X OUTER MIGRATION

W. Scott Dunbar and Jon F. Claerbout

With the discovery of an explicit difference scheme for the equation [1]

$$P_{zt} = \frac{v}{2} P_{xx} \tag{1}$$

it became possible to develop a migration algorithm with the following properties:

(1) an outer DO loop on the number of traces

(2) no arrays depending on the number of traces

(3) uses only three traces for migration (diffraction) at a time since P_{xx} involves only three numbers.

The following derivation will largely follow [1] but will be consistent with the program in the Appendix. We define a grid by $(X-1)\Delta X \quad 1 \leq X \leq NX$, $(T-1)\Delta T \quad 1 \leq T \leq NT$ and $(Z-1)\Delta Z \quad 1 \leq Z \leq NZ$.

In equation 11.3.19 in the last chapter of Claerbout's book (included in this report) is the equation

$$Q_{t't''} = -(v^2/8)\, Q_{yy} \tag{2}$$

where y is midpoint coordinate, t' is the two way travel time coordinate, t'' is the depth coordinate scaled to two way travel time and Q is the seismic section. In order to use the difference equation

$$\delta_{t''}\, \delta_{t'}\, P = -4a \quad \delta_{xx} \quad P \tag{3}$$

make the interpretation that

$$4a = \frac{v^2 \Delta t' \Delta t''}{8 (\Delta y)^2}$$

From this we deduce that

$$\frac{NT}{NZ} = \frac{\Delta t''}{\Delta t'} = 32a \left(\frac{\Delta y}{v \Delta t'} \right)^2$$

In the program $F = 2a$ and $a = 1/8$. Stability limits $a < 1/4$. For most accurate horizontal space derivative calculation a should be $1/12$. However for an accurate vertical derivative it is important that $\Delta t''$ $(=\Delta z)$ should be no larger than an eigth of a wavelength. Thus, (particularly if Δy is large and the data has steep dips) it may be necessary to take a less than $1/12$. The computational star for the implicit scheme for the equation $P_{zt} = a P_{xx}$ with $T = -\delta_{xx}$, see [2], is

$$\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} + a \begin{bmatrix} T & T \\ T & T \end{bmatrix}$$

For the explicit scheme it is

$$\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} + a \begin{bmatrix} 0 & 2T \\ 2T & 0 \end{bmatrix}$$

For the equation $P_{zt} = - a P_{xx}$ it is

$$\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} + a \begin{bmatrix} 2T & 0 \\ 0 & 2T \end{bmatrix}$$

which is equivalent to

$$- \begin{bmatrix} (1+2aT) & -1 \\ -1 & 1+2aT \end{bmatrix}$$

which implies the computing scheme

$$P(Z,TR,X) = - P(Z-1,TR+1,X) + (P(Z,TR+1,X) + P(Z-1,TR,X)) + F*T(P(Z,TR+1,X) + P(Z-1,TR,X)) \qquad (4)$$

where TR = NT-1, NT-2, ..., 1 .

Figure 1 shows the order of computation used for (4). The template at the bottom of Figure 2 may be used to verify the order of calculation.

In the interest of economy, we want to read the tape serially. This cannot be done with the computational order in Figure 1. To circumvent this difficulty (and introduce another problem, as we shall see), we transform or skew the data according to

$$X' = X + NT - T + Z - 1 \qquad (5)$$

$$1 \leqslant T \leqslant NT \qquad 1 \leqslant Z \leqslant NZ$$

The reader may see by using (5) and Figure 1 that this gives the computational scheme shown in Figure 3. The appropriate template is shown in Figure 4. By inspection of Figures 1 and 3 and using equation (5), the numbers required for equation (4) obey the skewed equation

$$P(Z, T, X+2) = P(Z, T-1, X+1) + P(Z-1, T, X+1) - P(Z-1, T-1, X) + 2a\,T[\,P(Z, T-1, X+1) + P(Z-1, T, X+1)\,] \qquad (6)$$

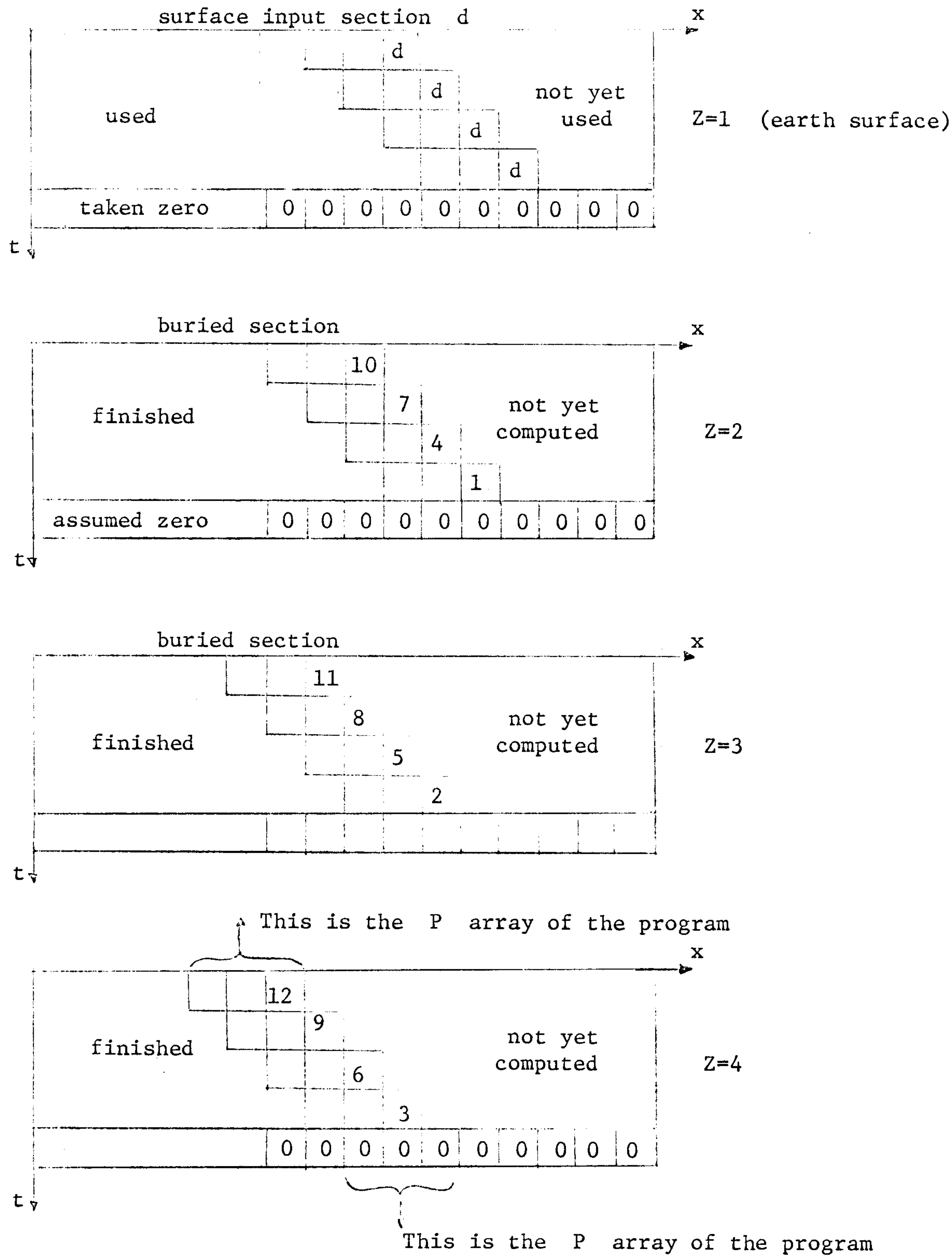
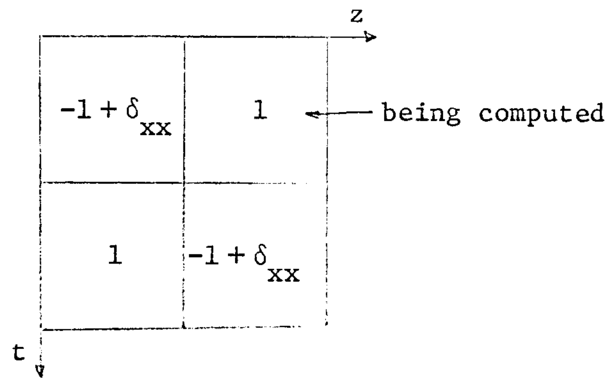


Figure 1. The numbers in the squares indicate the order in which the downward continuation is done. To verify that a cell is based on previously computed cells, cut out and overlay the template at the bottom of Figure 2.



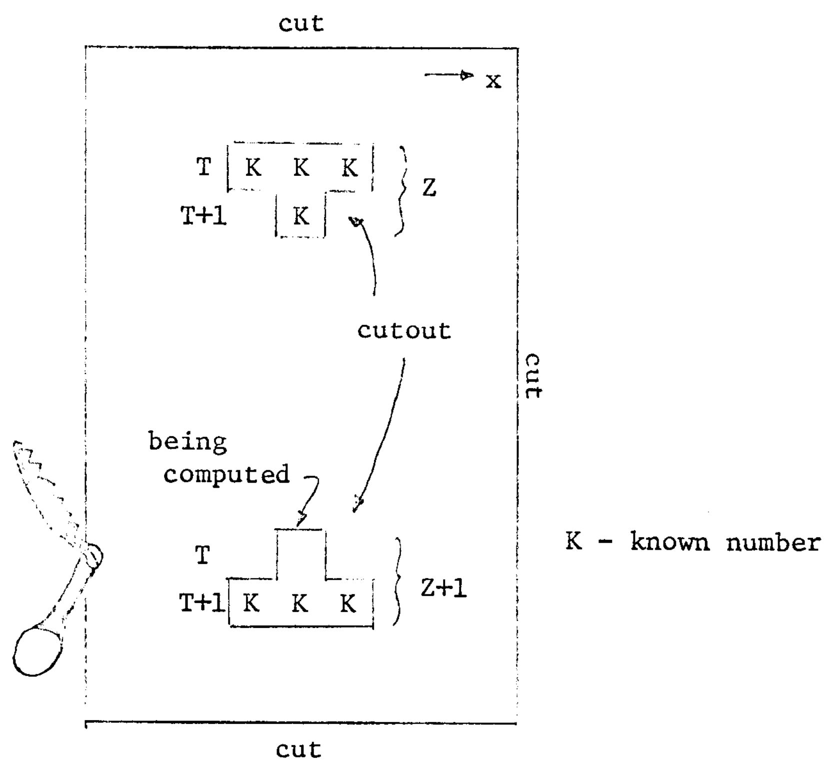


Figure 2. The top diagram shows the computational star in the Z-T plane for the explicit calculation of $P_{zt} = - v/2\ P_{xx}$. See "An Explicit Scheme for $P_{zt} = \frac{v}{2} P_{xx}$" (this report) and p. 73, SEP March 1974. The bottom diagram is a template for the same star for use in Figure 1.

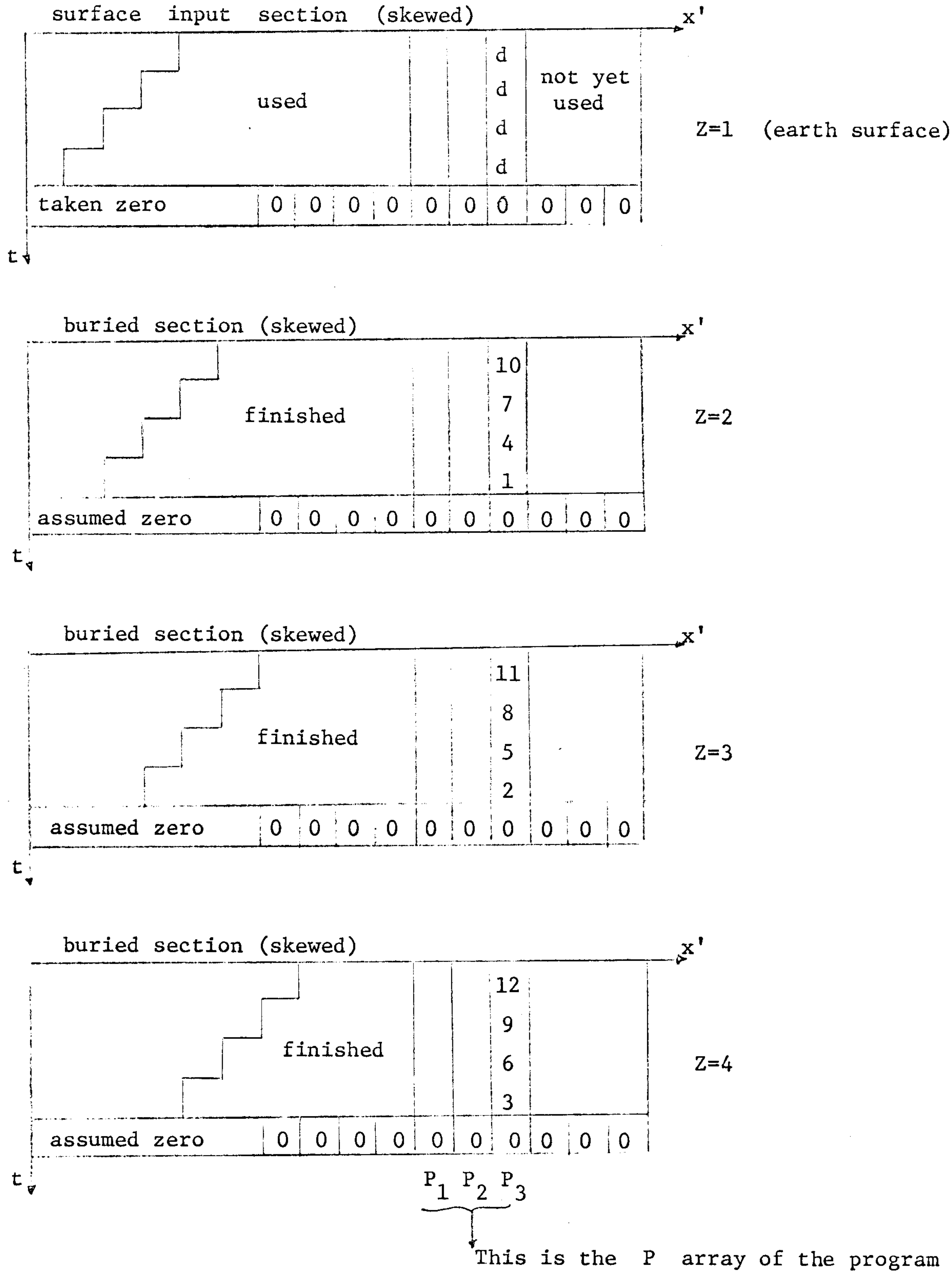


Figure 3. The order of computation is the same as that of Figure 1. Each section has been skewed in the x direction according to $X' = X + NT - T + Z - 1$. Only three skewed traces are needed. The template of Figure 4 can be used to verify this.

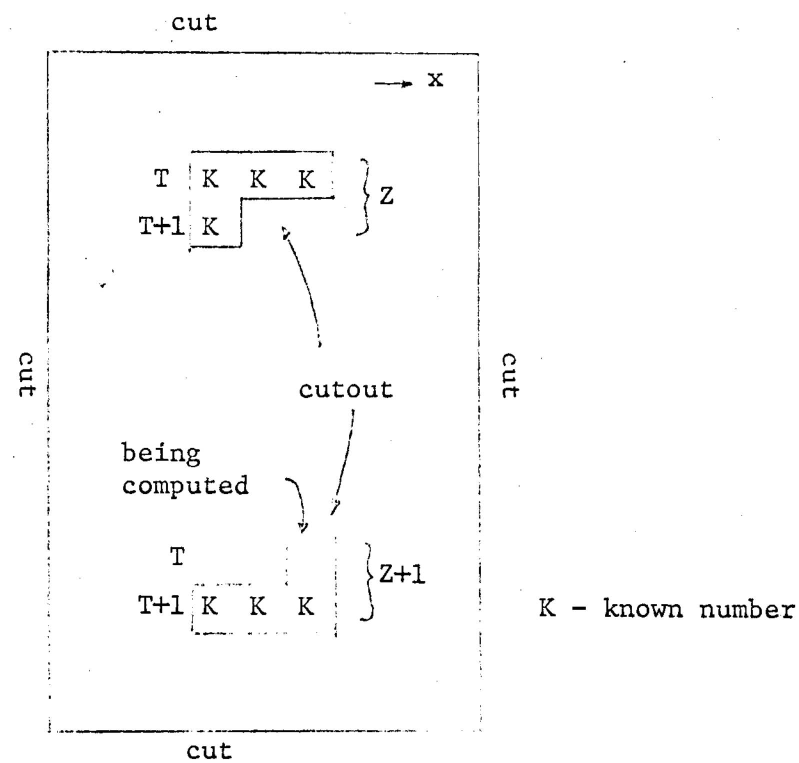


Figure 4. The skewed-operator of Figure 2. For use as a template on Figure 3.

Since the computations are now ordered so that only three traces are used, the P array of the program need only be dimensioned P(NZ, NT, 3) . X in (6) always equals 1 . When one migration step is finished, the trace corresponding to X=1 is not needed and can be used for storing the next trace, i.e., the X subscript of P is rotated.

Now, how do we do the skewing? By looking at Figures 1 and 3, it can be seen that we are going to need storage equal to NT x NT no matter how the skewing is done. This is the major practical problem of the algorithm.

In the program, the skewing is done by using NT x NT array called SKEW . A pointer array, IT , is used to designate specific columns of SKEW . SKEW and P are initialized to zero. IT is initialized to 1, 2, ..., NT . Integer variables X1, X2 and X3 are initialized to 1, 2 and 3 respectively. The first trace is read from the original tape and placed in the IT(NT)th column of SKEW. The diagonal of SKEW is then put into P(1, T, X3 = 3) . The first migration step takes place for all T and Z , using the dead traces P(NZ, NT, 1) and P(NZ, NT, 2) .

After migration, the migrated data is placed on the diagonal of SKEW . Figure 5 displays the entire operation. The unskewed migrated traces are in the IT(1)th column of SKEW . However, there is a time lag of NT before any complete traces can be plotted, i.e., it takes NT steps of X in the skewed frame before complete unskewed traces emerge from the IT(1)th column of SKEW . To complete the traces consecutively, the array IT is rotated according to

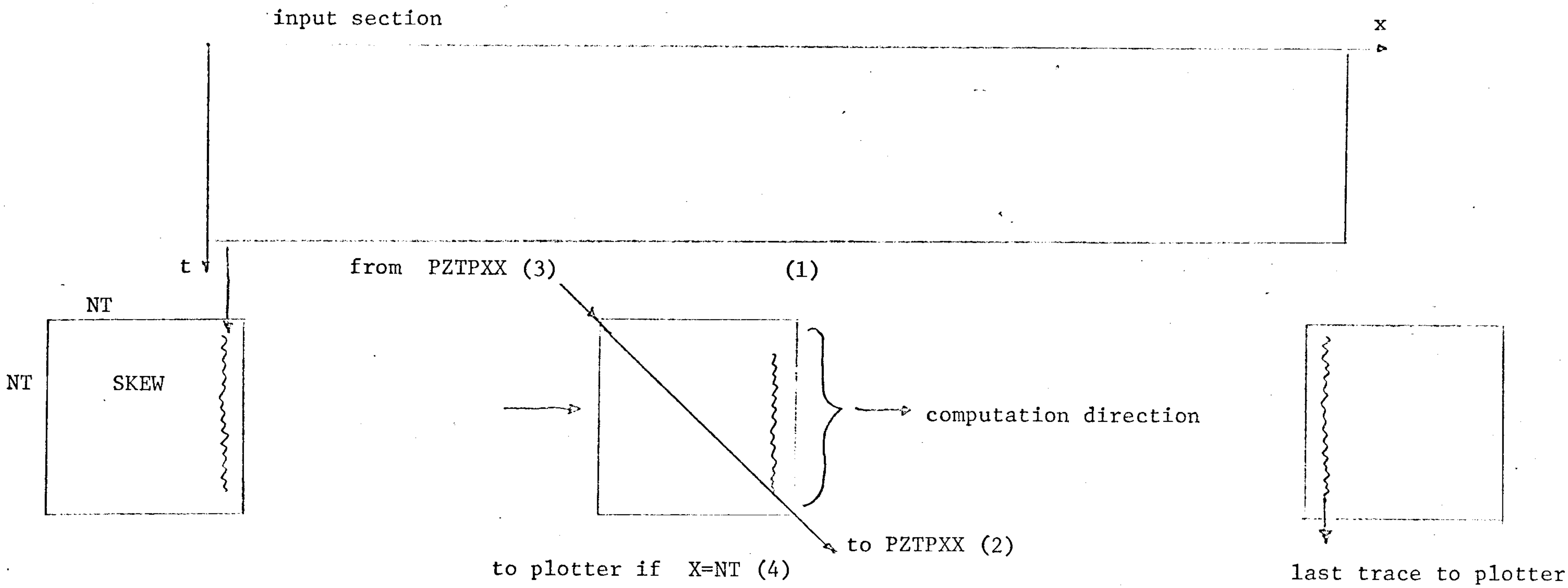


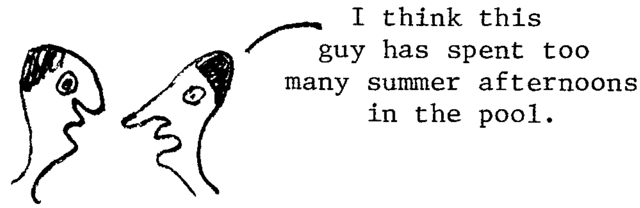
Figure 5. The operation of the array SKEW . Numbers indicate order of operation.

$$IT(T-1) = IT(T) \qquad 2 \leqslant T \leqslant NT$$

$$IT(NT) = IT(1)$$

X1, X2 and X3 are also rotated. Thus SKEW is like a cylindrical rotating machine passing across the section accepting original data from the tape, processing it and spitting out migrated traces at the other end.

Within the migration subroutine PZTPXX , the array P is decomposed into three arrays P1, P2 and P3 corresponding to X1, X2 and X3 respectively. The migration is done out to the diagonal of the Z-T plane (see [2]) and the numbers along the diagonal are shifted to the NZth position of P3 . This makes it easier to reference the skewed migrated trace after exiting PZTPXX .



A listing of the program is shown overleaf. The comments should explain the order of the operations and the arrays are those of the text. Tape read and write routines would have to be inserted in the appropriate places. S is a temporary storage array which may not be necessary, depending on the plot mode (real or integer). If the plot mode is integer, the array BUFFER could be used instead.

Because of the time lag for completed traces, the outer X loop actually has the limit NX + NT . IF statements within the loop control the reading and writing of tapes. Migration still continues when $X > NX$.

The program is currently set up for 600 traces, 5 sec of 4 ms data and a "magic number" $a = 1/8$.

```
      INTEGER*2 BUFFER(1550),SKEW(1250,1250)
      INTEGER T,Z,X1,X2,X3,XH,IT(1250)
      REAL*4 S(1250),P(22,1250,3)
C
C BUFFER IS FOR TAPE READS,S IS A TEMPORARY
C STORAGE ARRAY WHICH MAY NOT BE NECESSARY
C
      NX=600
      NT=1250
      NTBUF=1550
      NZ=22
      NXTOT=NX+NT
C
C INITIALIZE  ARRAYS
C
      DO 10 X=1,3
      DO 10 T=1,NT
      DO 10 Z=1,NZ
   10 P(Z,T,X)=0.
      DO 15 J=1,NT
      DO 15 I=1,NT
   15 SKEW(I,J)=0
      DO 20 T=1,NT
   20 IT(T)=T
      X1=1
      X2=2
      X3=3
C
C MIGRATE THE DATA
C
      DO 65 X=1,NXTOT
C
C READ A TRACE IF ONE IS AVAILABLE
C
      DO 25 T=1,NTBUF
   25 BUFFER(T)=0
      IF (X.GT.NX) GO TO 30
C
C ***** TAPE READ ROUTINE IS USED HERE *****
C
C USE ALL OR PART OF TRACE (DECIMATE,IF NECESSARY)
C
   30 DO 35 T=1,NT
   35 S(T)=BUFFER(T)
C
C PUT S ON IT(NT)TH COLUMN OF SKEW,EXTRACT
C DIAGONAL OF SKEW AND MIGRATE
C
      ITT=IT(NT)
      DO 40 T=1,NT
      SKEW(T,ITT)=S(T)
   40 P(1,T,X3)=SKEW(T,IT(T))
      CALL PZTPXX (P(1,1,X1),P(1,1,X2),P(1,1,X3),NT,NZ)
C
C PUT MIGRATED SKEWED TRACE ON DIAGONAL OF SKEW
C
      DO 45 T=1,NT
   45 SKEW(T,IT(T))=P(NZ,T,X3)
C
```

```
C WRITE IT(1)TH CCLUMN CF SKEW TO PLOT TAPE IF
C RCTATION OF IT(T) HAS COMPLETED ONE CYCLE
C
      IF (X.LT.NT) GO TO 55
      ITT=IT(1)
      DO 50 T=1,NT
   50 S(T)=SKEW(T,ITT)
C
C***** TAPE WRITE RCUTINE IS USED HERE *****
C
C RCTATE IT(T)
C
   55 ITH=IT(1)
      DO 60 T=2,NT
   60 IT(T-1)=IT(T)
      IT(NT)=ITH
C
C ROTATE X1,X2,X3
C
      XH=X1
      X1=X2
      X2=X3
      X3=XH
   65 CONTINUE
      STOP
      END
      SUBROUTINE PZTPXX (P1,P2,P3,NT,NZ)
      REAL*4 P1(NZ,NT),P2(NZ,NT),P3(NZ,NT)
      INTEGER T,Z,TR
      DATA F/0.25/
      DC 30 T=2,NT
      TR=NT-T+1
      NZDIAG=2+TR*(NZ-1.)/NT
      DO 10 Z=2,NZDIAG
      P3(Z,TR)=P2(Z,TR+1)+P2(Z-1,TR)-P1(Z-1,TR+1)
     $             +F*(P1(Z,TR+1)+P1(Z-1,TR)
     $             -2.*(P2(Z,TR+1)+P2(Z-1,TR))
     $             +P3(Z,TR+1)+P3(Z-1,TR))
10    CONTINUE
      NZDP1=NZDIAG+1
      IF(NZDP1.GT.NZ) GO TO 30
      DO 20 Z=NZDP1,NZ
20    P3(Z,TR)=P2(Z-1,TR)
30    CONTINUE
      RETURN
      END
```

References

[1] J. F. Claerbout, "An Explicit Scheme for $P_{zt} = \frac{v}{2} P_{xx}$", this report.

[2] J. F. Claerbout, "Two Techniques for Wave Equation Migration, SEP March 1974, p. 73.