LEVITY: Levinson Recursion Reprogrammed

by Jon F. Claerbout

This is the description of a program for prediction and shaping filters. The present version resembles the Burg algorithm (p. 137 in my book, hereafter referred to as FGDP). It has, however, been generalized to get its reflection coefficients both in the conventional way NORM = 2 and with medians NORM = 1 . There is also the possibility of forcing filters to have all positive coefficients IPOLE = 1 or the usual option of permitting both positive and negative coefficients IPOLE = 2 . Also, the forward prediction reflection coefficient c^+ is not constrained to equal the backward prediction coefficient c^- , so in this respect the algorithm resembles the multichannel Levinson recursion on p. 141 and 142 in FGDP. There has been no attempt to keep c^+ and c^- less than unity in magnitude, however it is easily shown that

NORM = 2

$$-1 \leq \frac{2}{\frac{1}{c^+} + \frac{1}{c^-}} \leq 1 \qquad (1)$$

NORM = 1

$$c^+ c^- \leq 1 \qquad (2)$$

It should be quite easy to go into the program and make slight modifications to force $c^+ = c^- \leq 1$ if you are interested in spectral analysis of stationary series.

The Golub method of doing least squares requires about half the computing precision as inverting A^TA, likewise, I believe the Burg algorithm and the present program require about half the precision of the usual Levinson recursion.

Let us define the program's data inputs

$(x_i, i=1,N)$ = input time series

$(y_i, i=1,N)$ = desired output series

The general objective is to find a causal filter f which when convolved with x gives y. Outputs of the program will be

e_i^+ = forward prediction error residual of x

e_i^- = backward prediction error residual of x

d_i^+ = shaping filter residual = filtered x minus y

d_i^- = shaping filter residual = filtered x minus y

If you are not interested in shaping filters you can simply remove from the program all reference to y, d^+, and d^-.

Optional inputs may be prepared by appending a delta function to x_i and zero to y_i as follows:

$$x_i \leftarrow \begin{cases} x_i & , i = 1, N \\ 0 & , i = N+1, N+LC-1 \\ 1 & , i = N+LC \\ 0 & , i = N+LC+1 , N+2*LC-1 \end{cases}$$

$$y_i \leftarrow \begin{cases} y_i\ , & i = 1,\ N \\ 0\ , & i = N+1,\ N+2*LC-1 \end{cases}$$

These optional inputs give x and y an extended length

$$NN = N + 2 * LC - 1 \qquad (3)$$

The optional inputs result in the optional outputs

a_i = forward prediction error filter = e_i^+ , i=N+LC,N+2*LC-1

b_i = backward prediction error filter = e_i^- , i=N+1,N+LC

f_i^+ = Levinson shaping filter = d_i^+ , i=N+LC,N+2*LC-1

f_i^- = Simpson shaping filter = d_i^- , i=N+LC,N+2*LC-1

The preparation of the optional inputs is in lines 13 to 18 in the test program

```
 5.          DIMENSION X(21),Y(21),EP(21),EM(21),DP(21),DM(21)
 6.          N=10
 7.    20    READ 40,JOB,NORM,IPOLE
 8.          CALL SETJOB(JOB,LC,N,X,Y)
 9.          PRINT 30
10.          PRINT 40,JOB,LC,N,NORM,IPOLE
11.    30    FORMAT('   JOB     LC      N   NORM IPOLE')
12.    40    FORMAT(5I6)
13.          NN=N+2*LC-1
14.          N1=N+1
15.          DO 50 I=N1,NN
16.          Y(I)=0.
17.    50    X(I)=0.
18.          X(N+LC)=1.
19.          PRINT 10,(Y(I),I=1,NN)
20.          PRINT 10,(X(I),I=1,NN)
21.          CALL LEVITY(NORM,IPOLE,NN,N,X,EP,EM,Y,DP,DM,LC)
22.    10    FORMAT(17F6.2)
23.          PRINT 10,(EP(I),I=1,NN)
24.          PRINT 10,(EM(I),I=1,NN)
25.          PRINT 10,(DP(I),I=1,NN)
26.          PRINT 10,(DM(I),I=1,NN)
27.          GO TO 20
28.          END
```

A basic subroutine is PROJEK . Its inputs are NORM, IPOLE, NN, N, Y, X . First it computes a projection coefficient c of X onto Y using the first N components in X and Y .

NORM=1

$$c = \underset{i=1}{\overset{N}{\text{Median}}} \left(y_i / x_i \right) \tag{4a}$$

NORM=2

$$c = \left(\sum_{i=1}^{N} x_i y_i \right) / \left(\sum_{i=1}^{N} x_i^2 \right) \tag{4b}$$

If IPOLE=1 , the positive filter coefficient option has been selected, so if c is positive it is reset to zero. Then PROJEK produces its output YP by

$$y'_i = y_i - c\, x_i \qquad (i=1,NN) \tag{5}$$

The subroutine is

```
      SUBROUTINE PROJEK(NORM,IPOLE,NN,N,Y,X,YP)
      DIMENSION Y(NN),X(NN),YP(NN),T(21)
      GO TO (10,30),NORM
10    DO 20 I=1,N
20    T(I)=Y(I)/(X(I)+1.E-20)
      CALL SORT(N,T)
      C=(T((N+1)/2)+T((N+2)/2))/2.
      GO TO 50
30    XX=0.
      XY=0.
      DO 40 I=1,N
      XX=XX+X(I)*X(I)
40    XY=XY+X(I)*Y(I)
      C=XY/XX
50    IF(IPOLE.EQ.1.AND.C.GT.0.) C=0.
      DO 60 I=1,NN
60    YP(I)=Y(I)-C*X(I)
      RETURN
      END
```

get median

get $\frac{X \cdot Y}{X \cdot X}$

$y' = y - cx$

This program calls a SORT subroutine to order the numbers before selecting the median. Thus, it is very inefficient but it is fool proof and easy to document with the simple ordering routine

```
71.             SUBROUTINE SORT(N,T)
72.       C     SLOW SORTER
73.             DIMENSION T(N)
74.             DO 20 J=1,N
75.             DO 20 I=J,N
76.             IF(T(I).GE.T(J)) GO TO 20
77.             TEMP=T(I)
78.             T(I)=T(J)
79.             T(J)=TEMP
80.          20 CONTINUE
81.             RETURN
82.             END
```

The method for getting the prediction error residuals closely follows that on p. 136 of FGDP .

Initialize

$$e_i^{+(1)} = e_i^{-(1)} = x_i \qquad (i=1,NN)$$

Then for j = 2,LC

$$e_{i+j-1}^{+(j)} = e_{i+j-1}^{+(j-1)} - c\, e_i^{-(j-1)} \qquad (i = 1,NN-j+1) \qquad (6a)$$

$$e_i^{-(j)} = e_i^{-(j-1)} - c\, e_{i+j-1}^{+(j-1)} \qquad (i = 1,NN-j+1) \qquad (6b)$$

Except for the shifted indices it will be noted that both (6a) and (6b) take the form (5). Hence, they are computed by a succession of calls to PROJEK. Since $e^{+(j)}$ is overlain onto $e^{+(j-1)}$ a temporary storage vector T is used.

The Levinson shaping filter is defined on page 142 of FGDP to be a linear combination of backward prediction error filters. One may also deduce that the Levinson shaping filter residual of order j, namely $d^{+(j)}$ is defined by initializing $d_t^{+(1)}$ to $-y_t$ and then removing the best projection of the successive backward prediction error residuals from d, namely,

<u>initialize</u>

$$d_i^{+(0)} = -y_i \qquad (i=1,NN)$$

<u>then for j=1,LC</u>

$$d_{i+j-1}^{+(j)} = d_{i+j-1}^{+(j-1)} - \gamma^{+} e_i^{-(j)} \qquad (i=1,NN-j+1) \quad (7)$$

where γ^{+} is the c found by PROJEK of e^{-} onto d^{+}. The subroutine itself is

```
      SUBROUTINE LEVITY(NORM,IPOLE,NN,N,X,EP,EM,Y,DP,DM,LC)
      DIMENSION X(NN),Y(NN),EP(NN),EM(NN),DP(NN),DM(NN),T(21)
      DO 10 I=1,NN                                  e+  e-  d+  d-
      DP(I)=-Y(I)
      DM(I)=-Y(I)
      EM(I)=X(I)
   10 EP(I)=X(I)
      DO 40 JP=1,LC
      NJP=N-JP+1
      NNJP=NN-JP+1
      IF(JP.EQ.1) GO TO 30                                        / eqn (6a)
      CALL PROJEK(NORM,IPOLE,NNJP,NJP,EP(JP),EM,T(JP))
      CALL PROJEK(NORM,IPOLE,NNJP,NJP,EM,EP(JP),EM)              — eqn (6b)
      DO 20 I=JP,NN
   20 EP(I)=T(I)
   30 CALL PROJEK(NORM,IPOLE,NNJP,NJP,DP(JP),EM,DP(JP))          eqn (7)
      JM=LC-JP+1
      NJM=N-JM+1
      NNJM=NN-JM+1
      CALL PROJEK(NORM,IPOLE,NNJM,NJM,DM(JM),EP,DM(JM))          like (7)+(8)
   40 CONTINUE
      RETURN
      END
```

I have defined the Simpson shaping filter to be like the Levinson shaping filter except that it is a superposition of forward prediction error filters. In the case of least squares, both filters turn out to be the same. Although Levinson set his recursion to work on increasing lags of cross correlation, I'm defining the Simpson filter by having the recursion work on decreasing lags. The idea is seen by comparing equation (7-5-4) of FGDP to

$$\left[\ R\ \right]\left\{\begin{bmatrix}0\\ F_{N-2}\\ F_{N-1}\\ F_N\end{bmatrix}+\begin{bmatrix}I\\ A_1\\ A_2\\ A_N\end{bmatrix}\gamma\right\}=\left\{\begin{bmatrix}\bar{E_f}\\ G_{N-2}\\ G_{N-1}\\ G_N\end{bmatrix}+\begin{bmatrix}V\\ 0\\ 0\\ 0\end{bmatrix}\gamma\right\} \qquad (8)$$

(Actually "Simpson's sideways recursion" is slightly different. He develops F to order N ; then reduces it to $N-1$ by removing B ; then boosts it back up to order N with A .)

End Effects. The program uses Burg treatment of the ends of the data vectors, that is, there is no presumption of zero values beyond the ends. The program may however be used to get the usual Levinson results simply by appending at least $LC-1$ zeros on both ends of the input series. In fact, a test case used throughout the program debugging was

$$x = (0, 0, 2, 1, 0, 0)$$

$$y = (0, 0, 0, 85, 0, 0)$$

which for $LC=3$ implies the Toeplitz set

$$\begin{bmatrix} 5 & 2 & 0 \\ 2 & 5 & 2 \\ 0 & 2 & 5 \end{bmatrix} \begin{bmatrix} 1 \\ 40 \\ -16 \end{bmatrix} \qquad \begin{bmatrix} 85 \\ 170 \\ 0 \end{bmatrix}$$

Data initialization for various tests is in Figure 1. Results of the tests are in Figure 2.

```
      SUBROUTINE SETJOB(JOB,LC,N,X,Y)
      DIMENSION C(21),X(N),Y(N)
      DO 10 I=1,N
      X(I)=0.
      Y(I)=0.
10    C(I)=0.
      GO TO (100,200,300),JOB
100   CONTINUE
      LC=2
      C(2)=1.
      C(6)=1.
      C(7)=1.
      Y(2)=1.
      DO 20 I=2,N
20    X(I)=X(I-1)*.5+C(I)
      RETURN
200   CONTINUE
      LC=3
      X(3)=2.
      X(4)=1.
      Y(4)=85.
      RETURN
300   CONTINUE
      X(5)=1.
      X(6)=.4
      X(7)=-.4
      Y(5)=1.
      LC=3
      RETURN
      END
$DATA
      2     2     2
      1     2     2
      1     1     2
      2     2     1
      3     2     1
$STOP
```

Figure 1. Initialization of LEVITY test program.

```
   JOB     LC      N   NORM  IPOLE
     2      3     10      2      2
Y  0.00   0.00   0.00  85.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
X  0.00   0.00   2.00   1.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   1.00   0.00   0.00
e+ 0.00  -0.00   2.00   0.05  -0.10   0.19  -0.00  -0.00  -0.00  -0.00  -0.00  -0.00   1.00  -0.48   0.19  A
e- 0.38  -0.76   1.52   1.00  -0.00  -0.00  -0.00  -0.00  -0.00  -0.00   0.19  -0.48   1.00  -0.00   0.00  B
d+ 0.00  -0.00   2.00  -4.00   8.00-16.00  -0.00  -0.00  -0.00  -0.00  -0.00  -0.00   1.00  40.00-16.00  F+
d- 0.00  -0.00   2.00  -4.00   8.00-16.00  -0.00  -0.00  -0.00  -0.00  -0.00  -0.00   1.00  40.00-16.00  F-
   JOB     LC      N   NORM  IPOLE
     1      2     10      2      2
  0.00   1.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
  0.00   1.00   0.50   0.25   0.13   1.06   1.53   0.77   0.38   0.19   0.00   1.00   0.00
  0.00   1.00  -0.21  -0.11  -0.05   0.97   0.77  -0.33  -0.16  -0.08  -0.14   1.00  -0.71
 -0.71   0.65   0.32   0.16  -0.63  -0.02   0.99   0.49   0.25   0.19  -0.71   1.00   0.00
 -0.00  -0.64  -0.08  -0.04  -0.02   0.35   0.28  -0.12  -0.06  -0.03  -0.05   0.36  -0.26
 -0.00  -0.64  -0.08  -0.04  -0.02   0.35   0.28  -0.12  -0.06  -0.03  -0.05   0.36  -0.26
   JOB     LC      N   NORM  IPOLE
     1      2     10      1      2
  0.00   1.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
  0.00   1.00   0.50   0.25   0.13   1.06   1.53   0.77   0.38   0.19   0.00   1.00   0.00
  0.00   1.00  -0.00  -0.00  -0.00   1.00   1.00  -0.00  -0.00  -0.00  -0.10   1.00  -0.50  A
 -2.00  -0.00  -0.00  -0.00  -2.00  -2.00  -0.00  -0.00  -0.00   0.19  -2.00   1.00   0.00  B
 -0.00  -1.00  -0.00  -0.00  -0.00  -0.00  -0.00  -0.00  -0.00  -0.00  -0.00  -0.00  -0.00
 -0.00  -1.00  -0.00  -0.00  -0.00  -0.00  -0.00  -0.00  -0.00  -0.00  -0.00  -0.00  -0.00
   JOB     LC      N   NORM  IPOLE
     2      3     10      2      1
  0.00   0.00   0.00  85.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
  0.00   0.00   2.00   1.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   1.00   0.00   0.00
  0.00  -0.00   2.00   1.00  -0.00  -0.00  -0.00  -0.00  -0.00  -0.00  -0.00  -0.00   1.00  -0.00  -0.00
 -0.00  -0.00   2.00   1.00  -0.00  -0.00  -0.00  -0.00  -0.00  -0.00  -0.00  -0.00   1.00  -0.00   0.00
 -0.00  -0.00  34.00-13.60  27.20  -0.00  -0.00  -0.00  -0.00  -0.00  -0.00  -0.00  17.00  27.20  -0.00  F+
 -0.00  -0.00   6.80-13.60  34.00  -0.00  -0.00  -0.00  -0.00  -0.00  -0.00  -0.00   3.40  34.00  -0.00  F-
   JOB     LC      N   NORM  IPOLE
     3      3     10      2      1
  0.00   0.00   0.00   0.00   1.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
  0.00   0.00   0.00   0.00   1.00   0.40  -0.40   0.00   0.00   0.00   0.00   0.00   1.00   0.00   0.00
  0.00  -0.00  -0.00  -0.00   1.00   0.40  -0.10   0.12  -0.12  -0.00  -0.00  -0.00   1.00  -0.00   0.30
 -0.00  -0.00   0.30   0.12   0.88   0.40  -0.40  -0.00  -0.00  -0.00   0.30  -0.00   1.00  -0.00   0.00
 -0.00  -0.00  -0.00  -0.00  -0.17   0.33  -0.08   0.10  -0.10  -0.00  -0.00  -0.00   0.83  -0.00   0.25
 -0.00  -0.00  -0.00  -0.00  -0.17   0.33  -0.08   0.10  -0.10  -0.00  -0.00  -0.00   0.83  -0.00   0.25
```

Figure 2. Results of test program.