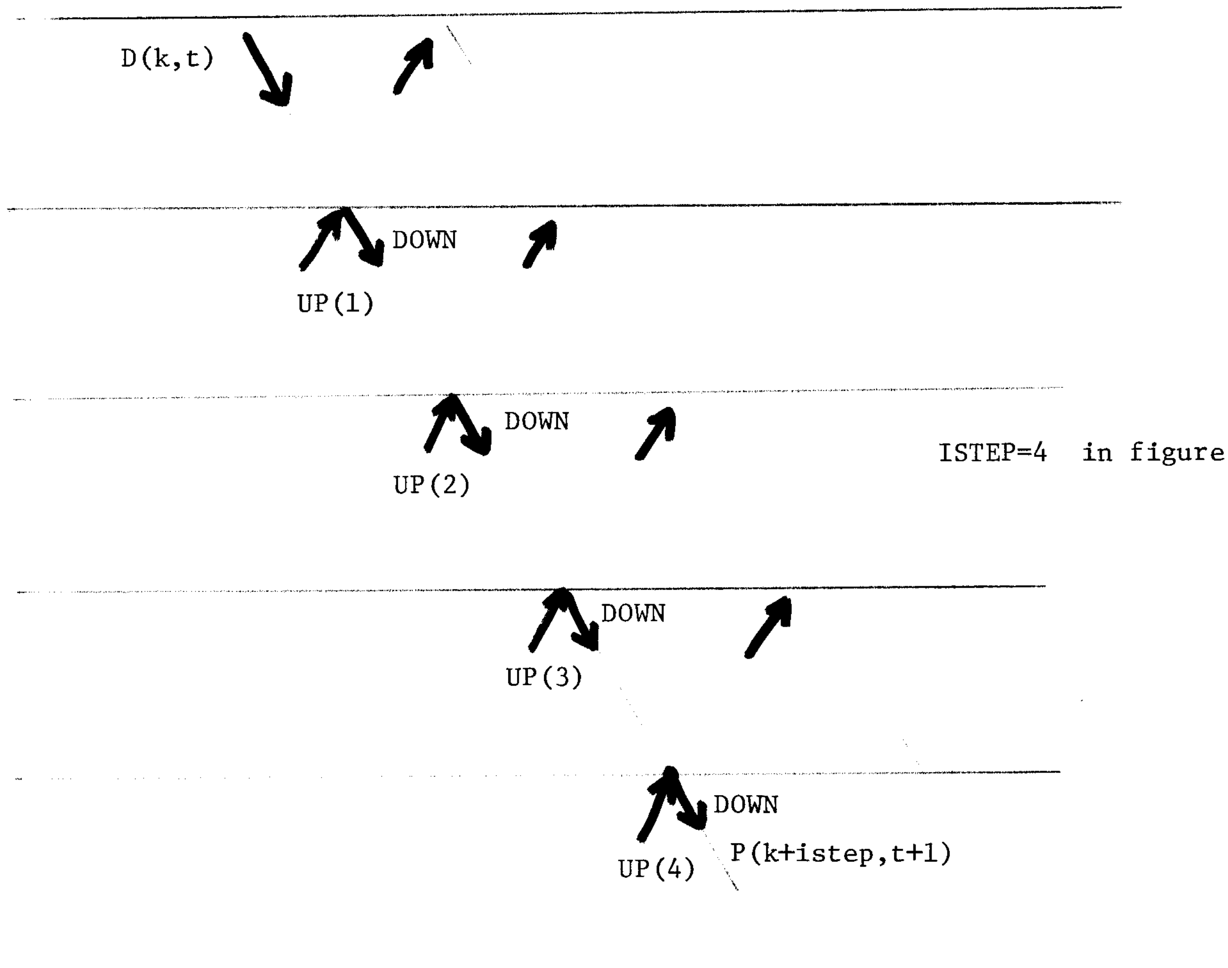

Fig. 6. For each D(k,t) proceed down along its direction and using UP(J) (J=1, ISTEP), successively compute new UP(J)'s and downward continue DOWN, i.e.,

```
     DOWN = D(K,T)
     DO 10 J=1, ISTEP
     UP(J) =UP(J)*(1.+C(K+1-J))-DOWN*C(K+1-J)
10   DOWN=UP(J)*C(K+1-J)+DOWN*(1.-C(K+1-J))
     P(K+ISTEP,T+1) = DOWN
```

Note that a shot 'IO' offset units away can only contribute to P(IO*ISTEP+1,T+1).

Inverse Problem

The inverse algorithm is analogous to the forward case. Once again the computation proceeds "along the upcoming wave", but in this case we move down from the receiver successively evaluating the reflection coefficients and subtracting the contributions of multiple reflections in the layers above. Figure 7 illustrates the procedure.

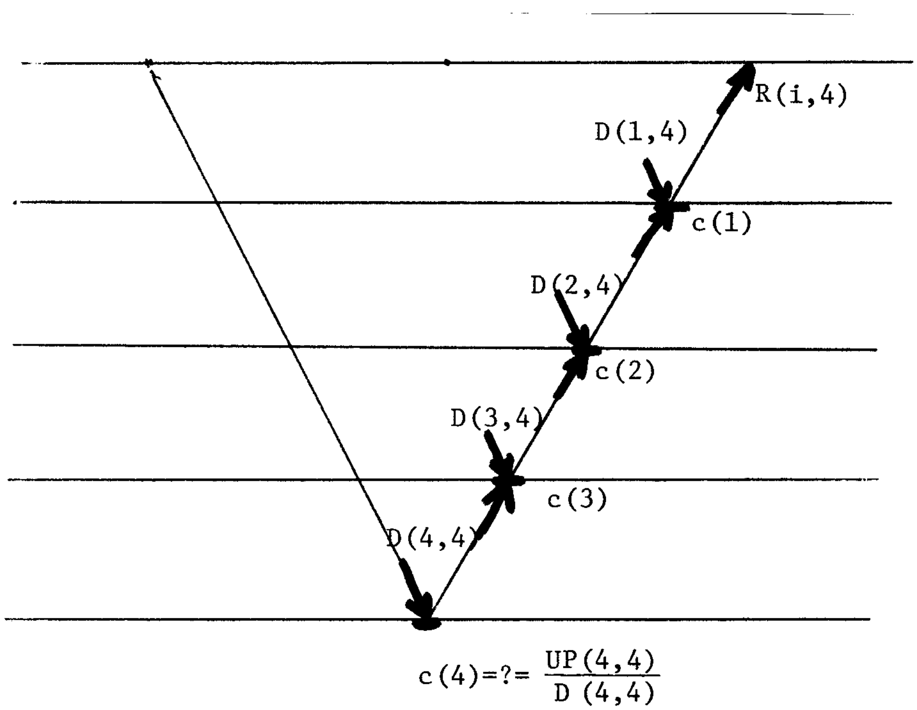

Fig. 7. Inverse Algorithm.

To determine the reflection coefficient at the nz^{th} layer we start with the received signal at $t=nz$ and go down as above since we have previously calculated the reflection coefficients at shallower levels until we meet the downgoing wave from shot nz lags before the receiver. Their ratio determines the reflection coefficient and this is used to downward continue the received signal at $t=nz+1$.

```
      SUBROUTINE FORWD(C,R,D,PREMD,IMAX,NZMAX,NZM)
      REAL UP
      DIMENSION PREMD(IMAX,NZM)
      DIMENSION C(IMAX,NZMAX),D(IMAX,NZM),R(IMAX,NZMAX)
      DO 240 I=2,IMAX
      R(I,1)=-C(I-1,1)*D(I-1,1)
      D(I,2)=R(I,1)+D(I-1,1)
240   CONTINUE
      DO 41 I=2,IMAX
41    D(I,1)=-R(I,1)
      DO 100 NT=2,NZMAX
      NTONE=NT+1
      DO 60 I=NTONE,IMAX
      UP=-C(I-NT,NT)*D(I-1,NT)
      PREMD(I,NT+1)=D(I-1,NT)*(1.0-C(I-NT,NT))
      DO 250 K=2,NT
      J=NT+2-K
      PREMD(I,J)=UP*C(I-J+1,J-1)+D(I-1,J-1)*(1.0-C(I-J+1,J-1))
250   UP=UP*(1.0+C(I-J+1,J-1))-C(I-J+1,J-1)*D(I-1,J-1)
      R(I,NT)=UP
60    PREMD(I,1)=-UP
      DO 70 I=NTONE,IMAX
      DO 70 K=1,NTONE
70    D(I,K)=PREMD(I,K)
100   CONTINUE
      RETURN
      END
```

Fig. 8. The Basic Forward Subroutine FORWD with 'equal' sampling along x- and z-directions. Suitable when number of major events (i.e., nonzero reflection coefficients) is of the same order as NZMAX.

```
      SUBROUTINE INVRS(R,DI,PREMDI,UPI,CI,IMAX,NZMAX,NZM,MCI)
      DIMENSION R(IMAX,NZMAX),DI(NZM,NZM),PREMDI(NZM,NZM)
      DIMENSION CI(NZMAX),UPI(NZMAX,NZMAX),MCI(NZMAX)
      DI(1,1)=100.0
      PREMDI(1,1)=100.0
      CI(1)=-R(1,1)/100.0
      DI(2,2)=100.0*(1.0-CI(1))
      DO 123 NZ=3,NZMAX
  123 DI(2,NZ)=0.0
      DO 200 I=2,IMAX
      IF(I.LT.NZMAX) GO TO 111
      KMAX=NZMAX
      GO TO 112
  111 KMAX=I
  112 DO 150 NT=2,KMAX
      DI(1,NT)=-R(I-1,NT-1)
      PREMDI(1,NT)=-R(I,NT-1)
  150 UPI(1,NT)=R(I,NT)
      CI(1)=-R(I,1)/100.0
      DO 170 NZ=2,KMAX
      NZI=NZ-1
      PREMDI(NZ,NZ)=DI(NZI,NZI)*(1.0-CI(NZI))
      CP=CI(NZI)
      DO 180 NT=NZ,KMAX
      UPI(NZ,NT)=(UPI(NZI,NT)+CP*DI(NZI,NT))/(1.0+CP)
  180 PREMDI(NZ,NT+1)=(CP*UPI(NZI,NT)+DI(NZI,NT))/(1.0+CP)
      CI(NZ)=-UPI(NZ,NZ)/DI(NZ,NZ)
  170 CONTINUE
      KM=KMAX+1
      IF(KM.GT.NZMAX) GO TO 165
      PREMDI(KM,KM)=DI(KMAX,KMAX)*(1.0-CI(KMAX))
      DO 160 NZ=KM,NZMAX
  160 CI(NZ)=0.0
      GO TO 166
  165 KM=NZMAX
  166 DO 140 NZ=2,KM
      DO 140 NT=NZ,KM
  140 DI(NZ,NT)=PREMDI(NZ,NT)
      DO 103 NZ=1,KMAX
  103 MCI(NZ)=100.5*CI(NZ)
      WRITE (6,104) I,(MCI(NZ),NZ=1,KMAX)
  104 FORMAT(20X,12I6)
  200 CONTINUE
      RETURN
      END
```

Fig. 9. The basic Inverse Subroutine INVRS with equal sampling along x- and z-directions.

```
$WATFIV
      DIMENSION C(52,10),D(52,11),PREMD(52,11),R(52,10)
      DIMENSION CI(10),UPI(10,10),DI(11,11),PREMDI(11,11),MCI(10)
      REAL UP
      IMAX=52
      NZMAX=10
      NZM=NZMAX+1
      DO 211 I=1,IMAX
      DO 210 NT=1,NZMAX
  210 R(I,NT)=0.0
  211 D(I,1)=100.0
      CALL DATA(C)
      CALL OUT(IMAX,NZMAX,C)
      CALL FORWD(C,R,D,PREMD,IMAX,NZMAX,NZM)
      CALL OUTR(IMAX,NZMAX,R)
      CALL INVRS(R,DI,PREMDI,UPI,CI,IMAX,NZMAX,NZM,MCI)
      STOP
      END
      SUBROUTINE DATA(C)
      DIMENSION C(52,10)
      DO 10 I=1,52
      DO 20 NT=1,10
   20 C(I,NT)=0.0
   10 C(I,5)=0.1
      DO 30 I=1,20
   30 C(I,2)=0.8
      DO 40 I=40,52
   40 C(I,2)=0.8
      RETURN
      END
      SUBROUTINE OUT(IMAX,NZMAX,C)
      DIMENSION C(IMAX,NZMAX),LINE(100)
      PRINT 310
  310 FORMAT('1 NEXT SECTION')
      DO 330 I=1,IMAX
      DO 320 NZ=1,NZMAX
  320 LINE(NZ)=100.5*C(I,NZ)
  330 PRINT 340,I,(LINE(NZ),NZ=1,NZMAX)
  340 FORMAT(20X,12I6)
      RETURN
      END
      SUBROUTINE OUTR(IMAX,NZMAX,R)
      DIMENSION R(IMAX,NZMAX),LINE(100)
      PRINT 310
  310 FORMAT('1 NEXT SECTION')
      DO 330 I=1,IMAX
      DO 320 NZ=1,NZMAX
  320 LINE(NZ)=R(I,NZ)
  330 PRINT 340,I,(LINE(NZ),NZ=1,NZMAX)
  340 FORMAT(20X,12I6)
      RETURN
      END
```

Fig. 10. Test program including data and display subroutines namely DATA and OUT.

IX NZ

1	0	80	0	0	10	0	0	0	0	0
2	0	80	0	0	10	0	0	0	0	0
3	0	80	0	0	10	0	0	0	0	0
4	0	80	0	0	10	0	0	0	0	0
5	0	80	0	0	10	0	0	0	0	0
6	0	80	0	0	10	0	0	0	0	0
7	0	80	0	0	10	0	0	0	0	0
8	0	80	0	0	10	0	0	0	0	0
9	0	80	0	0	10	0	0	0	0	0
10	0	80	0	0	10	0	0	0	0	0
11	0	80	0	0	10	0	0	0	0	0
12	0	80	0	0	10	0	0	0	0	0
13	0	80	0	0	10	0	0	0	0	0
14	0	80	0	0	10	0	0	0	0	0
15	0	80	0	0	10	0	0	0	0	0
16	0	80	0	0	10	0	0	0	0	0
17	0	80	0	0	10	0	0	0	0	0
18	0	80	0	0	10	0	0	0	0	0
19	0	80	0	0	10	0	0	0	0	0
20	0	80	0	0	10	0	0	0	0	0
21	0	0	0	0	10	0	0	0	0	0
22	0	0	0	0	10	0	0	0	0	0
23	0	0	0	0	10	0	0	0	0	0
24	0	0	0	0	10	0	0	0	0	0
25	0	0	0	0	10	0	0	0	0	0
26	0	0	0	0	10	0	0	0	0	0
27	0	0	0	0	10	0	0	0	0	0
28	0	0	0	0	10	0	0	0	0	0
29	0	0	0	0	10	0	0	0	0	0
30	0	0	0	0	10	0	0	0	0	0
31	0	0	0	0	10	0	0	0	0	0
32	0	0	0	0	10	0	0	0	0	0
33	0	0	0	0	10	0	0	0	0	0
34	0	0	0	0	10	0	0	0	0	0
35	0	0	0	0	10	0	0	0	0	0
36	0	0	0	0	10	0	0	0	0	0
37	0	0	0	0	10	0	0	0	0	0
38	0	0	0	0	10	0	0	0	0	0
39	0	0	0	0	10	0	0	0	0	0
40	0	80	0	0	10	0	0	0	0	0
41	0	80	0	0	10	0	0	0	0	0
42	0	80	0	0	10	0	0	0	0	0
43	0	80	0	0	10	0	0	0	0	0
44	0	80	0	0	10	0	0	0	0	0
45	0	80	0	0	10	0	0	0	0	0
46	0	80	0	0	10	0	0	0	0	0
47	0	80	0	0	10	0	0	0	0	0
48	0	80	0	0	10	0	0	0	0	0
49	0	80	0	0	10	0	0	0	0	0
50	0	80	0	0	10	0	0	0	0	0
51	0	80	0	0	10	0	0	0	0	0
52	0	80	0	0	10	0	0	0	0	0

Fig. 11. The given model. Peglegs are expected.

IX \ NT										
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	-80	0	0	0	0	0	0	0	0
4	0	-80	0	0	0	0	0	0	0	0
5	0	-80	0	-64	0	0	0	0	0	0
6	0	-80	0	-64	-3	0	0	0	0	0
7	0	-80	0	-64	-3	-51	0	0	0	0
8	0	-80	0	-64	-3	-51	-5	0	0	0
9	0	-80	0	-64	-3	-51	-5	-40	0	0
10	0	-80	0	-64	-3	-51	-5	-40	-6	0
11	0	-80	0	-64	-3	-51	-5	-40	-6	-32
12	0	-80	0	-64	-3	-51	-5	-40	-6	-32
13	0	-80	0	-64	-3	-51	-5	-40	-6	-32
14	0	-80	0	-64	-3	-51	-5	-40	-6	-32
15	0	-80	0	-64	-3	-51	-5	-40	-6	-32
16	0	-80	0	-64	-3	-51	-5	-40	-6	-32
17	0	-80	0	-64	-3	-51	-5	-40	-6	-32
18	0	-80	0	-64	-3	-51	-5	-40	-6	-32
19	0	-80	0	-64	-3	-51	-5	-40	-6	-32
20	0	-80	0	-64	-3	-51	-5	-40	-6	-32
21	0	-80	0	-64	-3	-51	-5	-40	-6	-32
22	0	-80	0	-64	-3	-51	-5	-40	-6	-32
23	0	0	0	0	-1 *	0	-1	0	-1	0
24	0	0	0	0	-1 *	0	-1	0	-1	0
25	0	0	0	0	-1 *	0	-1	0	-1	0
26	0	0	0	0	-10	0	-8	0	-6	0
27	0	0	0	0	-10	0	-8	0	-6	0
28	0	0	0	0	-10	0	0	0	0	0
29	0	0	0	0	-10	0	0	0	0	0
30	0	0	0	0	-10	0	0	0	0	0
31	0	0	0	0	-10	0	0	0	0	-1
32	0	0	0	0	-10	0	0	0	0	-1
33	0	0	0	0	-10	0	0	0	0	-1
34	0	0	0	0	-10	0	0	0	0	-1
35	0	0	0	0	-10	0	0	0	0	-1
36	0	0	0	0	-10	0	0	0	0	-1
37	0	0	0	0	-10	0	0	0	0	-1
38	0	0	0	0	-10	0	0	0	0	-1
39	0	0	0	0	-10	0	0	0	0	-1
40	0	0	0	0	-10	0	0	0	0	-1
41	0	0	0	0	-10	0	0	0	0	-1
42	0	-80	0	0	-17 *	0	-8	0	0	-1
43	0	-80	0	0	-17 *	0	-8	0	0	-1
44	0	-80	0	-64	-17 *	0	-14	0	-6	-1
45	0	-80	0	-64	-3	0	-14	1	-6	0
46	0	-80	0	-64	-3	-51	-14	1	-11	0
47	0	-80	0	-64	-3	-51	-5	1	-11	0
48	0	-80	0	-64	-3	-51	-5	-40	-11	0
49	0	-80	0	-64	-3	-51	-5	-40	-6	0
50	0	-80	0	-64	-3	-51	-5	-40	-6	-32
51	0	-80	0	-64	-3	-51	-5	-40	-6	-32
52	0	-80	0	-64	-3	-51	-5	-40	-6	-32

Fig. 12. The received waveform time series. Note peglegs and the signals indicated by *** . This behavior is explained in Fig. 14.

IX \ NZ										
[illegible]		0								
3	0	80	0							
4	0	80	0	0						
5	0	80	0	0	0					
6	0	80	0	0	10	0				
7	0	80	0	0	10	0	0			
8	0	80	0	0	10	0	0	0		
9	0	80	0	0	10	0	0	0	0	
10	0	80	0	0	10	0	0	0	0	0
11	0	80	0	0	10	0	0	0	0	0
12	0	80	0	0	10	0	0	0	0	0
13	0	80	0	0	10	0	0	0	0	0
14	0	80	0	0	10	0	0	0	0	0
15	0	80	0	0	10	0	0	0	0	0
16	0	80	0	0	10	0	0	0	0	0
17	0	80	0	0	10	0	0	0	0	0
18	0	80	0	0	10	0	0	0	0	0
19	0	80	0	0	10	0	0	0	0	0
20	0	80	0	0	10	0	0	0	0	0
21	0	80	0	0	10	0	0	0	0	0
22	0	80	0	0	10	0	0	0	0	0
23	0	0	0	0	10	0	0	0	0	0
24	0	0	0	0	10	0	0	0	0	0
25	0	0	0	0	10	0	0	0	0	0
26	0	0	0	0	10	0	0	0	0	0
27	0	0	0	0	10	0	0	0	0	0
28	0	0	0	0	10	0	0	0	0	0
29	0	0	0	0	10	0	0	0	0	0
30	0	0	0	0	10	0	0	0	0	0
31	0	0	0	0	10	0	0	0	0	0
32	0	0	0	0	10	0	0	0	0	0
33	0	0	0	0	10	0	0	0	0	0
34	0	0	0	0	10	0	0	0	0	0
35	0	0	0	0	10	0	0	0	0	0
36	0	0	0	0	10	0	0	0	0	0
37	0	0	0	0	10	0	0	0	0	0
38	0	0	0	0	10	0	0	0	0	0
39	0	0	0	0	10	0	0	0	0	0
40	0	0	0	0	10	0	0	0	0	0
41	0	0	0	0	10	0	0	0	0	0
42	0	80	0	0	10	0	0	0	0	0
43	0	80	0	0	10	0	0	0	0	0
44	0	80	0	0	10	0	0	0	0	0
45	0	80	0	0	10	0	0	0	0	0
46	0	80	0	0	10	0	0	0	0	0
47	0	80	0	0	10	0	0	0	0	0
48	0	80	0	0	10	0	0	0	0	0
49	0	80	0	0	10	0	0	0	0	0
50	0	80	0	0	10	0	0	0	0	0
51	0	80	0	0	10	0	0	0	0	0
52	0	80	0	0	10	0	0	0	0	0

Fig. 13. Model reconstructed from received time series. Note that this is exactly the same as the original model.

Notice in Figure 9, the odd behavior of the "peglegs" (indicated by (***). In our formulation it apparently makes a difference if we traverse it upwards or downwards. Figure 14 shows this simply. The reason behind seems to be that we have not truly taken care of the lateral impedance variation.

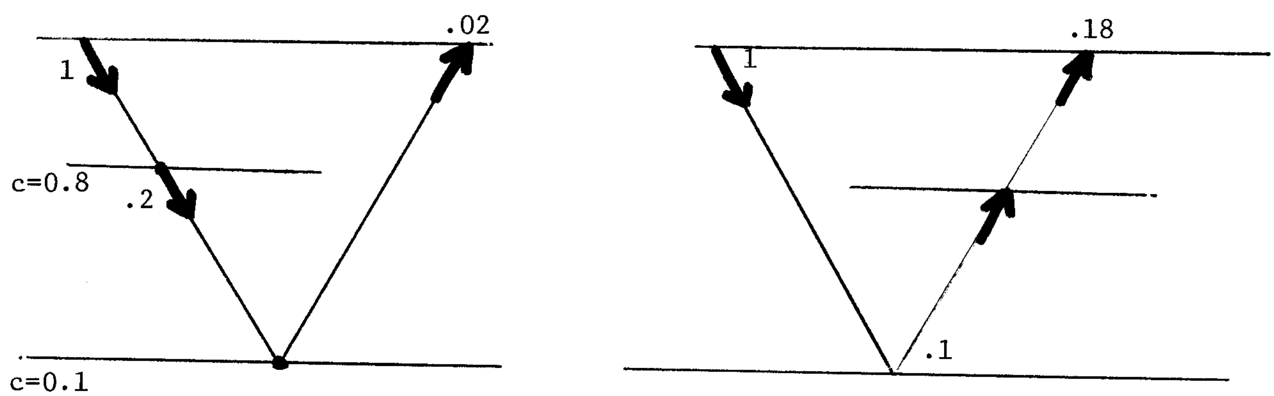

Fig. 14. Pegleg edge behavior.

Another example illustrating an elementary bright spot is illustrated in Figures 18 through 20.

```
      SUBROUTINE FWDXO(C,CF,R,D,PREMD,IMAX,NZMAX,NZM,I,UP)
      DIMENSION C(IMAX,NZMAX),R(IMAX,NZMAX),D(NZM,NZM),PREMD(NZM,NZM)
      DIMENSION CF(NZMAX),L(3)
      D(1,1)=100.0
      PREMD(1,1)=100.0
      I=1
      R(1,1)=-C(1,1)*D(1,1)
      D(1,2)=-R(1,1)
      D(2,2)=R(1,1)+D(1,1)
      DO 425 NZ=1,NZMAX
      DO 425 NT=3,NZMAX
  425 D(NZ,NT)=0.0
      WRITE(6,429) I,R(1,1)
  429 FORMAT(5X,I6,F7.2)
      DO 100 I=2,IMAX
      IF(I.LT.NZMAX) GO TO 430
      NTMAX=NZMAX
      GO TO 431
  430 NTMAX=I
  431 KMAX=0
      DO 410 NZ=1,NTMAX
      CF(NZ)=C(I-NZ+1,NZ)
      IF(CF(NZ).EQ.0.0) GO TO 410
      KMAX=KMAX+1
      L(KMAX)=NZ
  410 CONTINUE
      R(I,1)=-CF(1)*D(1,1)
      PREMD(1,2)=-R(I,1)
      PREMD(2,2)=R(I,1)+D(1,1)
      DO 470 NZ=1,NTMAX
      DO 470 NT=NZ,NTMAX
  470 PREMD(NZ+1,NT+1)=D(NZ,NT)
      DO 460 NT=2,NTMAX
      UP=-CF(NT)*D(NT,NT)
      NTPLS=NT+1
      NTMIN=NT-1
      PREMD(NTPLS,NTPLS)=UP+D(NT,NT)
      KM=KMAX
  415 IF(L(KM).LT.NT) GO TO 416
      KM=KM-1
      IF(KM.EQ.0) GO TO 421
      GO TO 415
  416 DO 420 K=1,KM
      J=L(KM-K+1)
      PREMD(J+1,NTPLS)=UP*CF(J)+D(J,NT)*(1.0-CF(J))
  420 UP=UP*(1.0+CF(J))-D(J,NT)*CF(J)
  421 R(I,NT)=UP
  460 PREMD(1,NTPLS)=-UP
      NTM=NZMAX
      IF(I.LT.NZMAX) NTM=NTMAX+1
      DO 480 NZ=1,NTM
      DO 480 NT=NZ,NTM
  480 D(NZ,NT)=PREMD(NZ,NT)
      WRITE(6,471) I,(R(I,NT),NT=1,NTMAX)
  471 FORMAT(5X,I6,12F7.2)
  100 CONTINUE
      RETURN
      END
```

Fig. 15. Variation of the forward subroutine to reduce the number of computations when there are only a few nonzero reflection coefficients. Pointers are generated within the program.

```
$WATFIV
      DIMENSION UP(4),D(12,5),P(17,6),R(5),CD(50,12),C(12)
      IMAX=50
      NZMAX=12
      NTM=5
      NTMAX=20
      ISTEP=4
      ISTMIN=ISTEP-1
      DO 1 NZ=1,NZMAX
      DO 2 NT=1,NTM
      D(NZ,NT)=0.0
    2 P(NZ,NT)=0.0
      DO 3 I=1,IMAX
    3 CD(I,NZ)=0.0
    1 CONTINUE
      DO 4 I=1,IMAX
    4 CD(I,5)=0.1
      DO 5 I=1,20
      CD(I,2)=0.8
    5 CD(30+I,2)=0.8
      D(1,1)=1.0
      NT=1
      DOWN=D(1,1)
      DO 10 J=1,ISTEP
      C(J)=CD(1,J)
      UP(J)=-C(J)*DOWN
   10 DOWN=DOWN+UP(J)
      P(ISTEP+1,2)=DOWN
      DO 30 L=1,ISTMIN
      DOWN=-UP(L)
      JMIN=L+1
      DO 40 J=JMIN,ISTEP
      DOWN2=DOWN*(1.0-C(J-L))+UP(J)*C(J-L)
      UP(J)=DOWN2+UP(J)-DOWN
   40 DOWN=DOWN2
   30 P(ISTEP+1-L,NT+1)=DOWN
      R(NT)=UP(ISTEP)
      P(1,NT+1)=-R(NT)
      WRITE(6,50) (R(NT),NT=1,NTM)
   50 FORMAT(F10.3)
C FIRST TRACE GENERATED
      DO 1000 I=2,IMAX
      KMAX=NZMAX
      IF((I*ISTEP).LT.KMAX) KMAX=(I*ISTEP)+1
      DO 20 NZ=1,KMAX
      IK=2+(NZ-1)/2
      MK=MOD(IK,ISTEP)
      M=IK/4.05
   20 C(NZ)=(MK*CD(I-M,NZ)+(ISTEP-MK)*CD(I-M+1,NZ))/ISTEP
C     THIS INTERPOLATES THE "C'S"
```

Fig. 16. Cont'd. on next page.

```
      NTM=NTMAX/ISTEP
      IF(NTM.GT.I) NTM=I
      DO 900 NT=1,NTM
      DO 101 J=1,ISTEP
  101 UP(J)=0.0
      KTMAX=KMAX
      IF(NT.EQ.1) GO TO 200
      IF(NT*4.LE.KTMAX) KTMAX=((NT-1)*ISTEP)+1
      KTMIN=KTMAX-1
      GO TO 100
  200 DOWN=D(1,1)
      DO 210 J=1,ISTEP
      UP(J)=-C(J)*DOWN
  210 DOWN=DOWN+UP(J)
      P(ISTEP+1,2)=DOWN
      GO TO 250
  100 DO 110 K=1,KTMAX
      DOWN=D(KTMAX+1-K,NT)
      DO 120 J=1,ISTEP
      IF((KTMAX-K+J).GT.NZMAX) GO TO 105
      DOWN2=DOWN*(1.0-C(KTMAX-K+J))+UP(J)*C(KTMAX-K+J)
      UP(J)=DOWN2+UP(J)-DOWN
      GO TO 120
  105 DOWN2=DOWN
  120 DOWN=DOWN2
  110 P(KTMAX+ISTEP+1-K,NT+1)=DOWN
  250 DO 130 L=1,ISTMIN
      DOWN=-UP(L)
      JMIN=L+1
      DO 140 J=JMIN,ISTEP
      DOWN2=DOWN*(1.0-C(J-L))+UP(J)*C(J-L)
      UP(J)=DOWN2+UP(J)-DOWN
  140 DOWN=DOWN2
  130 P(ISTEP+1-L,NT+1)=DOWN
      R(NT)=UP(ISTEP)
  900 P(1,NT+1)=-R(NT)
      WRITE(6,950) (R(NT),NT=1,NTM)
  950 FORMAT(5F10.3)
      DO 960 NZ=1,KTMAX
      DO 960 NT=2,NTM
  960 D(NZ,NT)=P(NZ,NT)
 1000 CONTINUE
      STOP
      END
$DATA
```

Fig. 16. Algorithm for coarse sampling along x-direction. Figure 6 explains the principles.

NT →

IX ↓

-0.640	0.000			
-0.640	-0.407	0.000		
-0.640	-0.407	-0.260	0.000	
-0.640	-0.407	-0.260	-0.169	0.000
-0.640	-0.407	-0.260	-0.169	-0.111
-0.640	-0.407	-0.260	-0.169	-0.111
-0.640	-0.407	-0.260	-0.169	-0.111
-0.640	-0.407	-0.260	-0.169	-0.111
-0.640	-0.407	-0.260	-0.169	-0.111
-0.640	-0.407	-0.260	-0.169	-0.111
-0.640	-0.407	-0.260	-0.169	-0.111
-0.640	-0.407	-0.260	-0.169	-0.111
-0.640	-0.407	-0.260	-0.169	-0.111
-0.640	-0.407	-0.260	-0.169	-0.111
-0.640	-0.407	-0.260	-0.169	-0.111
-0.640	-0.407	-0.260	-0.169	-0.111
-0.640	-0.407	-0.260	-0.169	-0.111
-0.640	-0.407	-0.260	-0.169	-0.111
-0.160	-0.101	-0.069	-0.048	-0.033
0.000	0.000	-0.003	-0.003	-0.003
0.000	0.000	-0.002	-0.002	-0.001
0.000	0.000	0.000	0.000	-0.000
0.000	0.000	0.000	0.000	-0.000
0.000	0.000	0.000	0.000	-0.000
0.000	0.000	0.000	0.000	-0.000
0.000	0.000	0.000	0.000	-0.000
0.000	0.000	0.000	0.000	-0.000
0.000	0.000	0.000	0.000	-0.000
-0.160	0.006	-0.006	0.000	-0.000
-0.640	-0.094	-0.011	-0.004	0.000
-0.640	-0.407	-0.065	-0.009	-0.002
-0.640	-0.407	-0.260	-0.047	-0.008
-0.640	-0.407	-0.260	-0.169	-0.034
-0.640	-0.407	-0.260	-0.169	-0.111
-0.640	-0.407	-0.260	-0.169	-0.111
-0.640	-0.407	-0.260	-0.169	-0.111
-0.640	-0.407	-0.260	-0.169	-0.111
-0.640	-0.407	-0.260	-0.169	-0.111
-0.640	-0.407	-0.260	-0.169	-0.111
-0.640	-0.407	-0.260	-0.169	-0.111
-0.640	-0.407	-0.260	-0.169	-0.111
-0.640	-0.407	-0.260	-0.169	-0.111
-0.640	-0.407	-0.260	-0.169	-0.111
-0.640	-0.407	-0.260	-0.169	-0.111
-0.640	-0.407	-0.260	-0.169	-0.111
-0.640	-0.407	-0.260	-0.169	-0.111
-0.640	-0.407	-0.260	-0.169	-0.111
-0.640	-0.407	-0.260	-0.169	-0.111

Fig. 16a. Received time series of Fig. 16.

```
      SUBROUTINE FMDXO(C,CF,R,D,PREMD,IMAX,NZMAX,NZM,I,UP,NGL)
      DIMENSION C(IMAX,NZMAX),R(IMAX,NZMAX),D(NZM,NZM),PREMD(NZM,NZM)
      DIMENSION CF(NZMAX)
      D(1,1)=1.0
      PREMD(1,1)=1.0
      IXC=2*NGL-1
      I=IXC
      IXC2=2*IXC
      R(I,1)=-C(I,1)*D(1,1)
      D(1,2)=-R(I,1)
      D(2,2)=R(I,1)+D(1,1)
      DO 425 NZ=1,NZMAX
      DO 425 NT=3,NZMAX
  425 D(NZ,NT)=0.0
      DO 100 I=IXC2,IMAX,IXC
      IF(I.LT.IXC*NZMAX) GO TO 430
      NTMAX=NZMAX
      GO TO 431
  430 NTMAX=I/IXC
      NTM=NTMAX+1
  431 DO 410 NZ=1,NTMAX
  410 CF(NZ)=C(I-NGL*NZ+1,NZ)
      R(I,1)=-CF(1)*D(1,1)
      PREMD(1,2)=-R(I,1)
      PREMD(2,2)=R(I,1)+D(1,1)
      DO 460 NT=2,NTMAX
      UP=-CF(NT)*D(NT,NT)
      NTPLS=NT+1
      NTMIN=NT-1
      PREMD(NTPLS,NTPLS)=UP+D(NT,NT)
      DO 420 K=1,NTMIN
      J=NT-K
      PREMD(J+1,NTPLS)=UP*CF(J)+D(J,NT)*(1.0-CF(J))
  420 UP=UP*(1.0+CF(J))-D(J,NT)*CF(J)
      R(I,NT)=UP
  460 PREMD(1,NTPLS)=-UP
      DO 470 NZ=1,NTM
      DO 470 NT=NZ,NTM
  470 D(NZ,NT)=PREMD(NZ,NT)
  100 CONTINUE
      RETURN
      END
```

Fig. 17. Algorithm for the rare situation where the c's are sampled more densely than required along the x-direction.

IX NZ

1	0	0	0	0	0	0	0	10	0	0	0	0
2	0	0	0	0	0	0	0	10	0	0	0	0
3	0	0	0	0	0	0	0	10	0	0	0	0
4	0	0	0	0	0	0	0	10	0	0	0	0
5	0	0	0	0	0	0	0	10	0	0	0	0
6	0	0	0	0	0	0	0	10	0	0	0	0
7	0	0	0	0	0	0	0	10	0	0	0	0
8	0	0	0	0	0	0	0	10	0	0	0	0
9	0	0	0	0	0	0	0	10	0	0	0	0
10	0	0	0	0	0	0	0	10	0	0	0	0
11	0	0	0	0	0	0	0	10	0	0	0	0
12	0	0	0	0	0	0	0	10	0	0	0	0
13	0	0	0	0	50	-50	0	10	0	0	0	0
14	0	0	0	0	50	-50	0	10	0	0	0	0
15	0	0	0	0	50	-50	0	10	0	0	0	0
16	0	0	0	0	50	-50	0	10	0	0	0	0
17	0	0	0	0	50	-50	0	10	0	0	0	0
18	0	0	0	0	50	-50	0	10	0	0	0	0
19	0	0	0	0	50	-50	0	10	0	0	0	0
20	0	0	0	0	50	-50	0	10	0	0	0	0
21	0	0	0	0	50	-50	0	10	0	0	0	0
22	0	0	0	0	50	-50	0	10	0	0	0	0
23	0	0	0	0	50	-50	0	10	0	0	0	0
24	0	0	0	0	50	-50	0	10	0	0	0	0
25	0	0	0	0	0	0	0	10	0	0	0	0
26	0	0	0	0	0	0	0	10	0	0	0	0
27	0	0	0	0	0	0	0	10	0	0	0	0
28	0	0	0	0	0	0	0	10	0	0	0	0
29	0	0	0	0	0	0	0	10	0	0	0	0
30	0	0	0	0	0	0	0	10	0	0	0	0
31	0	0	0	0	0	0	0	10	0	0	0	0
32	0	0	0	0	0	0	0	10	0	0	0	0
33	0	0	0	0	0	0	0	10	0	0	0	0
34	0	0	0	0	0	0	0	10	0	0	0	0
35	0	0	0	0	0	0	0	10	0	0	0	0
36	0	0	0	0	0	0	0	10	0	0	0	0
37	0	0	0	0	0	0	0	10	0	0	0	0
38	0	0	0	0	0	0	0	10	0	0	0	0
39	0	0	0	0	0	0	0	10	0	0	0	0
40	0	0	0	0	0	0	0	10	0	0	0	0
41	0	0	0	0	0	0	0	10	0	0	0	0
42	0	0	0	0	0	0	0	10	0	0	0	0
43	0	0	0	0	0	0	0	10	0	0	0	0
44	0	0	0	0	0	0	0	10	0	0	0	0
45	0	0	0	0	0	0	0	10	0	0	0	0
46	0	0	0	0	0	0	0	10	0	0	0	0
47	0	0	0	0	0	0	0	10	0	0	0	0
48	0	0	0	0	0	0	0	10	0	0	0	0
49	0	0	0	0	0	0	0	10	0	0	0	0
50	0	0	0	0	0	0	0	10	0	0	0	0
51	0	0	0	0	0	0	0	10	0	0	0	0
52	0	0	0	0	0	0	0	10	0	0	0	0

Fig. 18. Model for a simple bright spot.

IUN NT IX

IUN												
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	-10	0	0	0	0
10	0	0	0	0	0	0	0	-10	0	0	0	0
11	0	0	0	0	0	0	0	-10	0	0	0	0
12	0	0	0	0	0	0	0	-10	0	0	0	0
13	0	0	0	0	0	0	0	-10	0	0	0	0
14	0	0	0	0	0	0	0	-10	0	0	0	0
15	0	0	0	0	0	0	0	-10	0	0	0	0
16	0	0	0	0	0	0	0	-10	0	0	0	0
17	0	0	0	0	0	0	0	-10	0	0	0	0
18	0	0	0	0	-50	0	0	-15	0	0	0	0
19	0	0	0	0	-50	37	0	-7	-3	0	0	0
20	0	0	0	0	-50	37	9	-7	-1	0	0	0
21	0	0	0	0	-50	37	9	-3	-1	0	0	0
22	0	0	0	0	-50	37	9	-3	-2	0	0	0
23	0	0	0	0	-50	37	9	-3	-2	-26	0	0
24	0	0	0	0	-50	37	9	-3	-2	-26	37	0
25	0	0	0	0	-50	37	9	-3	-2	-26	37	-4
26	0	0	0	0	-50	37	9	-3	-2	-26	37	-4
27	0	0	0	0	-50	37	9	-3	-2	-26	37	-4
28	0	0	0	0	-50	37	9	-3	-2	-26	37	-4
29	0	0	0	0	-50	37	9	-3	-2	-26	37	-4
30	0	0	0	0	0	25	6	-2	-1	0	12	-6
31	0	0	0	0	0	0	0	-7	-1	0	0	0
32	0	0	0	0	0	0	0	-7	-1	0	0	0
33	0	0	0	0	0	0	0	-10	0	0	0	0
34	0	0	0	0	0	0	0	-10	0	0	0	0
35	0	0	0	0	0	0	0	-10	0	0	0	0
36	0	0	0	0	0	0	0	-10	0	0	0	0
37	0	0	0	0	0	0	0	-10	0	0	0	0
38	0	0	0	0	0	0	0	-10	0	0	0	0
39	0	0	0	0	0	0	0	-10	0	0	0	0
40	0	0	0	0	0	0	0	-10	0	0	0	0
41	0	0	0	0	0	0	0	-10	0	0	0	0
42	0	0	0	0	0	0	0	-10	0	0	0	0
43	0	0	0	0	0	0	0	-10	0	0	0	0
44	0	0	0	0	0	0	0	-10	0	0	0	0
45	0	0	0	0	0	0	0	-10	0	0	0	0
46	0	0	0	0	0	0	0	-10	0	0	0	0
47	0	0	0	0	0	0	0	-10	0	0	0	0
48	0	0	0	0	0	0	0	-10	0	0	0	0
49	0	0	0	0	0	0	0	-10	0	0	0	0
50	0	0	0	0	0	0	0	-10	0	0	0	0
51	0	0	0	0	0	0	0	-10	0	0	0	0
52	0	0	0	0	0	0	0	-10	0	0	0	0

Fig. 19. Received time series.

2		0										
3	0	0	0									
4	0	0	0	0								
5	0	0	0	0	0							
6	0	0	0	0	0	0						
7	0	0	0	0	0	0	0					
8	0	0	0	0	0	0	0	0				
9	0	0	0	0	0	0	0	10	0			
10	0	0	0	0	0	0	0	10	0	0		
11	0	0	0	0	0	0	0	10	0	0	0	
12	0	0	0	0	0	0	0	10	0	0	0	0
13	0	0	0	0	0	0	0	10	0	0	0	0
14	0	0	0	0	0	0	0	10	0	0	0	0
15	0	0	0	0	0	0	0	10	0	0	0	0
16	0	0	0	0	0	0	0	10	0	0	0	0
17	0	0	0	0	0	0	0	10	0	0	0	0
18	0	0	0	0	50	0	0	10	0	0	0	0
19	0	0	0	0	50	-50	0	10	0	0	0	0
20	0	0	0	0	50	-50	0	10	0	0	0	0
21	0	0	0	0	50	-50	0	10	0	0	0	0
22	0	0	0	0	50	-50	0	10	0	0	0	0
23	0	0	0	0	50	-50	0	10	0	0	0	0
24	0	0	0	0	50	-50	0	10	0	0	0	0
25	0	0	0	0	50	-50	0	10	0	0	0	0
26	0	0	0	0	50	-50	0	10	0	0	0	0
27	0	0	0	0	50	-50	0	10	0	0	0	0
28	0	0	0	0	50	-50	0	10	0	0	0	0
29	0	0	0	0	50	-50	0	10	0	0	0	0
30	0	0	0	0	0	-50	0	10	0	0	0	0
31	0	0	0	0	0	0	0	10	0	0	0	0
32	0	0	0	0	0	0	0	10	0	0	0	0
33	0	0	0	0	0	0	0	10	0	0	0	0
34	0	0	0	0	0	0	0	10	0	0	0	0
35	0	0	0	0	0	0	0	10	0	0	0	0
36	0	0	0	0	0	0	0	10	0	0	0	0
37	0	0	0	0	0	0	0	10	0	0	0	0
38	0	0	0	0	0	0	0	10	0	0	0	0
39	0	0	0	0	0	0	0	10	0	0	0	0
40	0	0	0	0	0	0	0	10	0	0	0	0
41	0	0	0	0	0	0	0	10	0	0	0	0
42	0	0	0	0	0	0	0	10	0	0	0	0
43	0	0	0	0	0	0	0	10	0	0	0	0
44	0	0	0	0	0	0	0	10	0	0	0	0
45	0	0	0	0	0	0	0	10	0	0	0	0

Fig. 20. Reconstructed model.