Benchmark Program for Migration/Diffraction

Don C. Riley

The benchmark program included in this section was written in order to compare various types of computers. As such, it contains a representative mix of the type of operations required for partial differential equations applications, and to insure compatability was written in basic FORTRAN. The program also serves well to illustrate some of the techniques used in programming finite difference solutions.

The basic continuation equation for extrapolating wave fields propagating at small angles from the +Z direction is

$$D_{zt} = c/2 \; D_{xx} \tag{1}$$

In obtaining finite difference solutions to equation (1) we have a wide variety of differencing schemes to choose from. The mixed method of Crank and Nicolson overcomes the fine-grid restrictions of explicit schemes in addition to smaller truncation errors than wholly implicit schemes. Let us discretize the coordinates as follows: $x = j\Delta x$, $z = k\Delta z$, $t = n\Delta t$ in which case the approximation to D(x,z,t) will be denoted by $D^n_{k,j}$.

The Crank-Nicolson scheme is

$$\frac{D^n_{k,j} - D^{n-1}_{k,j} - D^n_{k-1,j} + D^{n-1}_{k-1,j}}{(\Delta z)(\Delta t)} = c/2 \; \frac{\delta_x^2 [D^n_{k,j} + D^{n-1}_{k,j} + D^n_{k-1,j} + D^{n-1}_{k-1,j}]}{4} \tag{2}$$

where the central-difference operator is defined by

$$\delta_x^2 \; [D^n_{k,j}] = \frac{D^n_{k,j+1} - 2D^n_{k,j} + D^n_{k,j-1}}{(\Delta x)^2}$$

Introducing the notation $\underline{D}_k^n$ for a vector with the elements

$$D_{k,1}^n, D_{k,2}^n, \ldots, D_{k,N}^n$$

in which case we may represent the second space differencing as

$$\delta_x^2 [\underline{D}_k^n] = \frac{-T}{(\Delta x)^2} \underline{D}_k^n$$

where T is the tridiagonal matrix of dimension NxN

$$T = \begin{bmatrix} 2 & -1 & 0 & \ldots & 0 \\ -1 & 2 & -1 & \ldots & 0 \\ \vdots & & & & \\ 0 & 0 & \ldots & -1 & 2 \end{bmatrix}$$

Equation (2) written in matrix notation is

$$\underline{D}_k^n - \underline{D}_k^{n-1} - \underline{D}_{k-1}^n + \underline{D}_{k-1}^{n-1} = -\frac{c\Delta z\Delta t}{8(\Delta x)^2} T[\underline{D}_k^n + \underline{D}_k^{n-1} + \underline{D}_{k-1}^n + \underline{D}_{k-1}^{n-1}] \quad (3)$$

letting $a = \frac{c\Delta z\Delta t}{8(\Delta x)^2}$ and rearranging (3)

$$[I + aT]\underline{D}_k^n = [I - aT](\underline{D}_k^{n-1} + \underline{D}_{k-1}^n) - [I + aT]\,\underline{D}_{k-1}^{n-1} \quad (4)$$

Equation (4) may be used to extrapolate from $z_1 = (k-1)\Delta z$ to $z_2 = k\Delta z$ knowing the wave field $D(x,t)$ at z_1. To start the recursion in time (n) we usually assume that $\underline{D}_k^n = 0$ for $n < 0$, i.e. the first arrival of the wave field is represented on the finite grid.

In programming equation (4) the simplest way is to have two separate grids, one for $\underline{D}_{k-1}^n$ $n = 0,1,2\ldots$(old) and one for $\underline{D}_k^n$ $n = 0,1,2\ldots$(new).

These grids, the old and new would be swapped back and forth in extrapolating the waves over many steps in z. In an effort to reduce the amount of core we might examine the algorithm to see where we might overlay some of the storage. It is somewhat less confusing to consider the following integrating form of the algorithm for diffracting or migrating wave fields "in place", i.e. using only one 2-D grid.

Let us express the time differencing in z-transform notation $\partial_t \approx \frac{2}{\Delta t}\frac{1-z}{1+z}$ and equation (3) may be rewritten as

$$(1-z)[\underline{D}_k - \underline{D}_{k-1}] = -aT(1+z)[\underline{D}_k + \underline{D}_{k-1}]$$

$$\underline{D}_k - \underline{D}_{k-1} = -aT\frac{(1+z)}{(1-z)}[\underline{D}_k + \underline{D}_{k-1}] \qquad (5)$$

Let us expand $(1+z)/(1-z)$ in terms of an infinite series

$$\frac{(1+z)}{(1-z)} = 1 + 2z + 2z^2 + 2z^3 + \dots$$

and inserting this into (5)

$$\underline{D}_k^n - \underline{D}_{k-1}^n = -aT[\underline{D}_k^n + \underline{D}_{k-1}^n] - 2aT[\underline{D}_k^{n-1} + \underline{D}_{k-1}^{n-1} + \underline{D}_k^{n-2} + \underline{D}_{k-1}^{n-2} + \dots]$$

$$[I + aT]\underline{D}_k^n = [I - aT]\underline{D}_{k-1}^n - 2aT\sum_{i=1}^{\infty}(\underline{D}_k^{n-i} + \underline{D}_{k-1}^{n-i}) \qquad (6)$$

Equation (6) is the integrating form which allows us to use a single grid for projecting the wave field over successive k steps.

The algorithm is

$$\underline{D}^n \leftarrow \text{Initial wave field } D(x,0,t)$$

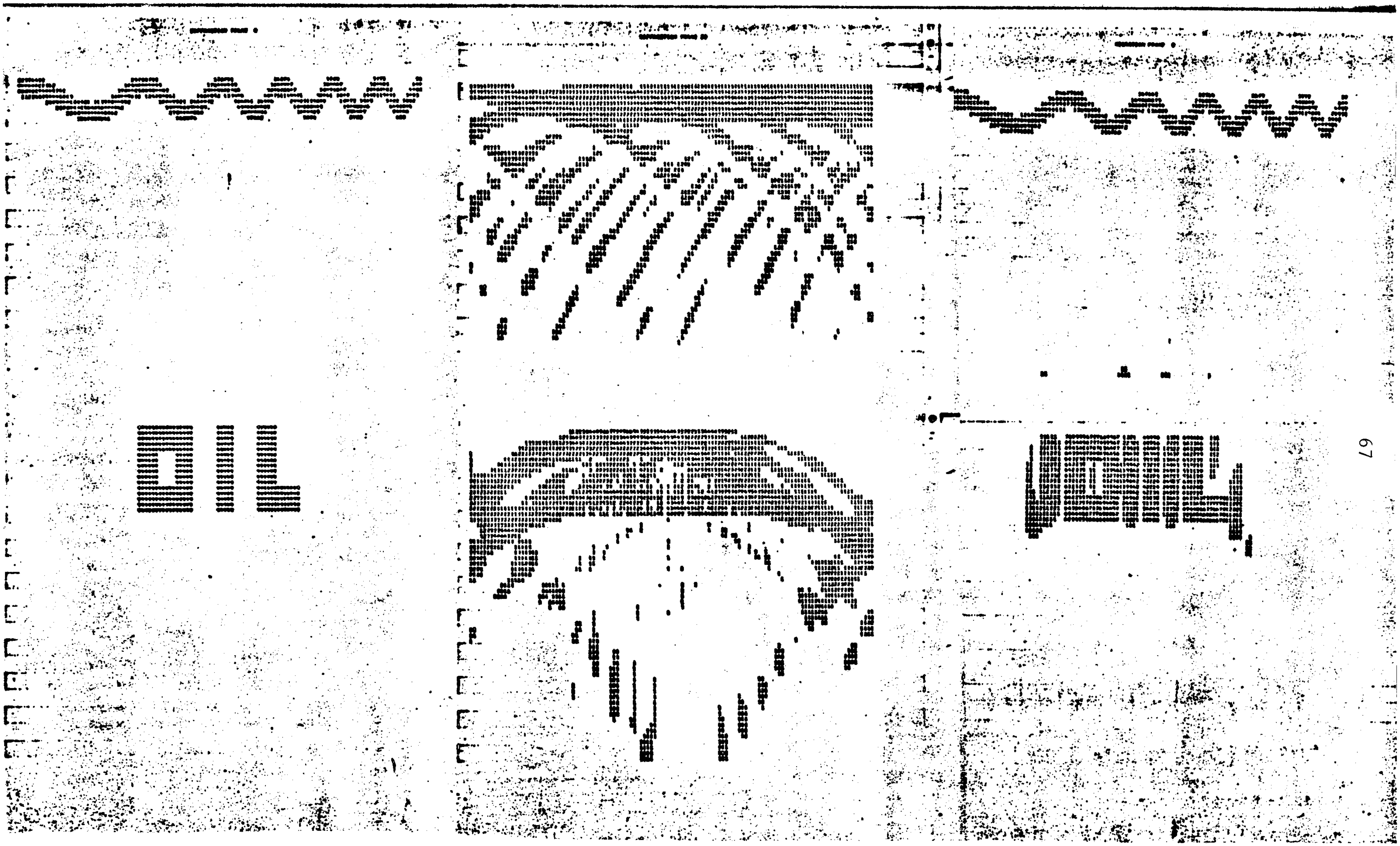
$$k = 1,2,\ldots \quad \left\{ \begin{array}{l} \underline{u} \leftarrow (0,0,\ldots,0) \quad \text{assume } \underline{D}_k^n = 0 \quad n \le 0 \\ n = 1,2,\ldots \quad \left\{ \begin{array}{l} \underline{s} \leftarrow \underline{D}^n \\ \underline{f} \leftarrow \underline{s} - T(a\underline{s} + 2a\underline{u}) \\ (I + aT)\,\underline{D}^n = \underline{f} \\ \underline{u} \leftarrow \underline{u} + (\underline{D}^n + \underline{s}) \end{array} \right. \end{array} \right.$$

The following computer program takes an initial wave field and outward continues (diffracts) it through 20 steps in the +z direction. At this point the wave field is inward continued (migrated) through 20 steps in the -z direction. The final reconstructed frame is to be compared with the initial frame.

```
C       A BENCHMARK PROGRAM
C       FOR PARTIAL DIFFERENTIAL EQUATIONS APPLICATIONS
C
C       FOR A COMPARISON TO YOUR COMPUTER CENTER FILL IN THE
C       BLANKS BELOW AND RETURN TO :
C                JON CLAERBOUT
C                GEOPHYSICS DEPARTMENT
C                STANFORD, CAL. 94305
C
C       YOUR NAME
C             INSTITUTION
C             ADDRESS
C             COMUTER BRAND AND MODEL
C             RUN TIME INCLUDING COMPILE
C             JOB COST EXCLUDING OVERHEAD
C             JOB COST INCLUDING OVERHEAD
C             PRIORITY (DAY,NIGHT,HIGH,LOW,ETC.)
C             DATE AND TIME
C             TURNAROUND TIME
C
C      ON AN IBM 360/67 THIS JOB TAKES LESS THAN FIVE MINUTES
C      AND REQUIRES LESS THAN 250,000 BYTES STORAGE
C
      DIMENSION WAVE(240,240)
      RICKY(T)=(1.-2.*AR*T*T)*EXP(-AR*T*T)
      CHIRP(I)=10.*.5*(1.-COS((1.+2.*(I-1)/(NX-1))*
     1         6.283*2.0*(I-1)/(NX-1)))+20.
      AR=8./17./17.
      A=1.00
      LS=9
      NX=240
      NT=240
      NZ=20
      DO 1000 J=1,NT
      DO 1000 I=1,NX
      IF(I-120) 330,301,301
  301 IF(I-150) 302,302,330
  302 IF(J-80)  330,303,303
  303 IF(J-154) 304,304,330
  304 IF(J-106) 305,305,309
  305 IF(I-128) 340,340,306
  306 IF(I-142) 307,340,340
  307 IF(J-88)  340,340,308
  308 IF(J-98)  330,340,340
  309 IF(J-124) 310,310,311
  310 IF(J-116) 311,340,340
  311 IF(J-134) 330,312,312
  312 IF(J-142) 340,340,313
  313 IF(I-142) 330,340,340
  340 WAVE(J,I)=0.75
      GO TO 1000
  330 WAVE(J,I)=0.0
 1000 CONTINUE
      DO 2000 I=1,NX
```

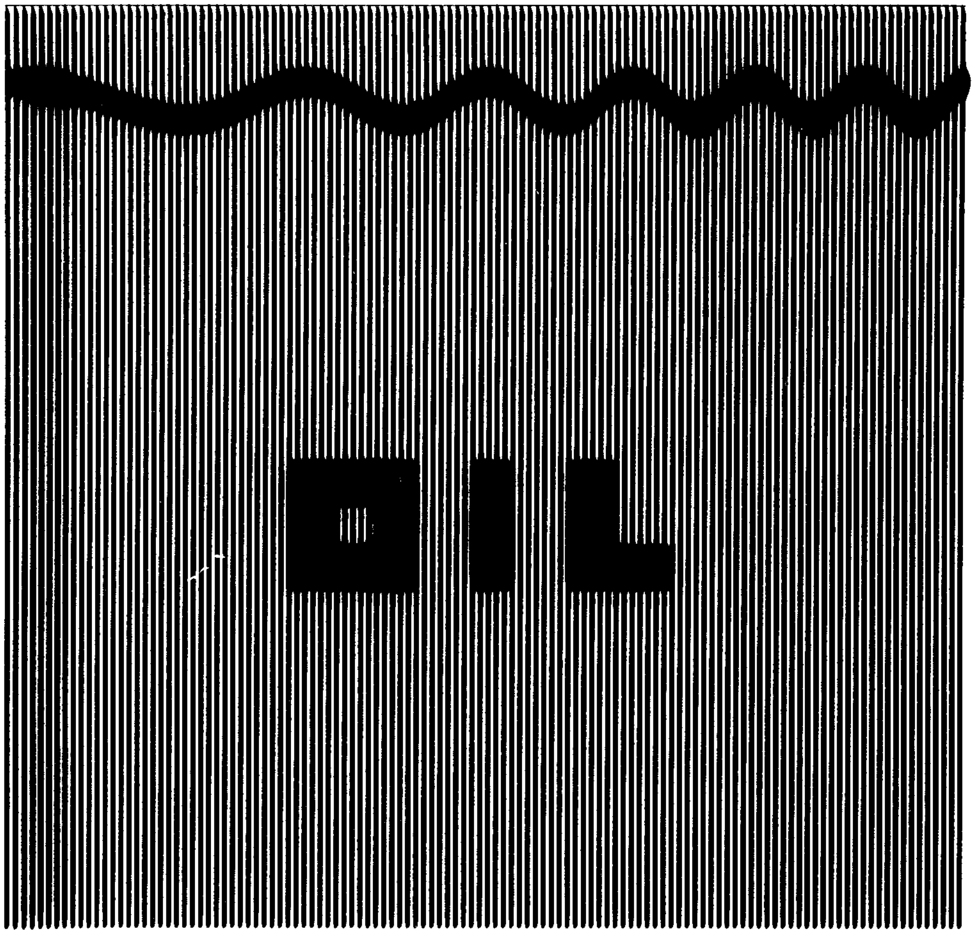
```
      D=CHIRP(I)
      IST=D-LS/2+.499
      IEN=IST+LS-1
      DO 2000 J=IST,IEN
 2000 WAVE(I,J)=RICKY(D-J)
      K=0
      PRINT 100,K
      CALL OUT(WAVE,NX,NT)
      DO 4000 K=1,NZ
      CALL FAST15(WAVE,NX,NT,A,+1)
      IF(MOD(K,5).NE.0) GO TO 4000
      PRINT 100,K
      CALL OUT(WAVE,NX,NT)
 4000 CONTINUE
      DO 5000 K=1,NZ
      N=NZ-K
      CALL FAST15(WAVE,NX,NT,A,-1)
      IF(MOD(N,5).NE.0) GO TO 5000
      PRINT 200,N
      CALL OUT(WAVE,NX,NT)
 5000 CONTINUE
      STOP
  100 FORMAT(1H1,50X,18HDIFFRACTION FRAME ,I2)
  200 FORMAT(1H1,50X,16HMIGRATION FRAME ,I2)
      END
      SUBROUTINE FAST15(WAVE,NX,NT,A,MODE)
      DIMENSION WAVE(NX,NT),S(240),U(240),V(240),E(240),F(240)
      NXM1=NX-1
      A2=2.*A
      ADIAG=1.+A2
      AOFF=-A
      NBASE=0
      IF(MODE.EQ.-1) NBASE=NT+1
      DO 1000 I=1,NX
 1000 U(I)=0.
      DO 4000 JT=1,NT
      J=NBASE+MODE*JT
      DO 2000 I=1,NX
      S(I)=WAVE(I,J)
 2000 E(I)=A*S(I)+A2*U(I)
      DO 3000 I=2,NXM1
 3000 V(I)=S(I)+E(I-1)+E(I+1)-E(I)-E(I)
      CALL TRI(AOFF,ADIAG,AOFF,NX,WAVE(1,J),V,E,F)
      DO 4000 I=1,NX
 4000 U(I)=U(I)+S(I)+WAVE(I,J)
      RETURN
      END
      SUBROUTINE TRI(A,B,C,N,T,D,E,F)
      DIMENSION T(N),D(N),F(N),E(N)
      N1=N-1
      E(1)=1.
      F(1)=0.
      DO 1000 I=2,N1
      DEN=B+C*E(I-1)
```

```
      E(I)=-A/DEN
 1000 F(I)=(D(I)-C*F(I-1))/DEN
      T(N)=F(N1)/(1.0-E(N1))
      DO 2000 J=1,N1
      I=N-J
 2000 T(I)=E(I)*T(I+1)+F(I)
      RETURN
      END
      SUBROUTINE OUT(WAVE,NX,NT)
      DIMENSION WAVE(NX,NT),ICHAR(21),LINE(120)
      DATA ICHAR/1HH,1HG,1HF,1HE,1HD,1HC,1HB,1HA,1H ,1H ,1H1,1H2,1H3,
     1           1H4,1H5,1H6,1H7,1H8,1H9,1H0,1H*/
      NX2=NX/2
      B=0.
      DO 1000 J=2,NT,2
      DO 1000 I=2,NX,2
      T=ABS(WAVE(I,J))
      IF(T.GT.B) B=T
 1000 CONTINUE
      DO 3000 J=2,NT,2
      DO 2000 I=1,NX2
      IVAL=10.+WAVE(2*I,J)*12./B
 2000 LINE(I)=ICHAR(MIN0(21,MAX0(09,IVAL)))
 3000 PRINT 100,(LINE(I),I=1,120)
  100 FORMAT(1H ,120A1)
      RETURN
      END
```

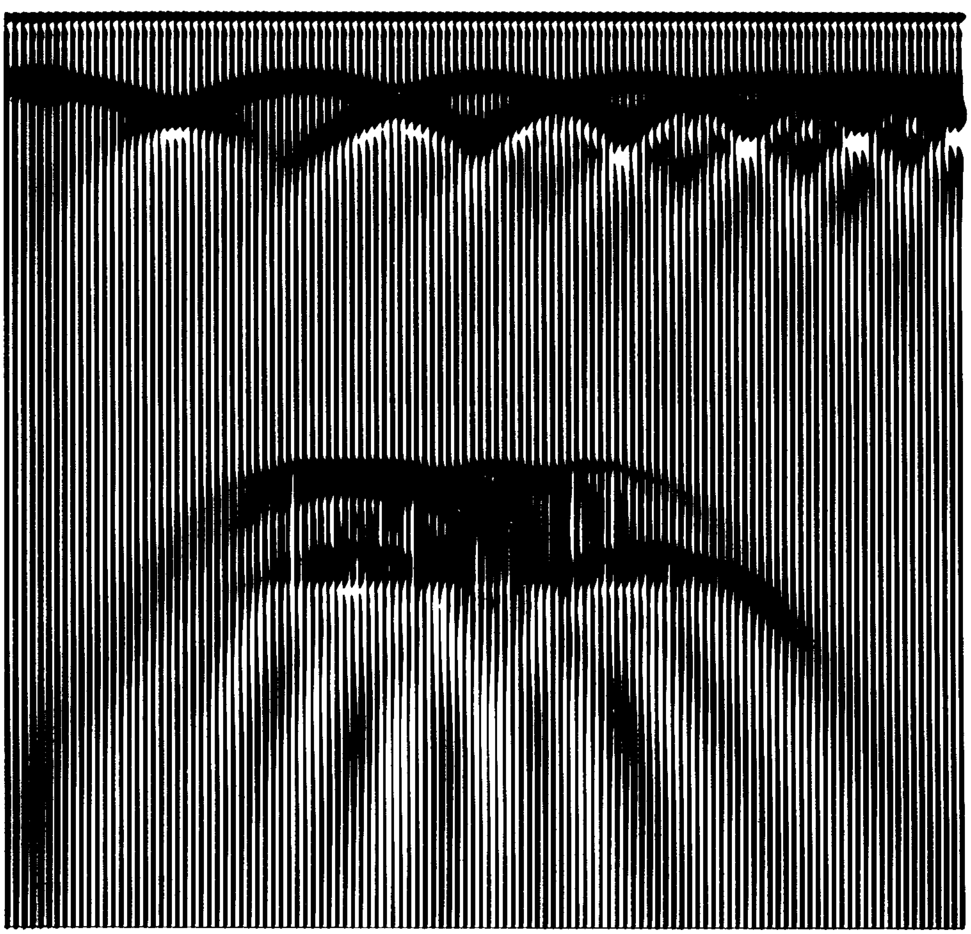


This is an example of the printer-plot output from the benchmark program. The frames are (from left to right) the initial wavefield, the waves projected outward (diffracted) 20 Δz steps, and attempted reconstruction of the original wavefield after migration back through 20 Δz steps.

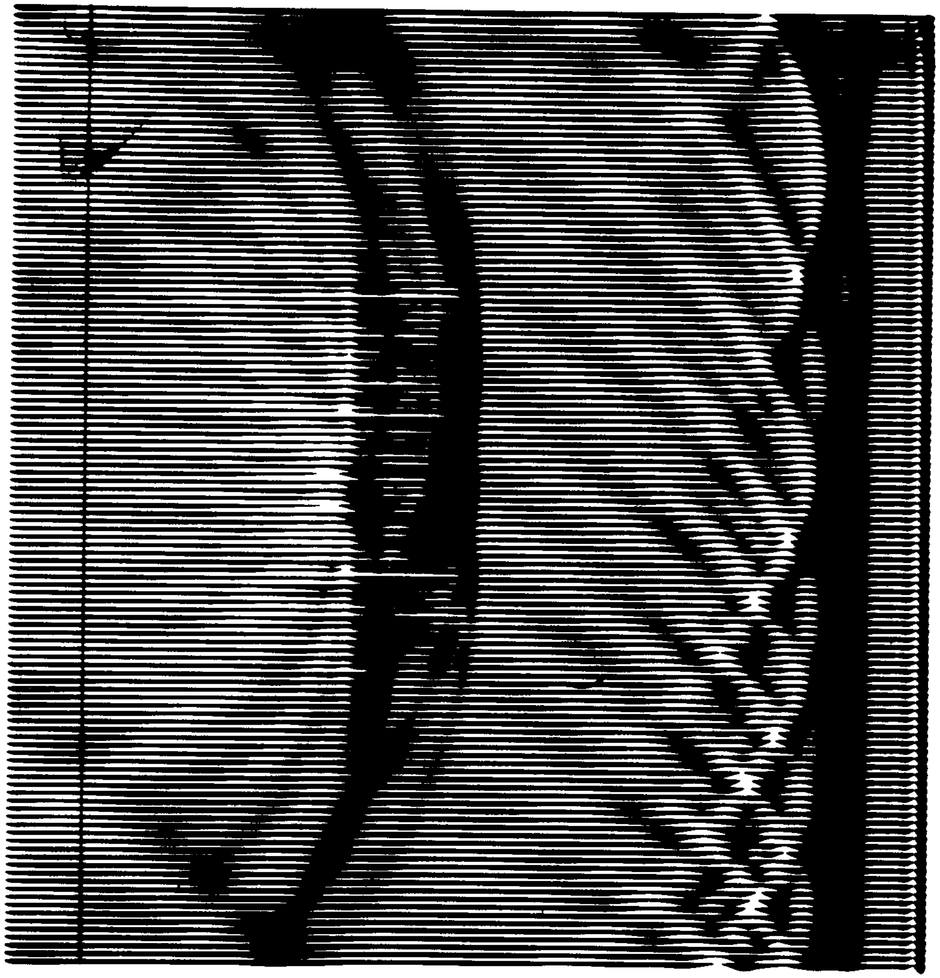
The following plots are the same output displayed on a seismic section plotter. The intermediate frames are displayed in increments of 5 Δz steps.



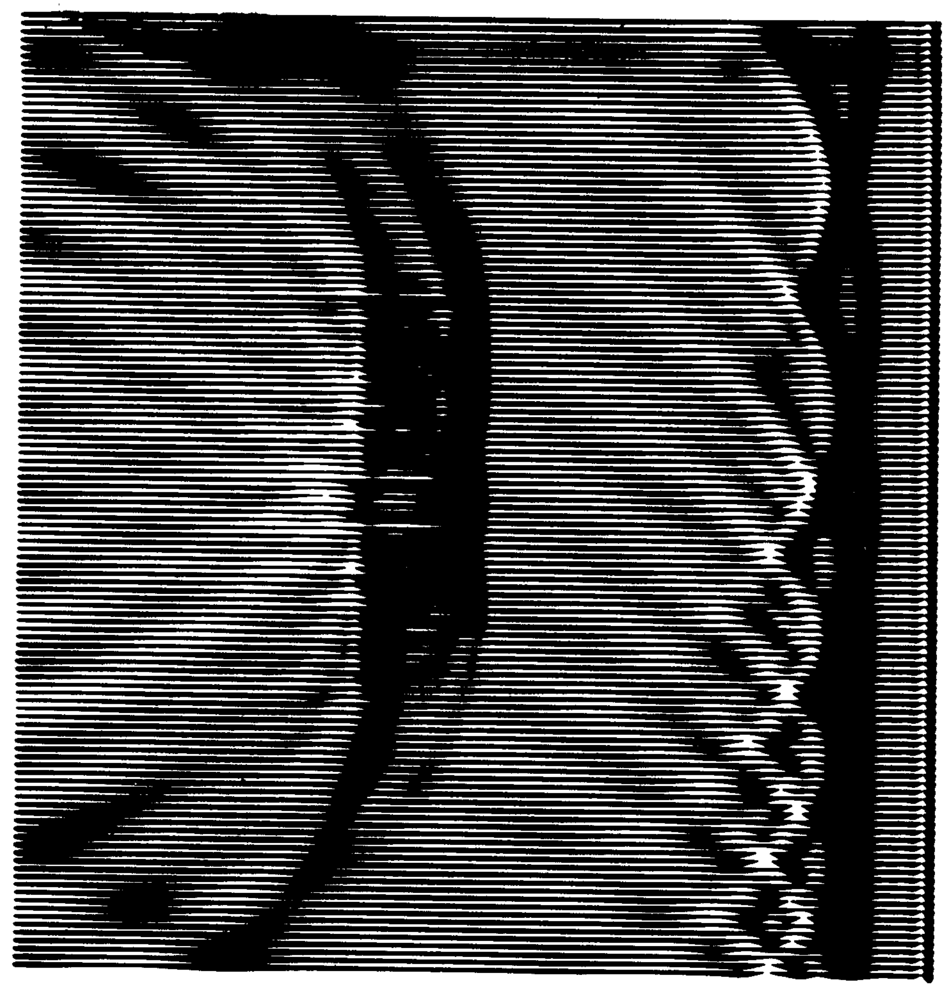
Diffraction Frame 0



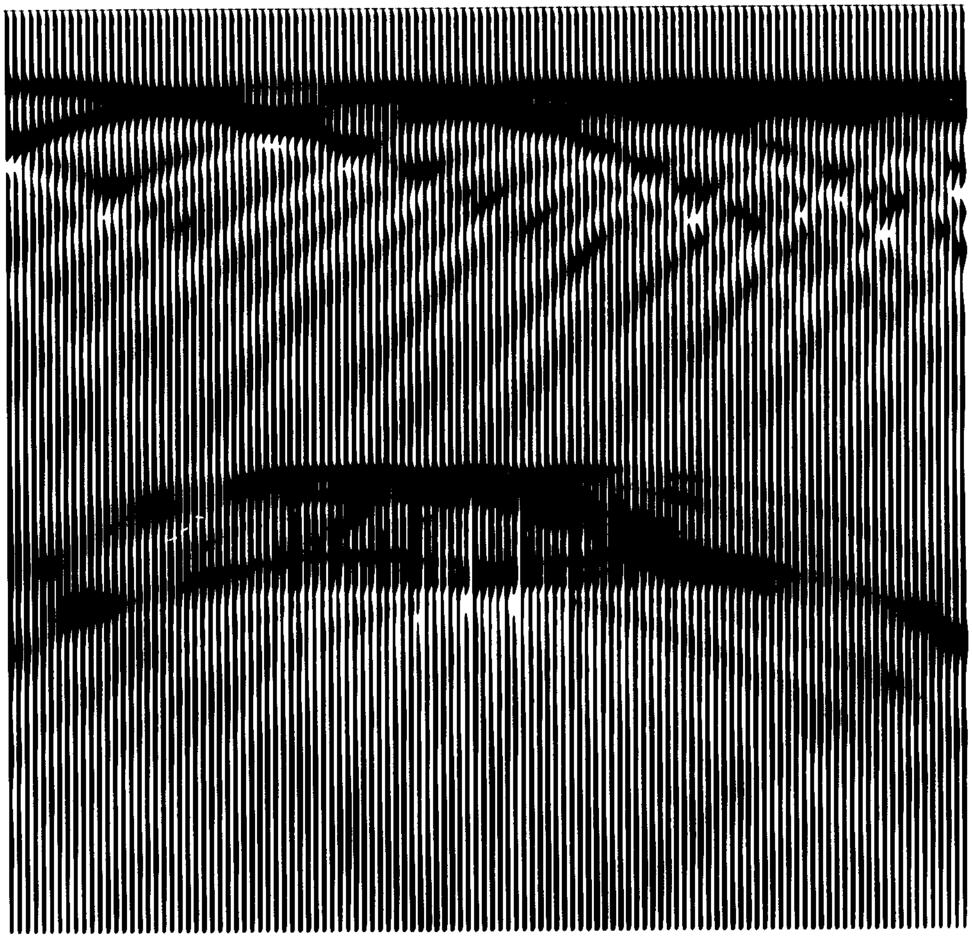
Diffraction Frame 5



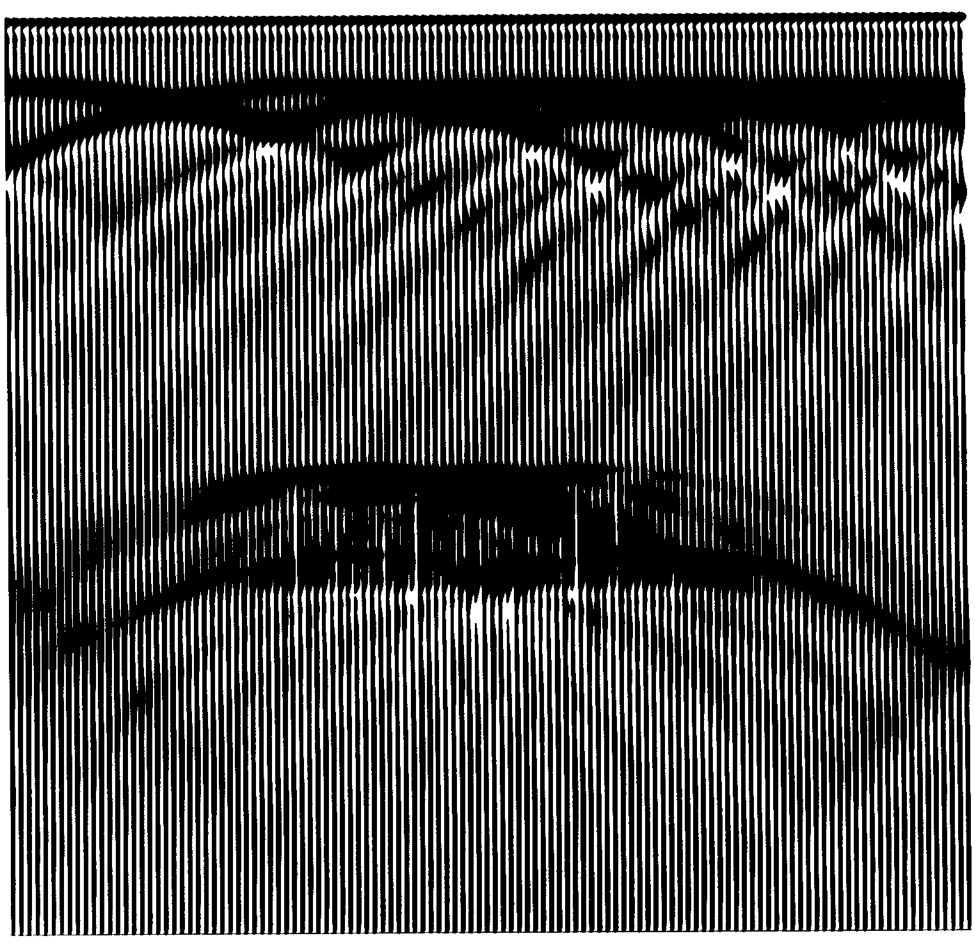
Diffraction Frame 15



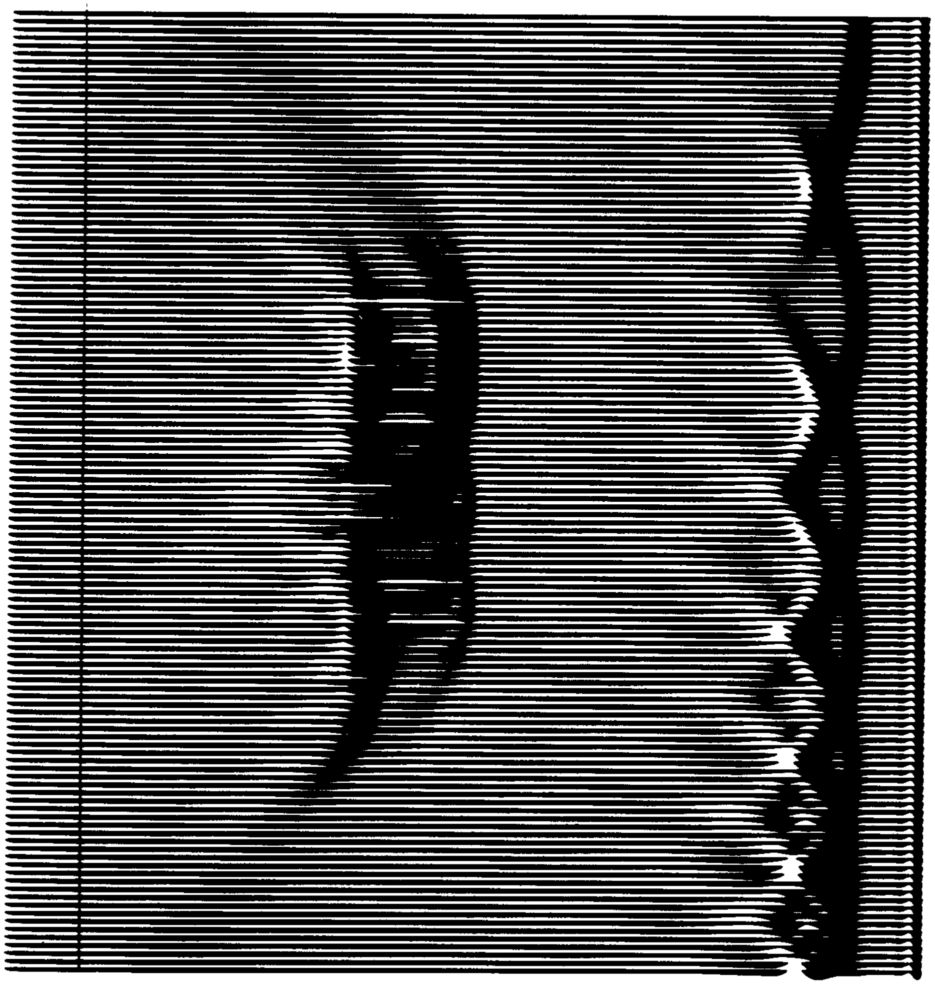
Diffraction Frame 10



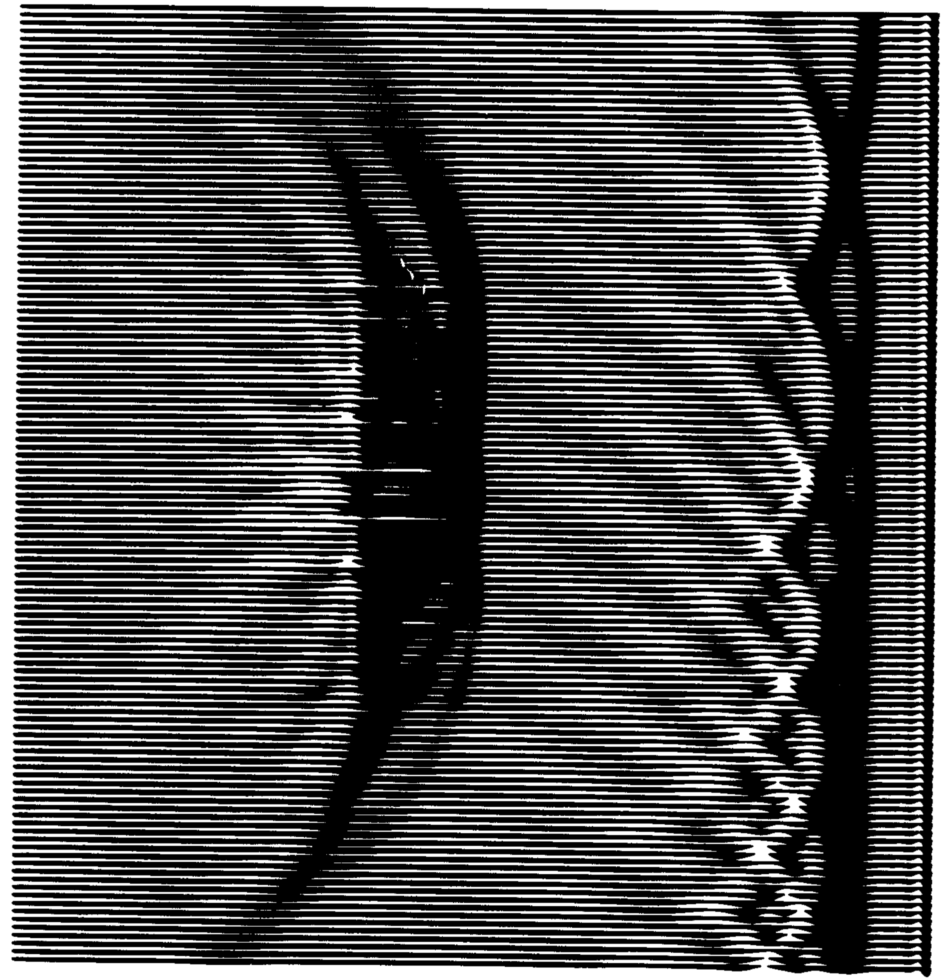
Diffraction Frame 20



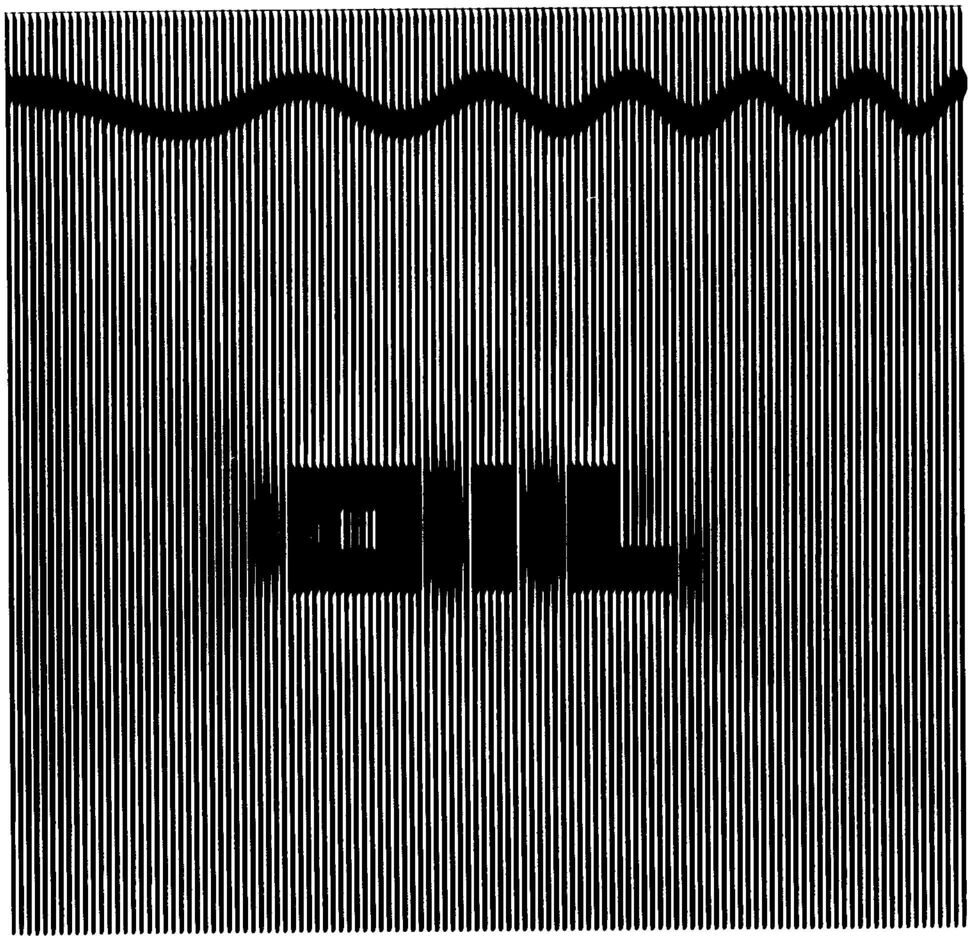
Migration Frame 15



Migration Frame 5



Migration Frame 10



Migration Frame 0

The above frame is the attempted reconstruction of the original wave field (diffraction frame 0) after being continued 20 steps in the $+z$ direction, then 20 steps in the $-z$ direction. It is very interesting to note the apparent good quality of the reconstruction. Energy lost flowing off the bottom of the grid is responsible for the imperfections in the reconstruction. Theoretically, the results cannot be expected to be as good as those above. The exact one-way wave equation does not model evanescent energy ($|k| > |\omega/c|$) . These non-propagating waves (such as those arising from the vertical bars in the letters) may not, in theory, be used in the migration. The migrating differential equation we used does model energy in the entire(ω,k) space, though incorrectly in some portions (excessive dip and k_x). However, when we run time backwards in migrating a diffracted wavefield the invalid portions of (ω,k) are treated the same way as when time runs forward. Reversibility of the equation accounts for the good reconstruction.