APPENDIX

A P P E N D I X

A2.1 RATIONALISATION OF INPUT DATA

```
COMPILE RNDATA $R
MICRASOFT COMPILER V2.0
        1: REM RATIONALISATION OF DATA
        2: LPRINTER WIDTH 80
        3: INPUT "N=" ; N
        4: INPUT "K =" ; K
        5: DIM X(N,K)
        6: INPUT "DATA FILE =" ; DF
        7: OPEN DF AS 1
        8: FOR J = 1 TO K
        9: FOR I = 1 TO N
       10: READ #1 ; X( I,J )
       11: NEXT  I
       12: NEXT  J
       13: CLOSE 1
       14: FOR  J = 1  TO  K
       15: MIN = X( 1,J )
       16: FOR  I = 1  TO  N
       17: IF MIN < X( I,J ) THEN 19
       18: MIN = X( I,J )
       19: 19 NEXT  I
       20: MAX = X( 1,J )
       21: FOR I = 1  TO  N
       22: IF MAX > X( I,J )  THEN 24
       23: MAX = X( I,J )
       24: 24 NEXT  I
       25: FOR I = 1  TO  N
       26: X( I,J ) = ( X( I,J ) - MIN )/( MAX - MIN )
       27: NEXT  I
       28: NEXT  J
```

```
   29 :  INPUT "FILE NAME =" ; A$
   30 :  CREATE A$ AS 2
   31 :  FOR  J = 1  TO  K
   32 :  FOR  I = 1  TO  N
   33 :  PRINT # 2 ; X( I,J )
   34 :  NEXT  I
   35 :  NEXT  J
   36 :  CLOSE  2
   37 :  PRINT
   38 :  STOP
   39 :  END
NO ERRORS DETECTED
CONSTANT AREA :          8
CODE SIZE :            603
DATA STMT  AREA :        0
VARIABLE AREA  :        78
A
```

A1.8 DETERMINATION OF CORRELATION CO-EFFICIENTS

```
COMPILE THIS $R
MICRASIC COMPILER V2.0
   1: LPRINTER WIDTH 80
   2: REM CORRELATION
   3: INPUT "DATA, VARIABLES, STEPS = "; N,K,K
   4: DIM Y( N,K ), YM( K ), COR ( K-1, K )
   5: INPUT "DATA FILE ="; D$
   6: OPEN D$ AS  1
   7: FOR J = 1  TO  K
   8: FOR I = 1  TO  N
   9: READ # 1 ; Y ( I,J )
  10: NEXT  I
  11: NEXT  J
  12: CLOSE  1
  13: FOR  J = 1  TO  K
  14: SUM = 0
  15: FOR  I = I  TO  N
  16: SUM = SUM + Y ( I,J )
  17: NEXT  I
  18: YM( J) = SUM/N
  19: NEXT  J
  20: M1 = K + 1
  21: FOR  K1 = 1  TO  M1
  22: SUMA = 0
  23: M2 = N - K1 + 1
  24: FOR  I = 1  TO  M2
  25: SUMA = SUMA+ (Y( I,1) - YM(1))*(Y(I,1) - YM(1))
  26: NEXT  I
  27: FOR  J = 1  TO  K
  28: SUMC = 0
  29: SUMD = 0
  30: FOR  I = 1  TO  M2
```

```
    31: II = I + K1 - 1
    32: SUM2 =SUM2 + (Y(I,1) -YM(1))*(Y(II,J) - YM(J))
    33: SUM2 = SUM2 + (Y ( II,J ) - YM(J))*(Y(II,J) -YM(J))
    34: NEXT I
    35: COR ( K1,J ) = SUM2/DQR ( SUM1*SUM2 )
    36: NEXT J
    37: NEXT K1
    38: PRINT TAB ( 20 ); "CORRELATION": PRINT
    39: FOR K1 = 1 TO K1
    40: K11 = K1 - 1
    41: PRINT TAB ( 10 );"ZERO VARIANT (";K11 ;")" : PRINT
    42: FOR J = 1 TO K
    43: PRINT COR ( K1, J ),
    44: NEXT J
    45: PRINT
    46: NEXT K1
    47: PRINT
    48: STOP
    49: END
NO ERRORS DETECTED
CONSTANT AREA :          0
CODE SIZE     :        907
DATA STMT AREA :         0
VARIABLE AREA  :       1AA
A
```

```
A1.3  DETERMINATION OF INPUT ARGUMENTS
COMPILE NEXTART $2
MICRASIC COMPILER V2.0
 1: REM CALCULATION OF PRIMESTART
 2: LPRINTER WIDTH  80
 3: INPUT "N,K,M1, M2 =";  N,K,M1,M2
 4: DIM X(N,K), Y(M1,M2)
 5: INPUT "FILE NAME =" ; N$
 6: OPEN N$ AS  1
 7: FOR  J = 1  TO  K
 8: FOR  I = 1  TO  N
 9: READ # 1 ; X( I,J )
10: NEXT  I.
11: NEXT  J
12: CLOSE  1
13: FOR  I = 1  TO  M1
14: Y( I,1 ) = X( I + 4, 1 )
15: Y( I,2 ) = X(I + 3,1) : Y( I,3 ) = X( I + 2, 1 )
16: Y( I,4 ) = X(I + 1,1) : Y( I,5 ) = X( I,1 ) : Y( I,6 ) = I
17: NEXT  I
18: INPUT "FILE NAME =" ; A$
19: CREATE A$ AS  2
20: FOR  J = 1  TO  M2
21: FOR  I = 1  TO  M1
22: PRINT # 2 ; Y ( I, J )
23: NEXT  I
24: NEXT  J
25: CLOSE  2
26: PRINT
27: STOP
28: END
NO ERRORS DETECTED
CONSTANT  AREA :          0
CODE SIZE  :          543
DATA WORK AREA :          0
VARIABLE AREA  :         80
A
```

```
A1.A  ON LINE RECURSIVE INSTRUMENT VARIABLES ALGORITHM

COMPILE MARKV $E
MEDASYS COMPILER V8.0
  1: PRINT TAB (10); "ON LINE HOURLY ESTIMATED FLOWS"
  2: PRINT TAB (10); "AT CORONATION MINING POINT OF THE TEESTA RIVER"
  3: PRINT
  4: EXPLICIT VISUAL ON
  5: REM LEAST SQUARE, LSQ, RECURSIVE & LSQ.INS.IV.ALGORITHMS
  6: INPUT "NO. OF DATA, NO. OF PREDICTORS, VARIABLE + 1 =";NL,NP,N
  7: NP =  ( N - 1 ) * ( NP + 1 ) + NP
  8: REM TO (NL,N),SV1(NP), P(NP,NP), G1(NP), SV(NP)
  9: REM PE ( NL + 2), YA( NL,4 ), P1(NP), P2 (NP )
 10: REM YE( NL ), GE( NP ), SVE ( NP )
 11: INPUT "FILE NAME : CORON. INP";D$
 12: OPEN D$ AS 1
 13: FOR J = 1 TO N : FOR I = 1 TO NL
 14: READ # 1 ; YE( I,J ): NEXT I, J : CLOSE 1
 15: INPUT " INITIAL PREDICTORS, NP = " ; NP
 16: FOR N = NP TO NP : NS = N + ( N - 1 )*( N + 1 )
 17: INPUT "WARMUP ALGM, REQD. ? YES = 1  NO = 0 =";WAR
 18: IF WAR = 0  THEN 500
 19: INPUT "NO. OF ERROR TERMS MV =" ; MV
 20: MT = N + MV +( N - 1 )*( N + 1 )
 21: REM S(MT,MT),A(MT),  Z(MT),RA(MT),MB(MT),MG(MT)
 22: REM MK ( MT )
 23: FOR I = 1 TO MT : A(I) = 0: Z(I)= 0: FOR J = 1 TO  MT
 24: S(I,J) = 0:S( I,I ) = 1 : NEXT  J, I
 25: MN1 = 1 : MN2 = NL
 26: PRINT "PARAMETERS TRACKING ALGMS." : PRINT
 27: PRINT "SERIAL NO.,  OBSERVED,  MODELLED & ERROR" : PRINT
 28: FOR KD = MN1 TO MN2: SUMZ = 1 : FOR  I = 1  TO  MT
 29: SUMA = 0: SUMB = 0 : FOR  KD = 1  TO  MT
 30: SUMA = SUMA+ S(I,KD)*Z(KD): SUMB =SUMB+ Z(KD)*S(KD,I)
```

```
51: NEXT KD:BA(I) = SUBA:BD(I)= SUBD: NEXT I: FOR J=1 TO NP
52: SUBD= SUBD + BD(I)*Z(I): NEXT I
53: FOR I = 1 TO NP: DC(I) = BA(I)/SUBD:FOR J = 1 TO NP
54: S( I,J ) = S( I,J ) - DC( I )* BD( J ): NEXT J,I
55: SUBA= YD(KD,1): FOR I = 1 TO NP: SUBA = SUBA -A(I)* Z(I)
56: NEXT I: PE( KD ) = SUBA: YE( KD )= YD( KD,1) - PE( KD )
57: PRINT KD, YD( KD,1 ), YE( KD ), PE( KD ) : PRINT
58: IF KD = KM2 THEN 600
59: FOR J=1 TO NP: SUBD = 0: FOR J= 1 TO NP
60: SUBD = SUBD + S( I,J )* Z( J )
61: BE( I ) = SUBD : NEXT J, I
62: FOR I = 1 TO NP: A( I ) = A( I )+ PE( KD )* BE( I ): NEXT I
63: REM GENERATION OF Z - VECTOR
64: REM I = 1 TO N:KKL = KD - I + 1
65: IF KKL<= 0 THEN 5
66: Z(I) = YD( KKL,1 ) : GO TO 4
67: 5 Z( I ) = 0
68: 4 NEXT I
69: FOR I = 1 TO NV:KKL = KD - I + 1 : II = N + I
70: IF KKL <= 0 THEN 6
71: Z( II ) = PE( KKL )
72: GO TO 7
73: 6 Z( II ) = 0
74: 7 NEXT I
75: FOR J = 2 TO N: FOR I = 1 TO N + 1: KKL = KD - I
76: IF KKL = 0 THEN 8
77: II=N+NV*( J-2)*( N+1 )+ I: SV( II ) = YD( KKL,J )
78: GO TO 9
79: 8 Z( II ) = 0
80: 9 NEXT I
81: NEXT J
82: NEXT KD
83: 600 INPUT "PARAMETRIC ALGOL, ERROR FILE =";LE$
84: CREATE LE$ AS 4: FOR KD = 1 TO KL : PRINT # 4; PE( KD )
```

```
65: NEXT I0: CLOSE 1
66: SUM1=0:SUM2=0:SUM3=0:FOR I=1 TO N1:SUM1 = SUM1 +V1(I)*V1(I)
67: SUM2 = SUM2 + V0(I,1)* V0(I,1) : SUM3 = SUM3 + V1 (I)
68: NEXT I : SUM2 = SUM2/SUM3 : SUM1 = SUM1 / ( N1 - N )
69: PRINT " EUCLIDEAN SQUARE ERROR & MEAN ERROR =" ; SUM2, SUM1
70: 500 INPUT " LOG SKIP = 1  OTHERWISE = 0 =" ; SKIP
71: IF SKIP = 1  THEN 105
72: PRINT "LARGE SQUARE METHOD" : PRINT
73: FOR I = 1  TO  N3 : SV(I) = 0 : SV1(I) = 0
74: NEXT I : FOR  I = 1  TO N3 : FOR  J = 1  TO  N3
75: P(I,J) = 0 : NEXT  J, I : N2 = N1 - N
76: INPUT " HOW MANY DATA SKIPPED FOR LOG CONV =" ; LOG
77: FOR  I = 1  TO  N2  - LOG : NI = I
78: GOSUB  100
79: GO TO 11
80: 100 REM GENERATION OF X - VECTOR
81: FOR I = 1  TO N : II = N + NI - I
82: SV(I) = V0(II,1) : SV 2(I) = V0(II) : NEXT  I
83: FOR J = 2 TO N : FOR I = 1  TO  N -1:II= N+1-I:I1=II-1: V0(J-1)*(V0,1)=I
84: SV( IJ ) = V0(II, J): SV 2(IJ) = V0(II,J): NEXT I, J: RETURN
85: 11 FOR I = 1  TO  N3 : FOR  J = 1  TO N3
86: P( I,J ) = P ( I,J ) + SV( I ) * SV( J ) : NEXT  J,I
87: FOR I = I  TO N3 : NI = N + I
88: SV1(I) = SV1(I) + V0( NI,1 )* SV ( I )
89: NEXT  I, N : GOSUB 200
90: INPUT " MATRIX P(I,J = "; NI
91: GRAPH NI 20  0
92: FOR I = 1 TO N3 : FOR J = 1  TO N3
93: PRINT J 0 : P( I,J ) : NEXT J, I : CLOSE 0
94: GO  TO  21
95: 200  REM MATRIX INVERSION
96: FOR L = 1  TO  N3 : X = 1 / P( L,L ) : P( L,L ) = 1
97: FOR  I = 1  TO  N3 : P(I,L) = P(I,L) * X
98: NEXT I : FOR J = 1 TO  N3
```

```
 99: IF J = L THEN 80
100: X = P ( L,J ) : P ( L,J ) = 0 : FOR I = 1 TO N3
101: P( I,J ) = P( I,J ) - P( I,L )* X : NEXT I
102: GO NEXT J: NEXT L : RETURN
105: SI  FOR  I =1 TO N3 : SUM = 0: FOR J = 1 TO N3
104: SUM = SUM + P ( I,J )* EV1( J ): NEXT J
105: G1(I) = SUM: NEXT I : PRINT "LSQ. COEFF.="
106: FOR I = 1 TO N3 : PRINT G1 (I), : NEXT I : PRINT
107: INPUT "LSQ COEF FILE NAME =" ; CF$
108: CREATE CF$ AS 3
109: FOR I = 1 TO N3 : PRINT # 3 ; G1(I) : NEXT I : CLOSE 3
110: PRINT " LEAST SQUARE INTERPOLATION" : PRINT
111: PRINT " SERIAL NO.,  OBSERVED, CALCULATED & ERROR ": PRINT
112: FOR  K = 1  TO  N2
113: KK = K : GOSUB  200
114: SUM = 0 : FOR I = 1  TO  N3
115: SUM = SUM + EV (I)* G1(I): NEXT  I
116: YB = SUM: KK = K + K: ERR = YB( KK, 1 ) - YB
117: YA(K,1) = KK: YA( K,2 ) = YB( KK,1 ): YA( K,3 ) = YB
118: YA( K,4 )= ERR:PRINT YA(K,1), YA(K,2),YA( K,3), YA(K,4): NEXT K
119: INPUT " LSQ ERROR FILE NAME "; L1 $
120: CREATE L1 $ AS 2
121: FOR K = 1 TO N2 : PRINT # 2 ; YA(K,4) : NEXT K :  CLOSE 2
122: SUMS = 0: SUMM = 0 : SUMA = 0
123: FOR K = 1 TO  N 2 : SUMA = SUMA + YA( K,4 )
124: SUMM = SUMM + YA( K,4 )* YA( K,4 )
125: SUMS = SUMS + YA( K,2 )*YA( K,2 ): NEXT K
126: SUMM = (SUMM / SUMS ): SUMA = SUMA / N 2
127: PRINT "INTEGRAL SQR.ERROR & MEAN ERROR ="; SUMM, SUMA
128: PRINT : PRINT : PRINT : PRINT
129: GO TO  33
130: 205 INPUT "LSQ COEF FILE NAME ="; CF$
131: OPEN CF$ AS  3 : FOR  I = 1  TO N3
132: READ  # 3 ; G1(I) : NEXT I: CLOSE  3
```

```
153: INPUT "  FILE NAME =" ; XX $
154: OPEN XX $ AS 9
155: FOR I = 1  TO N 5 : FOR J = 1 TO  N 5 .
156: READ # 9 ; P(I,J) : NEXT J,I : CLOSE  9
157: INPUT "LAST STEP = 2 OTHERWISE  0 = "; STEP
158: IF  STEP = 2  THEN 505
159: 55 PRINT " LEAST SQUARE RECURSIVE ALGORITHMS"
160: PRINT: PRINT " SERIAL NO.,  OBSERVED, MODELLED & ERROR": PRINT
161: FOR  I=1 TO N 5: SV(I) = 0:SVR(I) = 0:GI(I) =GI(I): NEXT  I
162: NA = NL - N - 2
163: FOR  K = 1  TO  NA
164: KL = K: GOSUB  100
165: FOR J=1 TO  N 5 : SUM = 0 : FOR  J = 1  TO  N 5
166: SUM = SUM + P(I, J)* SV(I) : NEXT  J
167: P1(I)= SUM : NEXT I : D = 1.0
168: FOR I = 1  TO N 5 : D = D + SV(I)* P1(I)
169: NEXT I : FOR J = 1  TO N 5 : SUM = 0
170: FOR I = 1  TO  N 5 : SUM = SUM + SV(I)* P(I,J)
171: NEXT I : P 1(J) = SUM : NEXT  J
172: D = 1.0 / D: FOR I= 1  TO  N 5 : FOR  J = 1  TO  N 5
173: P(I,J) = P(I,J) - P1( I)* P1(J)*D
174: NEXT J,I : KL =  K + 1 : GOSUB  100
175: FOR J=1  TO  N 5: SUM = 0: FOR  J = 1 TO  N 5
176: SUM = SUM + P(I,J)* SV(J) : NEXT  J
177: P1(I) = SUM : NEXT I : D = 1.0 : FOR  I = 1  TO  N 5
178:  D = D + SV(I) * P1(I) :  NEXT  I
179: D = 1 / D : ML = 0.0 : FOR  I = I TO  N 5
180: ML = ML+ SV(I)* GI(I) : NEXT I : NK 2 = N + K + 1
181: Y = YB (NK 2,1) - ML : Y = Y*D : FOR I = 1  TO  N  5
182: GI(I) = GI(I) + P1(I) * Y : Next  I
183: NK 2 = NK 2 + I :  KL = KL +  1 : GOSUB 100
184: YB = 0: FOR I = 1 TO  N 5  : YB = YB + SV(I)* GI(I)
185: NEXT I : ERR = YB( NK 2, 1 ) - YB
186: YA(K,1) = NK 2: YA(K,2) = YB(NK 2,1): YA( K,3 ) = YB
187: YA(K,4) = ERR: PRINT YA(K,1),YA(K,2),YA(K,3),YA(K,4) : PRINT
```

```
168: NEXT  K
169: INPUT "LOGR ERROR FILE NAME =" ; L $
170: CREATE L $ AS 4 : FOR K = 1 TO 10:PRINT #4; ZA(K,4):NEXT K:CLOSE 4
171: SUMA = 0 :   SUMB = 0 : SUMC = 0
172: FOR K = 1 TO  N4  : SUMA = SUMA + ZA( K,4 )
173: SUMB = SUMB + ZA( K,4 )* ZA( K,4 )
174: SUMC = SUMC + ZA( K,2 )* ZA( K,2 ) : NEXT  K
175: SUMA = SUMA / N4 : SUMB = ( SUMB / SUMC )
176: PRINT "LOGR INTCPT, OR ERROR & MEAN ERROR.=";SUMB, SUMA
177: INPUT "FILE FILE NAME ="; XXX $
178: OPEN XXX$ AS 10 : FOR I = 1  TO  N3  : FOR J = 1  TO  N3
179: READ # 10 ; P(I,J) : NEXT  J, I : CLOSE  10
180: GO TO  99
181: REM INPUT " PAR. FILE NAME NAM. I = " ; A $
182: OPEN A$ AS  1  FOR  I = 1  TO  NL
183: READ # 1 ; PR(I) : NEXT I : CLOSE  1
184: FOR  I = 1 TO NL: YR(I) = YR(I,1) - PR(I) : NEXT  I
185: 99 PRINT : PRINT : PRINT : PRINT
186: PRINT "LOG. ERRS. INDP. ALGN."
187: PRINT : PRINT " SERIAL NO., OBSERVED, NORMALIZED & ERRORS": PRINT
188: N4 = NL - 2^N - 2
189: FOR I = 1  TO  N3 :  OX( I ) = OX( I )
190: NV(I) = 0 : NV1(I) = 0: NV2(I) = 0 : NEXT  I
191: FOR K = 1  TO  N4
192: KL = K :KL=KL+1:GOSUB 100:FOR J=1 TO N 3 : NV1(I) =NV(I)
193: NEXT  I : KL = K
194: FOR  I = 1  TO  N 3 : SUM = 0
195: FOR J = 1  TO  N 3 : SUM = SUM + P(I,J)* NV2(J)
196: NEXT J :  P1(I) = SUM : NEXT  I
197: D = 1 : FOR I = 1  TO  N 3: D = D *NV1(I)* P1(I)
198: NEXT I : FOR J = 1  TO  N 3 : SUM = 0
199: FOR I = 1  TO  N 3 : SUM = SUM + NV1(I)* P(I,J)
200: NEXT I :  P 2(J) = SUM : NEXT  J
201: D = 1 / D : FOR  I = 1  TO  N 3
202: FOR J = 1 TO N 3: P(I,J)= P(I,J) - P1(I)* P 2 (J)* D
```

```
203: NEXT J,I : K 2 = K1 + 1
204: K1 = N + K 2 : GOSUB 100
205: FOR I = 1 TO N3 : XV1(I) = XV(I) : NEXT I : K1 = K + 2
206: FOR I = 1 TO N 3 : SUM = 0 : FOR J = 1 to N 3
207: SUM = SUM + P( I,J )* XV2(J) : NEXT J
208: P1( I ) = SUM : NEXT I : D = 1.0
209: FOR I = 1 TO N 3 : D = D + XV(I)* P1(I)
210: NEXT I : D = 1.0 / D : Z1 = 0.0
211: FOR I = 1 TO N 3 : Z1 = Z1 + XV1(I)* GZ(I)
212: NEXT I : MK 2 = 2 * N + K 2
213: Y = YD( MK 2 , 1 ) - Z 1 : Y = Y* D
214: FOR I = 1 TO N 3 : GZ(I) = GZ(I) + P1(I) * Y : NEXT I
215: MK 2 = MK 2 +1 : K1 = N + K2 + 1 : GOSUB 100 : K1 = K + 1
216: YD = 0 : FOR I = 1 TO N 3
217: YD = YD + XV1 ( I )* GZ( I ) : NEXT I
218: MKK = YD ( MK 2,1 ) - YD
219: YA( K,1) = MK 2 : YA( K,2 ) = YD( MKK,1 )
220: YA ( K,3 ) = YD : YA( K,4 ) = MKK
221: PRINT YA(K,1),YA(K,2),YA(K,3),YA(K,4) : PRINT : NEXT K
222: INPUT " ENTER MODEL FILE =" ; Q$
223: CREATE Q$ AS 7
224: FOR K = 1 TO N 4
225: PRINT # 7 ; YA( K,4 ) : NEXT K
226: CLOSE 7
227: SUMA = 0 : SUMB = 0 : SUMC = 0
228: FOR K = 1 TO N 4 : SUMA = SUMA + YA ( K,4 )
```

```
249 : SUMB = SUMB + YA ( K,4 ) * YA ( K,4 )
250 : SUMC = SUMC + YA ( K,2 ) * YA ( K,2 ) : NEXT K
251 : SUMA = SUMA / N4 : SUMD = ( SUMB / SUMC )
252 : PRINT " INTEGRAL SQR ERROR & MEAN ERROR"
253 : PRINT SUMD, SUMA : PRINT
254 : PRINT : PRINT : PRINT
255 : NEXT N
256 : STOP
257 : END
NO ERRORS DETECTED
CONSTANT AREA  :          24
CODE SIZE  :            2469
DATA STMT AREA :           0
VARIABLE AREA  :         960
A
```

A1.5 [illegible] OF COMPLEX VARIABLE OF [illegible]

```
COMPILE NUMBER 00
MICRONIC COMPILER V 2.0
   1: PRINT TAB(20);"TESTING FOR COMPLEX VARIABLE & MEAN [illegible]";PRINT
   2: [illegible] WIDTH 80
   3: REM TWO STATIONARY DOUBLY FLOW MODEL
   4: INPUT "DATA =";N
   5: INPUT "L L1, L L2 =" ; L L1, L L2
   6: DIM Y ( N )
   7: INPUT "DATA FILE=";N $
   8: OPEN N $  AS 1
   9: FOR I = 1  TO  N
  10: READ  # 1  ; Y ( I )
  11: NEXT  I
  12: CLOSE  1
  13: FOR N 1 = 2  TO  6
  14: NP = N 1
  15: DIM S(NP,NP),A(NP),B(NP),BA(NP),BB(NP),BS(NP)
  16: DIM BE(NP), PR( NP )
  17: FOR I = 1  TO  NP
  18: A(I) = 0
  19: B(I) = 0
  20: FOR J = 1,  TO  NP
  21: S(I,J) = 0
  22: S(I,I) = 1
  23: NEXT  J
  24: NEXT  I
  25: FOR K = 1  TO  N
  26: [illegible] = 1
  27: FOR  I = 1  TO  NP.
  28: SUMA = 0 : SUMB = 0
  29: FOR  KD = 1  TO  NP
  30: SUMA = SUMA + S ( I,KD ) * E ( KD )
```

```
31: SUMB = SUMB + X ( KD ) * S ( KD, I )
32: NEXT KD
33: BA ( I ) = SUMA : BB ( I ) = SUMB
34: NEXT I
35: FOR I = 1 TO NT
36: SUMT = SUMT + BB ( I ) * X ( I )
37: NEXT I
38: FOR I = 1 TO NT
39: BC ( I ) = BA ( I ) / SUMT
40: FOR J = 1 TO NT
41: S ( I,J ) = S ( I,J ) - BC ( I ) * BB ( J )
42: NEXT J : NEXT I
43: SUMA = Y ( K )
44: FOR I = 1 TO NT
45: SUMA = SUMA - A ( I ) * X ( I )
46: NEXT I
47: PE ( K ) = SUMA
48: FOR I = 1 TO NT
49: SUMD = 0
50: FOR J = 1 TO NT
51: SUMD = SUMD + S ( I,J ) * X ( J )
52: BK ( I ) = SUMD : NEXT J : NEXT I
53: FOR I = 1 TO NT
54: A ( I ) = A ( I ) + PE ( K ) * BK ( I )
55: NEXT I
56: REM GENERATION OF X VECTOR
57: FOR I = 1 TO N1
58: KK1 = K - I + 1
59: IF KK1 <= 0 THEN 5
60: X ( I ) = Y ( KK1 )
61: GO TO 6
62: 5 X ( I ) = 0
63 6 NEXT I
64: IF K < L1 THEN 3
65: IF K > L2 THEN 3
```

```
 66: PRINT "K = "; K
 67: PRINT " A ( I ) "
 68: FOR  I = 1  TO  NT
 69: PRINT  A ( I ) ,
 70: NEXT  I
 71: PRINT " S ( I ) "
 72: FOR  I = 1  TO  NT
 73: PRINT  S ( I ),
 74: NEXT  I
 72: J NEXT  K
 76: SUMX = 0 : SUMY = 0 : SUMXX = 0
 77: KK = NL + 1
 78: FOR  K = KK  TO  N
 79: SUMX = SUMX + PR ( K ) * PR ( K )
 80: SUMY = SUMY + Y ( K ) * Y ( K )
 81: SUMXX = SUMXX + PR ( K )
 82: NEXT  K
 83: PR ( N + 1 ) = SUMX / SUMY
 84: PR ( N + 2 ) = SUMXX / N
 85: PRINT "INT.SQR. & MEAN ERROR =";PR(N+1),PR(N+2): PRINT
 86: PRINT "NL+ 1 ="; KK : PRINT
 87: PRINT "FREQUENCY INTERVAL ="; NL
 88: INPUT "FILE NAME ="; C$
 89: CREATE  C$ AS  2
 90: FOR  I = KK  TO  N
 91: PRINT  # 2 ; PR ( I )
 92: NEXT  I : CLOSE  2
 93: PRINT : PRINT : PRINT
 94: NEXT  N1
 95: STOP
 96: END
NO ERRORS DETECTED
CONSTANT AREA :          8
CODE SIZE     :       1965
DATA STMT AREA :          0
VARIABLE AREA  :        258
A
```

```
A0.1 MODIFIED GROUP METHOD DATA HANDLING ALGORITHM
COMPILE GMDH $R
MICROSOFT COMPILER V2.0
   1: REM  CONSOLE
   2: DEFINTEGER W,N,N1 : CLS
   3: PRINT TAB( 10 ); "GMDH REGRESSION" : PRINT
   4: INPUT "DATA, VARIABLES =" ; N,K
   5: P = ( K - 1 ) * ( K - 2 ) / 2
   6: REM X ( N,K ), X1 ( N )
   7: INPUT "DATA FILE =" ; D $
   8: OPEN D $ AS  1
   9: FOR  J = 1  TO  K
  10: FOR I = 1  TO  N
  11: READ #1 , X ( I,J )
  12: NEXT  I
  13: NEXT  J
  14: CLOSE  1
  15: NM = N * .6 : N 1 = NM +1
  16: REM Y (N), XM (M), A1 (6), B1 (6)
  17: REM B2 (6), M3(6), M4(6), M5(6), M6(6), S(6)
  18: REM M0(6), M7(6), M8(6), M9(6), B6(6), M2(6)
  19: REM M4(6), M5(6), B6(6), F4(6), F5(6), F6(6)
  20: REM C5(6), C6(6), C(6), YH( P,N1 )
  21: FOR  I = 1  TO  N
  22: Y(I) =  X( I,1 )
  23: NEXT  I
  24: K1 = 0
  25: FOR  J = 2  TO  K - 1
  26: J1 = J - 1
  27: FOR  I = 1  TO  N
  28: X 1 (I) = X (I,J)
  29: NEXT  I
  30: JJ = J + 1
  31: FOR  L = JJ  TO  K
  32: K3 = K1 + L - J
```

```
55: AT1 = L - 1
56: FOR I = 1 TO N
57: X 2 (I) = X (I, L)
58: NEXT I
59: FOR I = 1 TO 6
60: A 1 (I) = 0
61: NEXT I
62: FOR I = 1 TO N
63: A 1 ( 1 ) = A 1 ( 1 ) + Y ( I )
64: A 1 (2) = A 1 (2) + Y (I) * X 1 (I)
65: A 1 (3) = A 1 (3) + Y (I) * X 2 (I)
66: A 1 (4) = A 1 (4) + Y (I) * ( X 1 (I) * X 2 (I))
67: A 1 (5) = A 1 (5) + Y (I) * ( X 1 (I) * X 1 (I))
68: A 1 (6) = A 1 (6) + Y (I) * ( X 2 (I) * X 2 (I))
69: NEXT I
70: B 1 ( 1 ) = N
71: FOR I = 2 TO 6
72: B 1 ( I ) = 0
73: NEXT I
74: FOR I = 1 TO N
75: B 1 (2) = B 1 (2) + X 1 (I)
76: B 1 (3) = B 1 (3) + X 2 (I)
77: B 1 (4) = B 1 (4) + X 1 (I) * X 2 (I)
78: B 1 (5) = B 1 (5) + X 1 (I) * X 1 (I)
79: B 1 (6) = B 1 (6) + X 2 (I) * X 2 (I)
80: NEXT I
81: B 2 ( 1)= B 1 (2) : B 2 ( 2 ) = B 1 (3) : B 2 (3) = B 1 (4)
82: FOR I = 4 TO 6
83: B 2 (I) = 0
84: NEXT I
85: FOR I = 1 TO N
86: B 2 (4) = B 2 (4) + X 1 (I) * ( X 1 (I) * X 2 (I))
87: B 2 (5) = B 2 (5) + X 1 (I) * X 1 (I) * X 1 (I)
88: B 2 (6) = B 2 (6) + X 1 (I) *(X 2 (I) * X 2 (I))
```

```
67: NEXT I
68: B 5 (1) = B 1 (5) : B 5 (2) = B 2 (5) : B 5 (3) = B 1 (6)
69: B 5 (4) = B 2 (6) : B 5 (5) = B 5 (4)
70: B 5 (6) = 0
71: FOR I = 1 TO N
72: B 5 (6)= B 5 (6) + X 2 (I) * ( X 2 (I) * X 2 (I))
73: NEXT I
74: B 4 (1) = B 1 (4) : B 4 (2) = B 2 (4) : B 4 (3) = B 3 (4)
75: FOR I = 4 TO 6
76: B 4 (I) = 0
77: NEXT I
78: FOR I = 1 TO N
79: B 4 (4) = B 4 (4) + (X 1(I)*X 1(I))*(X 2(I)* X 2 (I))
80: B 4 (5) = B 4 (5) + (X 1(I)*X 1(I))*(X 1(I)* X 2 (I))
81: B 4 (6) = B 4 (6) + (X 2(I)*X 2(I))*(X 1(I)* X 2 (I))
82: NEXT I
83: B 3 (2) = B 1 (2) : B 3 (2) = B 2 (3) : B 3 (3) = B 3 (3)
84: B 3 (4) = B 4 (2) : B 3 (5) = 0 : B 3 (6) = B 4 (2)
85: FOR I = 1 TO N
86: B 3 (3)= B 3 (3)+ (X 1 (I)* X 1 (I)) * (X 1 (I)* X 1 (I))
87: NEXT I
88: B 6 (2) = B 1 (6) : B 6 (2) = B 2 (6) : B 6 (3) = B 3 (6)
89: B 6 (4) = B 4 (6) : B 6 (5) = B 5 (6) : B 6 (6) = 0
90: FOR I = 1 TO N
91: B 6 (6) = B 6 (6)+ (X 2 (I)* X 2 (I))* ( X 2 (I) * X 2 (I))
92: NEXT I
93: Z (1) = B 1 (1) / B 2 (1)
94: FOR I = 2 TO 6
95: B 2 (I) = B 2 (I) * Z (I) - B 1 (I)
96: NEXT I
97: Z (2) = B 1 (1) / B 3 (I)
98: FOR I = 2 TO 6
99: B 3 (I) = B 3 (I) * Z (I) - B 1 (I)
100: NEXT I
101: Z (6) = B 1 (1) / B 4 (1)
```

```
102: FOR I = 2 TO 6
103: B 4 (I) = B 4 (I) * Z (5) - B 1 (I)
104: NEXT I
105: Z (4) = B 1 (I) / B 5 (I)
106: FOR I = 2 TO 6
107: B 5 (I) = B 5 (I) * Z (4) - B 1 (I)
108: NEXT I
109: Z (5) = B 1 (I) / B 6 (I)
110: FOR I = 2 TO 6
111: B 6 (I) = B 6 (I) * Z (5) - B 1 (I)
112: NEXT I
113: FOR I = 1 TO 5
114: II = I + 1
115: A 1 (II) = A 1 (II) * Z (I) - A 1 (I)
116: NEXT I
117: Z (1) = B 2 (2) / B 5 (2)
118: FOR I = 3 TO 6
119: B 5 (I) = B 5 (I) * Z (1) - B 2 (I)
120: NEXT I
121: Z (2) = B 2 (2) / B 4 (2)
122: FOR I = 3 TO 6
123: B 4 (I) = B 4 (I) * Z (2) - B 2 (I)
124: NEXT I
125: Z (3) = B 2 (2)/ B 5 (2) : Z (4) = B 2 (2)/ B 6 (2)
126: FOR I = 3 TO 6
127: B 5 (I) = B 5 (I) * Z (3) - B 2 (I)
128: B 6 (I) = B 6 (I) * Z (4) - B 2 (I)
129: NEXT I
130: FOR I = 1 TO 4
131: II = I + 2
132: A 1 (II) = A 1 (II) * Z (I) - A 2 (2)
133: NEXT I
134: Z (1) = B 3 (3)/ B 4 (3) : Z (2) = B 3 (3)/ B 5 (3)
135: Z (3) = B 3 (3)/ B 6 (3)
136: FOR I = 4 TO 6
137: B 4 (I) = B 4 (I) * Z (1) - B 3 (I)
```

```
130: P 5(I) = E 5 (I) * Z(2)  - E5 (I)
139: P 6(I) = E6 (I) * Z(5)  - E5 (I)
140: NEXT I
141: FOR  I = 1  TO  3
142: II = I + 3
143: A 1(II) = A 1 (II) * B(I) - A 1 (3)
144: NEXT  I
145: Z(1) = P 4(4)/P 5(4) : Z(2) = P 4(4)/ P 6(4)
146: FOR  I = 3  TO  6
147: G 5(I) = P 5 (I) * Z(I) - P 4 (I)
148: G 6(I) = P 6 (I) * Z(2) - P 4 (I)
149: NEXT  I
150: FOR  I = 1  TO  2
151: II = I + 4
152: A 1 (II) = A 1 (II) * Z(I) - A 1 (4)
153: NEXT  I
154: Z (1) = G 5 (5)/ G 6 (5)
155: E 6 = G 6 (6) * Z(1) - G 5 (6)
156: A 1 (6) = A 1 (6) * E (1) - A 1 (5)
157: C (6) = A 1 (6)/ E 6
158: C (5) = ( A 1 (5) - G 5 (6) * C (6))/ G 5(5)
159: C (4) = ( A 1 (4) - P 4 (5) * C (5) - P 4(6) * C(6))/ P 4 (4)
160: A 2 = E 5(4)* C(4) :  A 5 = E 5(5) * C(5)
161: A 4 = E 5(6) * C(6)
162: C(3) = ( A 1 (3) - A 2 - A 5 - A 6 )/ E 5 (3)
163: A 2 = D 2 (3) * C (3) : A 5 = D 2 (4) * C (4)
164: A 4 = D 2 (3) * C (3) : A 5 = D 2 (6) * C (6)
165: A 2 = A 2 + A 5 + A 4 + A 5
166: C (2) = ( A 1(2) - A 2 )/ D 2(2)
167: A 2 = E1(2) * C(2)  : A 3 = B 1 (3) *  C(3)
168: A 4 = B 1 (4) * C(4) : A 5 = B 1(5) * C(5)
169: A 6 = B 1 (6) * C(6) : A Z = A 2 + A 3 + A 4  + A 5 + A 6
170: C (1) =  ( A 1 (1) - A 2 )/ B 1 (1)
```

```
171: FOR I = 1  TO  6
172: XB ( K 3, N + I ) = C (I)
173: NEXT  I
174: FOR  I = 1  TO  N
175: A = C (1) + C(2) * X 1 (I) + C(3) * X 2 (I)
176: B = C (4) * X 1(I) * X 2(I)
177: C 1 = C(5) * X 1 (I) * X 1 (I)
178: D = C(6) * X 2 (I) * X 2 (I)
179: XB ( K 3, I ) = A + B + C 1 + D
180: NEXT  I
181: SUMM =  0 : SUMP = 0
182: FOR  I = 1  TO  N
183: SUMM = SUMM +(Y(I) - XB(K 3,I))*(Y(I) - XB (K 3,I))
184: SUMP = SUMP + Y(I) * Y(I)
185: NEXT  I
186: DEL 1 = SUMM / SUMP
187: PRINT TAB (10);"DEL 1 (";K 3 ;") =";  DEL 1
188: PRINT
189: XB ( K 3, NV + 1 ) = J1
190: XB ( K 3, NV + 2 ) = JJ1
191: XB ( K 3, NV + 3 ) = DEL 1
192: NEXT  L
193: K L = K 3
194: NEXT  J
195: FOR  I = 1  TO  P - 1
196: FOR  J = I + 1  TO  P
197: IF XB ( J,N 1 ) > Y B ( I,N 1)  THEN  10
198: FOR  L = 1  TO  N 1
199: S 1 = Y B ( I,L )
200: XB (I,L) = XB ( J,L )
201: XB ( J,L ) = S 1
202: NEXT  L
203: 10  NEXT  J
204: NEXT  I
205: REM FOR I = 1  TO  K - 1
```

```
    206: REM PRINT " ROW = " ; I
    207: REM FOR  J = N + 1  TO  N 1
    208: REM  PRINT  YB ( I, J ),
    209: REM  NEXT  J
    210: REM  NEXT  I
    211: INPUT "COEFF. & ERR FILE COFF. I ="; C $
    212: CREATE C $ AS 1 : FOR I = 1 TO K - 1 : FOR J = N + 1  TO  N 1
    213: PRINT # 1 ; YB ( I,J ) : NEXT  J : NEXT I: CLOSE  1
    214: FOR  J = 2  TO  K
    215: FOR  I = 1  TO  N
    216: X ( I,J ) = YB ( J - 1 , I )
    217: NEXT  I
    218: NEXT  J
    219: INPUT "DATA FILE ="; A $
    220: CREATE A $  AS  2
    221: FOR  J = 1  TO  K
    222: FOR  I = 1  TO  N
    223: PRINT # 2 ; X ( I, J )
    224: NEXT  I
    225: NEXT  J
    226: CLOSE  2
    227: PRINT
    228: STOP
    229: END
NO ERRORS DETECTED
CONSTANT AREA :          0
CODE SIZE :           6116
DATA SENT AREA :         0
VARIABLE AREA :        472
A
```

A5.1 DETERMINATION OF PERIODICITY IN THE INPUT DATA

```
COMPILE SPECTRA $ B
MBASIC COMPILER V 2.0
    1: REM POWER DENSITY
    2: LPRINTER WIDTH 80
    3: INPUT "DATA, VARS. = "; N,K
    4: N = K : N 1 = N + 1
    5: DIM P(N), PS(N 1 ), MP(N 1), S( N 1 ), C A ( N 1 ), RA ( N 1 )
    6: INPUT "FILE NAME = "; B $
    7: OPEN B $  AS 1
    8: FOR  I = 1  TO  N
    9: READ # 1 ; P (I)
   10: NEXT  I
   11: CLOSE  1
   12: SUM = 0
   13: FOR  I = 1  TO  N
   14: SUM = SUM + P (I)
   15: NEXT  I
   16: PBM = SUM / N
   17: FOR  K = 1  TO  M 1
   18: NK = N - K : SUM = 0
   19: FOR I = 1  TO  NK
   20: II = I + K - 1
   21: SUM = SUM + (P(I) - PBM) * (P(II) - PBM)
   22: NEXT  I
   23: CA(K) = SUM/(N - K + 1) : RA (K) = CA (K)/ CA (1)
   24: NEXT  K
   25: HK =.5
   26: FOR  KK  =1  TO  M 1
   27: C = KK - 1 : A = C * 3.142/ M :  SUM = 0
   28: MP ( KK ) = .5 * A/ 3.142
   29: FOR  K = 1  TO  M 1
   30: B = A * ( K - 1 )
```

```
   31: IF  K = 2  THEN  HK  = 1.
   32: IF  K = N 1  THEN  HK = .5
   33: SUM = SUM + HK * HA (K) * COS (B)
   34: NEXT K
   35: PS ( KK ) =  SUM *  2 / 3.000
   36: NEXT KK
   37: S(1) = .54 * PS(1) + .46 * PS(2)
   38: S(N 1) = .54 * PS(N 1) + .46 + PS(N)
   39: FOR  I = 2  TO  N
   40: S(I) = .23 * PS(I - 1) + .54 * PS(I) + .23 * PS (I + 1)
   41: NEXT  I
   42: PRINT "SPECTRA"
   43: FOR  I = 1  TO  N 1
   44: PRINT  S(I),
   45: NEXT  I
   46: PRINT "FREQUENCY"
   47: FOR  I = 1  TO  N 1
   48: PRINT  HP(I),
   49: NEXT  I
   50: PRINT
   51: STOP
   52: END
NO ERRORS DETECTED
CONSTANT AREA  :        46
CODE SIZE  :          1035
DATA STMT AREA  :        0
VARIABLE AREA  :       160
A
```

A2.2 COMBINATORIAL GROUP METHOD OF DATA HANDLING ALGORITHM

```
COMPILE GMDCOMB #E
HBASIC COMPILER V 2.0
    1: LPRINTER WIDTH 80
    2: PRINT "COMBINATORIAL GMDH ALGORITHM POLINOMIAL"
    3: PRINT "WITHOUT SQUARE TERMS"
    4: PRINT
    5: INPUT "DATA & VARIABLES" ; N,K
    6: KK = ( K - 1 ) * K / 2 + 2
    7: DIM Y(KK,KK),X(N,K+1),B(KK),XM(N),C(KK),B1(KK),
           YO(KK,KK), CR(15)
    8: DIM XE(N)
    9: INPUT "INPUT DATA FILE NAME" ; D $
   10: OPEN D$ AS 1: FOR I=1 TO N:READ #1;X(I,1):NEXT I
   11: FOR J = 3 TO K+1:FOR I = 1 TO N:READ #1; X(I,J):NEXT I:
           NEXT J:CLOSE 1
   12: FOR I = 1 TO N:X(I,2)=1: NEXT I
   13: PRINT "TEST DATA":FOR I = 1 TO K+1:PRINT X(1,I),:NEXT I
   14: PRINT
   15: INPUT "MATRIX FORMATION TO BE SKIPPED ? YES = 1 NO = 0";A
   16: IF A = 1  THEN 20
   17: FOR I= 1 TO KK - 1: FOR J = 1 TO KK:YO(I,J)=0:NEXT J,I
   18: FOR I= 1 TO N:B(1)=X(I,1):B(2)=1:M=2: FOR L=2 TOK:L1=L+1
   19: FOR J=L1 TO K+1:M=M+1:B(M)=X(I,L) * X(I,J):NEXT J,L
   20: FOR L=1 TO M-1:FOR J=1 TO M:YO(L,J)= YO(L,J)+B(L+1) *B(J):
           NEXT J: NEXT L
```

```
21: NEXT I:INPUT"COMBINATORIAL MATRIX FILE NAME COMBN.I";A$
22: CREATE A$ AS 1: FOR J=1 TO KK:FOR I=1 TO KK - 1
23: PRINT #1; YO(I,J): NEXT I,J: CLOSE 1
24: GO TO 30
25: 20 INPUT "COMBINATORIAL MATRIX FILE COMBN.I"; A$
26: OPEN A$ AS 1:FOR J=1 TO KK:FOR I=1 TO KK - 1
27: READ #1; YO(I,J): NEXT I, J : CLOSE 1
28: 30 INPUT "HOW MANY COMBN WANT TO TEST ?"; NL
29: FOR LOOP = 1 TO NL : FOR J = 1 TO KK: FOR I=1 TO KK - 1
30: Y( I,J ) = YO( I,J ): NEXT I,J
31: INPUT "HOW MANY TERMS OF POLMN. TO BE SKIPPED ?"; TN
32: IF TN = 0 THEN 40
33: IF TN = 100 THEN 100
34: FOR V = 1 TO TN: INPUT "WHICH ONE SERIALLY ?"; CR( V )
35: IF CR ( V ) = KK  THEN  60
36: FOR J=CR(V) - V+1 TO KK-V: FOR I = CR(V)-V TO KK - V - 1
37: Y(I,J)=Y(I+1,J+1):NEXT I,J,V
38: 40 FOR J = 1 TO KK-TN:B1(J)= Y(J,1): NEXT J
39: FOR J=1 TO KK-TN-1:FOR I=1 TO KK-TN-1:Y(I,J)=Y(I,J+1):NEXT I,J
40: NN=KK-TN-1:PRINT"NO. OF TERMS OF THE TESTED POLMN.";NN:PRINT
41: PRINT"Y(1,J)":PRINT:FOR J=1 TO NN:PRINT Y(1,J); :NEXT J:PRINT
42: REM MATRIX INVERSION ALGORITH
43: 60 FOR L = 1 TO NN
44: Z=1/Y(L,L):PRINT"Z=";Z:PRINT:Y(L,L)=1:FOR I=1 TO NN
45: Y(I,L) = Y(I,L) * Z:NEXT I: FOR J=1  TO  NN
```

```
46: IF J = L THEN  50
47: Z = Y(L,J): Y(L,J) = 0: FOR I = 1 TO NN
48: Y( I,J ) = Y ( I,J ) - Y( I,L )* Z : NEXT  I
49: 50 NEXT J
50: NEXT  L
51: PRINT "MATRIX INVERSED"
52: PRINT TAB (10);"COEFFICIENTS OF THE POLYNOMIAL": PRINT
53: FOR I=1 TO NN:SUM=0:FOR J=1 TO NN:SUM=SUM+Y(I,J)* B1(J)
54: NEXT J:C(I)=SUM:PRINT C(I),: NEXT I
55: FOR I=1 TO N:SUM=0:B(1)=X(I,1): B(2)= X(I,2):M=2
56: FOR L=2 TO K:L2=L+1:FOR J=L2 TO K+1:M=M+1
57: B(M)= X(I,L)* X(I,J): NEXT J: NEXT L
58: FOR J=2 TO M-TM:SUM=SUM+C(J - 1)  * B(J): NEXT  J
59: XM(I) = SUM : NEXT  I
60: FOR I=1 TO N:XE(I)=X(I,1) - XM(I): NEXT  I
61: PRINT TAB(1);"SERIAL";TAB(15);"OBSERVED";TAB(35);
       "MODELLED VALUES";  TAB
62: FOR I=1 TO N:PRINT TAB(1);I;TAB(15);X(I,1);TAB(35);XM(I);
       TAB(55);  XE(I)
63: PRINT : NEXT  I
64: SUMC = 0: SUMA= 0: SUMB= 0: FOR I = 1  TO  N
65: SUMA=SUMA+XE(I)* XE(I): SUMB=SUMB+X(I,1)* X(I,1)
66: SUMC = SUMC + XE ( I )
67: NEXT I: ERR = SUMA/SUMB: SUMC = SUMC/N
68: PRINT "INTEGRAL SQUARE ERROR =";  ERR : PRINT
69: PRINT "MEAN ERROR =";  SUMC : PRINT
```

```
70: INPUT "ERROR FILE NEEDED ? Y/N = 1/ 0"; A
71: IF A = 0  THEN 11
72: INPUT "ERROR FILE XERR. I"; A$
73: CREATE A$ AS 1: FOR I = 1 TO N:PRINT #1 ; XE( I )
74: NEXT I: CLOSE 1
75: INPUT "MATRIX PRINT FILE REQD. ? YES = 1 NO = 0"; B
76: IF B = 0  THEN  11
77: INPUT "INVERSED MATRIX FILE COMBMAT.I"; A$
78: CREATE A$  AS  1
79: FOR I=1 TO NN:FOR J=1 TO NN:PRINT #1;Y(I,J):NEXT J,I:CLOSE 1
80: 11  NEXT  LOOP
81: 100  STOP
82: END
NO ERRORS DETECTED
CONSTANT AREA :        8
CODE SIZE     :     3668
DATA STMT AREA :       0
VARIABLE AREA  :     264
A
```

A4.1 POWER SYSTEM STATES ESTIMATION

```
COMPILE NBSS $E
MBASIC COMPILER V2.0
    1: LPRINTER WIDTH 80
    2: PRINT TAB(5);"POWER SYSTEM STATE ESTIMATION"
    3: PRINT
    4: INPUT "NO. OF BUSES & NO. OF LINES ="; NB, LPQ
    5: DIM Y(LPQ,6), Z(LPQ),G(NB,NB), B(NB,NB)
    6: INPUT "P - Q; R,X; YPQ1/2 (R & I) FILE YZ ="; Y$
    7: OPEN Y$ AS 1: FOR I=1 TO LPQ : FOR J = 1 TO 6
    8: READ #1 ; Y(I,J): NEXT J,I
    9: CLOSE 1
   10: INPUT "PQRX PRINT NEEDED ? YES = 1; NO=0="; PQR
   11: IF PQR = 0 THEN 501
   12: PRINT "P Q R X YPZ1/2 - R YPQ1/2-X"
   13: FOR I=1 TO LPQ: PRINT" I=";I: FOR J = 1 TO 6
   14: PRINT Y(I,J),: NEXT J: PRINT: NEXT I
   15: 501 FOR I = 1 TO LPQ
   16: Z=Y(I,3) * Y(I,3)+Y(I,4) * Y(I,4):Y(I,3)=Y(I,3)/Z:
           Y(I,4)= -Y(I,4)/Z: NEXT I
   17: INPUT "PQ ADMITTANCE PRINT NEEDED ? YES = 1 NO=0="; L
   18: IF L = 0 THEN 502
   19: PRINT "P-Q LINE ADMITTANCE & LINE CHARGING"
   20: FOR I=1 TO LPQ:PRINT"I=";I:FOR J=1 TO 6:PRINT Y(I,J),:
           NEXT J: PRINT: NEXT I
   21: 502 FOR J=1 TO NB:FOR I=1 TO NB:G(I,J)=0:B(I,J)=0:NEXT I,J
```

```
22: FOR I=1 TO NB:SUMA=0:SUMB=0:FOR J=1 TO LPQ
23: IF (X( J,1 ) -I)<>0 THEN 10
24: SUMA=SUMA+Y (J,3)+Y (J,5):SUMB=SUMB+Y (J,4)+Y (J,6)
25: 10 NEXT J
26: FOR J=1 TO LPQ
27: IF (X(J,2) - I)<>0 THEN 11
28: SUMA=SUMA+Y (J,3)+Y (J,5):SUMB=SUMB+Y (J,4)+Y (J,6)
29: 11 NEXT J
30: G(I,I)= SUMA: B(I,I)= SUMB
31: IF I=NB THEN 9
32: FOR J=1 TO LPQ
33: IF (X(J,1) - I)<> 0 THEN 21
34: II=I+1: FOR K=II TO NB
35: IF (X(J,2)-K)<>0 THEN 41
36: G(I,K)= -Y(J,3):G(K,I)=G(I,K ):B(I,K)= -Y(J,4):B(K,I)=B(I,K)
37: 41 NEXT  K
38: 21 NEXT  J
39: 9 NEXT  I
40: INPUT "BUS ADMITTANCE PRINT NEEDED ? YES=1 NO= 0="; BUS
41: IF BUS = 0 THEN 503
42: PRINT "BUS ADMITTANCE MATRIX": PRINT
43: PRINT "ACTIVE ADMITTANCE COMPONENTS": PRINT
44: FOR I=1 TO NB:PRINT"ROW=";I:FOR J=1 TO NB:PRINT G(I,J),:
          NEXT J: PRINT
45: NEXT I:PRINT:PRINT"REACTIVE ADMITTANCE COMPONENTS"
```

```
46: FOR I=1 TO NB:PRINT"ROW=";I:FOR J=1 TO NB:PRINT B(I,J),:
        NEXT J: PRINT
47: NEXT I:PRINT:PRINT"ASSUMPTION OF BUS VOLTAGES"
48: 503 INPUT"REFERENCE BUS J1=";J1
49: DIM E(NB,1),F(NB,1),PL(NB,4),P(NB,2),Q(NB,2),S(2*NB,2*NB)
50: FOR I=1 TO NB:FOR J=1 TO NB:B(I,J)=-B(I,J):NEXT J,I:FOR I=1 TO N
51: INPUT "E(I,1) & F( I,1) =";  E( I,1),F( I,1):NEXT I
52: INPUT"GENERATION & LOAD POWER FILE =GL=";  B$
53: OPEN B$ AS 2:FOR I=1 TO NB:FOR J=1 TO 4:READ #2;PL(I,J):
        NEXT J,I: CLOSE 2
54: PRINT "GENERATION ACTIVE & REACTIVE LOAD ACTIVE & REACTIVE"
55: INPUT "MVA BASE =";  BASE
56: INPUT "LOAD POWER PRINT NEEDED ? YES=1 NO= 0=";  LP
57: IF LP = 0 THEN 504
58: FOR I=1 TO NB: FOR J=1 TO 4:PRINT PL(I,J),: NEXT J,I
59: 504 FOR I = 1 TO NB
60: PL(I,1)=(PL(I,1)-PL(I,3))/BASE:PL(I,2)= - (PL(I,2)-PL(I,4))/BASE
61: P(I,1) = PL(I,1): Q(I,1)= PL( I,2 )
62: NEXT I
63: INPUT "TOLLERANCE=" ;  TL
64: INPUT "SIGMA=";  SIGMA
65: INPUT "THETA=";  THETA
66: I1=1:NB2=2 * NB
67: FOR I=1 TO NB2 - 1: FOR J=1 TO NB2 - 1: S(I,J) = 0
68: S( I,I ) = SIGMA: NEXT J: NEXT I
69: 100 FOR I=1 TO NB
```

```
70: SUMA = 0 : SUMB = 0
71: FOR J = 1 TO NB
72: P1 =E( I,1) * (E(J,1)*G(I,J)+ F( J,1 ) * B( I,J ))
73: P2 =F( I,1) * (F(J,1)*G(I,J)- E( J,1 ) * B( I,J ))
74: Q1 =F( I,1) * (E(J,1)*G(I,J)+ F( J,1 ) * B( I,J ))
75: Q2 =- E(I,1) *(F(J,1) *G(I,J)- E(J,1 ) * B( I,J ))
76: SUMA = SUMA+ P1+P2: SUMB = SUMB + Q1 + Q 2
77: NEXT J:P(I,2)=SUMA:Q(I,2)= SUMB:NEXT I: KITN= ITN - 1
78: PRINT TAB(10);"REAL AND REACTIVE BUS POWER "
79: PRINT:PRINT "BUS NO. ,P (I,K+1),P (I,K),Q (I,K+1),Q (I,K)":PRINT
80: FOR I=1 TO NB:PRINT TAB(10);"BUS(";I;")": PRINT
81: PRINT P(I,2),P(I,1),Q(I,2),Q(I,1):PRINT: NEXT I: PRINT
82: FOR I=1 TO NB:PRINT TAB(10); Q(I,2),Q(I,1):PRINT:NEXT I:PRINT
83: DIM DELP(NB), DELQ(NB), C(NB), D(NB),JN(NB2,NB2)
84: DIM JN1(NB,NB),JN2(NB,NB),JN3(NB,NB),JN4(NB,NB)
85: DIM DELE(NB),DELF(NB)
86: PRINT TAB(10);"DELP(I) DELQ(I)":PRINT
87: FOR I = 1 TO NB
88: DELP(I)= P(I,1) -P(I,2):DELQ(I)= Q(I,1) - Q( I,2 )
89: PRINT I, DELP(I), DELQ(I): PRINT
90: NEXT I
91: DIM YD(NB,2)
92: FOR I= 1 TO NB
93: YD( I,1 ) = DELP(I): YD( I,2 ) = DELQ(I)
```

```
 94: NEXT I
 95: FOR I = 1  TO NB
 96: IF YD( I,1 )< 0  THEN 58
 97: GO TO 57
 98: 58 YD( I,1 ) = - YD( I,1 )
 99: 57 IF YD( I,2 )< 0  THEN 59
100: GO TO 60
101: 59 YD( I,2 ) = - YD( I,2 )
102: 60 NEXT  I
103: DX1 = 0: DX2 = 0
104: FOR I = 1  TO NB
105: IF ( DX1 - YD ( I,1 ))> 0  THEN  62
106: DX1 = YD( I,1 )
107: 62 IF ( DX 2 - YD ( I,2 ))> 0  THEN  61
108: DX 2 = YD ( I,2 )
109: 61 NEXT  I
110: IF ( DX1 - DX 2 )> 0  THEN  63
111: DX1 = DX2
112: 63 IF ( DX1 - TL )< = 0  THEN  64
113: PRINT "ITERATION COUNT = "; KITN: PRINT
114: FOR I = 1 TO NB: C(I) = 0:D(I)= 0:NEXT I
115: FOR I = 1 TO NB
116: D1 = E( I,1 ) * E( I,1 ) + F( I,1 ) * F( I,1 )
117: C(I)= ( P( I,2 )* E( I,1 )+ Q( I,2 )* F( I,1 ))/D1
118: D(I)= ( P( I,2 )* F( I,1 )- Q( I,2 )* E( I,1 ))/D1
```

```
119: NEXT  I
120: PRINT: PRINT: PRINT
121: INPUT "JACOBIAN REQRD ? YES = 1  NO = 0"; BB
122: IF BB = 0  THEN 555
123: FOR  I = 1  TO NB
124: JN1( I,I ) = E(I,1)* G(I,I)-F(I,1)* B(I,I)+ C(I)
125: NEXT  I
126: FOR  I = 1  TO NB
127: FOR  J = 1  TO NB
128: IF  I = J  THEN 167
129: JN1(I,J) = E(I,1)* G(I,J)-F(I,1)* B(I,J)
130: 167 NEXT  J
131: NEXT  I
132: FOR I = 1 TO NB : FOR J = 1  TO NB
133: JN2 (I,J) = 0: NEXT J,I
134: FOR I = 1  TO NB
135: JN2(I,I) =E(I,1)* B(I,I)+2* F(I,1)* G(I,I)+D(I)
136: NEXT  I
137: FOR I = 1  TO NB
138: FOR J = 1  TO NB
139: IF I = J THEN 72
140: JN2(I,J) = E(I,1)* B(I,J)+ F(I,1)* G(I,J)
141: 72 NEXT  J
142: NEXT  I
143: FOR  I = 1  TO NB
144: FOR  J = 1  TO NB
```

```
145: IF I = J  THEN  74
146: JN3(I,J) = E(I,1)*B(I,J)+ F(I,1)*G(I,J)
147: 74 NEXT  J
148: NEXT  I
149: FOR I = 1  TO NB
150: JN3(I,I) = E(I,1)*B(I,I)+ F(I,1)*G(I,I) - D(I)
151: NEXT  I
152: FOR  I = 1  TO NB
153: FOR  J = 1  TO NB
154: IF I = J  THEN 76
155: JN4(I,J) = - E(I,1)*G(I,J)+ F(I,1)*B(I,J)
156: 76 NEXT  J
157: NEXT  I
158: FOR  I = 1  TO NB
159: JN4(I,I) = -E(I,1)*G(I,I)+ F(I,1)*B(I,I)+ C(I)
160: NEXT  I
161: FOR I=1 TO NB:FOR J=1 TO NB:JN(I,J)= JN1(I,J): NEXT J,I
162: FOR I=1 TO NB:FOR J=NB+1 TO NB2:JN(I,J)=JN2(I,J-NB):NEXT J,I
163: FOR I= NB+ 1 TO NB2 : FOR J = 1  TO NB
164: JN( I,J ) = JN3( I - NB,J) : NEXT  J,I
165: FOR I=NB+1 TO NB2: FOR J= NB+1  TO NB2
166: JN( I,J ) = JN4( I-NB,J-NB) : NEXT  J,I
167: PRINT
168: INPUT "PRINT COPY OF JACOBIAN NEEDED YES = 1 NO=0="; LL
169: IF LL = 0 THEN 555
170: PRINT "ELEMENTS OF JACOBIAN"
```

```
171: PRINT: FOR I = 1 TO NB2: PRINT"ROW=";I
172: FOR J=1 TO NB2:PRINT JN(I,J),:NEXT J:PRINT:NEXT I:PRINT
173: 555 PRINT
174: DIM DELE(NB), DELP(NB)
175: FOR I=1 TO NB:DELE(I)= 0:DELP(I)=0:NEXT  I
176: FOR I = 1  TO NB2
177: FOR J=NB+1 TO NB2-1:JN(I,J)=JN(I,J+1):NEXT J:NEXT I
178: DIM YP(NB2),DELV(NB2),Z(NB2)
179: FOR I=1 TO NB:DELV(I)=DELE(I):NEXT I:FOR I=NB+1 TO NB2-1
180: DELV(I)= DELP(I-NB+1)
181: NEXT I:FOR I=1 TO NB:YP(I)= DELP(I): NEXT I
182: FOR I=NB+1 TO NB2:YP(I)= DELQ(I-NB): NEXT I
183: REM FOR  I=1 TO NB2-1: FOR J=1 TO NB2-1:S(I,J)=0
184: REM S(I,I) = SIGMA: NEXT J:NEXT I
185: FOR IS = 1 TO NB2
186: DIM PE(NB2),DK(NB2),A(NB2),DA(NB2),DB(NB2),DC(NB2)
187: FOR I = 1 TO NB2-1:A(I)= DELV(I): NEXT I
188: FOR I = 1 TO NB2-1
189: Z(I)= JN(IS,I):NEXT I:Y=YP(IS)
190: FOR I=1 TO NB2-1:SUMA=0:SUMB=0:FOR KD=1 TO NB2-1
191: SUMA=SUMA+S(I,KD) * Z(KD):SUMB=SUMB+Z(KD * S(KD,I):NEXT KD
192: DA(I)=SUMA:DB(I)=SUMB:DB(I)=DB(I) * THETA:NEXT I:SUMC=1
193: FOR I=1 TO NB2-1: SUMC=SUMC+DB(I) * Z(I):NEXT I
194: FOR I=1 TO NB2-1:DC(I)= DA(I)/SUMC: FOR J=1 TO NB2-1
195: S(I,J)=S(I,J) -DC(I) * DB(J):NEXT J:NEXT I: SUMA=Y
```

```
196: FOR I=1 TO NB2-1: SUMA=SUMA-A(I)*Z(I):NEXT I:PE(IS) = SUMA
197: FOR I=1 TO NB2-1: SUMD=0: FOR J=1 TO NB2-1
198: SUMD=SUMD+S(I,J)*Z(J)
199: DK(I)=SUMD:DK(I)=DK(I)*THETA:NEXT J:NEXT I:FOR I=1 TO NB2-1
200: A(I)= A(I)+PE(IS)*DK(I):NEXT I: NEXT IS
201: PRINT TAB(10); "A(I)": PRINT
202: PRINT "CHANGE OF ACTIVE & REACTIVE BUS VOLTAGES"
203: PRINT
204: FOR I=1 TO NB2-1: PRINT A(I),: NEXT I: PRINT
205: INPUT "S(I,J) PRINT REQD. ? YES = 1 NO=0"; PP
206: IF PP = 0  THEN 666
207: PRINT TAB(10);"S(I,J) MATRIX": PRINT
208: FOR I=1 TO NB2-1:FOR J=1 TO NB2-1:PRINT S(I,J),:NEXT J:
          PRINT:PRINT:NEXT I
209: 666 FOR I = 1 TO NB:DELE(I)=A(I):NEXT I
210: FOR I=2 TO NB:DELF(I)=A(NB+I-1):NEXT  I
211: FOR I=1 TO NB:E(I,1)=E(I,1)+DELE(I):NEXT I:ITN=ITN+1:DELF(J1)=0
212: FOR I=2 TO NB:F(I,1)= F(I,1)+ DELF(I): NEXT I
213: DIM EMAG (NB), GAMMA (NB)
214: FOR I=1 TO NB:A=E(I,1)*E(I,1)+ F(I,1)*F(I,1)
215: EMAG(I)=SQR(A):GAMMA(I)= -(180/3.142)*ATN(F(I,1)/E(I,1)):NEXT
216: PRINT TAB(5);"BUS NO. ACTIVE AND REACTIVE VOLTAGES VOLTAGE
          MAGNITUDE ANGL
217: PRINT
218: FOR I=1 TO NB:PRINT TAB(10);I,E(I,1),F(I,1):PRINT
```

```
 219: PRINT TAB(10); EMAG(I),GAMMA(I):PRINT:NEXT I
 220: FOR I=1 TO NB:P(I,1)=P(I,2):Q(I,1)=Q(I,2):NEXT I
 221: GO TO 100
 222: 64 PRINT
 223: PRINT TAB(10);"STATE ESTIMATION RESULTS"
 224: PRINT
 225: PRINT TAB(10); "ACTIVE BUS VOLTAGES": PRINT
 226: FOR I=1 TO NB: PRINT E(I,1), : NEXT I:PRINT
 227: PRINT TAB(10); "REACTIVE BUS VOLTAGES": PRINT
 228: FOR I = 1 TO NB:PRINT F(I,1),: NEXT I: PRINT
 229: STOP
 230: END
NO ERRORS DETECTED
CONSTANT AREA :        16
CODE SIZE :          9022
DATA STMT AREA  :       0
VARIABLE AREA  :       568
A
```

A5.1 OPTIMAL ORDERING BY TINNEY'S SCHEME 2

```
COMPILE OPTILOR $E
MBASIC COMPILER V2.0
  1: LPRINTER WIDTH 80
  2: REM OPTIMAL ORDERING ALGORITHMS TINNEY'S SCHEME 2
  3: INPUT "NO. OF BUSES & NO. OF LINES="; NB, LPQ
  4: DIM AX(NB,NB)
  5: DIM Y(LPQ,7),YBUSR(NB,NB),YBUSI(NB,NB)
  6: INPUT"P-Q R-X YPQR/2-YPQX/2-TR FILE YZ=";Y$
  7: OPEN Y$ AS 1: FOR I=1 TO LPQ:FOR J=1 TO 7
  8: READ #1;Y(I,J):NEXT J,I: CLOSE 1
  9: INPUT"LINE DATA PRINT REQRD. YES=1 NO = 0"; PP
 10: IF PP=0 THEN 501
 11: PRINT "LINE DATA": PRINT
 12: FOR I=1 TO LPQ:PRINT "LINE="; I: PRINT
 13: FOR J=1 TO 7:PRINT Y(I,J);:NEXT J:PRINT: NEXT I
 14: 501 FOR I=1 TO LPQ:Z=Y(I,3)*Y(I,3)+Y(I,4)*Y(I,4)
 15: Y(I,3)=Y(I,3)/Z:Y(I,4)=-Y(I,4)/Z: NEXT I
 16: INPUT "LINE ADMITTANCE FILE YA.I"; A$
 17: CREATE A$ AS 1: FOR I=1 TO LPQ:FOR J=1 TO 7
 18: PRINT #1 , Y(I,J): NEXT J,I: CLOSE 1
 19: FOR J=1 TO NB:FOR I=1 TO NB: YBUSR(I,J) = 0
 20: YBUSI(I,J) = 0: NEXT I,J
 21: FOR I=1 TO NB: SUMA=0: SUMB=0: FOR J=1 TO LPQ
 22: IF ( Y( J,1) - I )<>0  THEN 10
```

```
23: IF Y(J,7)<>1  THEN 10
24: SUMA=SUMA+Y(J,3)+Y(J,5): SUMB=SUMB+Y(J,4)+Y(J,6)
25: 10 NEXT J
26: FOR J = 1 TO LPQ
27: IF ( Y(J,2) -I)<>0  THEN  11
28: IF Y( J,7 )<>1  THEN  11
29: SUMA=SUMA+Y(J,3)+Y(J,5): SUMB=SUMB+ Y(J,4)+Y(J,6)
30: 11 NEXT  J
31: YBUSR (I,I)= SUMA:YBUSI(I,I)= SUMB
32: IF I = NB THEN  9
33: FOR  J=1 TO  LPQ
34: IF ( Y(J,1) -I)<>0  THEN 21
35: IF Y( J,7 )<>1  THEN  21
36: II=I+1: FOR K=II TO NB
37: IF ( Y(J,2) - K)<>0  THEN  41
38: YBUSR (I,K)= -Y(J,3): YBUSR(K,I) = YBUSR(I,K)
39: YBUSI (I,K)= -Y(J,4): YBUSI ( K,I)= YBUSI (I,K)
40: 41 NEXT  K
41: 21 NEXT  J
42: 9  NEXT  I
43: FOR I=1 TO NB: FOR  J=1 TO  LPQ
44: IF( Y(J,1) -I)<>0  THEN 101
45: IF Y(J,7) = 1 THEN  101
46: YBUSI (I,I)= YBUSI (I,I)+Y(J,4)/(Y(J,7) * Y( J,7))
47: 101 NEXT  J
48: NEXT  I
```

```
49: FOR I=1 TO NB: FOR J=1  TO  LPQ
50: IF (Y(J,2) -I)<>0  THEN  104
51: IF Y(J,7) =1  THEN  104
52: YBUSI( I,I ) = YBUSI( I,I ) + Y( J,4 )
53: 104  NEXT  J
54: NEXT  I
55: FOR  I=1 TO NB: FOR J=1  TO LPQ
56: IF ( Y( J,1 ) - I)<>0  THEN  102
57: IF Y( J,7 ) = 1 THEN  102
58: FOR K = 1 TO NB
59: IF (Y( J,2 ) -K)<>0  THEN  103
60: YBUSI (I,K)= -Y (J,4)/Y (J,7):YBUSI (K,I)= YBUSI (I,K)
61: 103 NEXT  K
62: 102 NEXT  J
63: NEXT  I
64: INPUT "IS ANY SHUNT CAP. YES = 1, NO= 0 ="; B
65: IF B = 0  THEN  106
66: 106  INPUT "CONNECTED TO WHICH BUS & VALUE=";BN,CS
67: YBUSI (BN,BN) = YBUSI ( BN,BN ) + CS
68: INPUT "ANY OTHER BUS WITH SHUNT YES = 1, NO=0=";BB
69: IF BB=0 THEN 106
70: GO TO 106
71: 106 INPUT "BUS ADMITTANCE MATRIX PRINT? YES=1 NO=0"; PP
72: IF PP = 0 THEN  502
73: PRINT "BUS ADMITTANCE MATRIX": PRINT
74: PRINT "R - COMPONENTS": PRINT
```

```
75: FOR I=1 TO NB : PRINT "ROW=";  I : FOR J=1  TO NB
76: PRINT YBUSR( I,J );  :NEXT J: PRINT : NEXT  I
77: PRINT "J - COMPONENTS": PRINT
78: FOR I=1 TO NB : PRINT "ROW=";I: FOR J=1 TO NB
79: PRINT YBUSI  (I,J);: NEXT J: PRINT : NEXT I: PRINT
80: FOR I=1 TO NB: PRINT "ROW=";I: FOR J=1 TO NB
81: PRINT YBUSI (I,J),: NEXT J: PRINT: NEXT I: PRINT
82: 502 INPUT "YBUSR FILE NAME YR"; A$ : CREATE A$ AS 1
83: FOR I=1 TO NB: FOR J=1 TO NB: PRINT #1;  YBUSR( I,J )
84: NEXT J: NEXT I: CLOSE 1
85: INPUT "YBUSI  FILE NAME YI"; B$:CREATE B$ AS 2:FOR I=1 TO NB
86: FOR J=1 TO NB:PRINT #2; YBSI (I,J):NEXT J,I:CLOSE 2
87: FOR I=1 TO NB:FOR J=1 TO NB:AX(I,J)= YBUSI (I,J):NEXT J,I
88: FOR I=1 TO NB : FOR J=1  TO NB
89: IF AX( I,J ) = 0 THEN 200
90: AX( I,J ) = 1
91: 200 NEXT  J
92: NEXT  I
93: INPUT "FILE NAME AXIJ="; C$
94: CREATE C$ AS 2: FOR I=1 TO NB: FOR J=1 TO NB
95: PRINT #2; AX( I,J ); NEXT J,I: CLOSE 2
96: PRINT "MATRIX AX( I,J )"
97: FOR I=1 TO NB: FOR J=1 TO NB: PRINT AX( I,J );
98: NEXT J:PRINT : NEXT I: PRINT : N=NB
99: DIM ORDER (N),A( N,N ), ADJ (N), TA(N), OFD(N)
100: FOR I=1 TO N: FOR J=1  TO  N
```

```
101: A( I,J ) = AX( I,J ) : NEXT J,I
102: FOR I = 1 TO  N: P = 0 : VA ( I ) = 0
103: FOR J =1  TO  N
104: IF I = J  THEN  610
105: IF A( I,J ) = 0  THEN  610
106: P = P + 1 : ADJ ( P ) = J
107: 610 NEXT  J
108: IF P = 1 THEN  30
109: FOR L = 1 TO P - 1 : U = ADJ ( L )
110: FOR S = L+ 1 TO  P: V = ADJ ( S )
111: IF A( U,V )<>0  THEN  20
112: VA ( I ) = VA ( I ) + 1
113: 20  NEXT  S,L
114: 30  NEXT I: STAGE = 0: VALANCY = 0
115: 35 STAGE = STAGE + 1
116: IF STAGE = N+ 1  THEN 110
117: SMALL = N
118: FOR I = 1  TO  N
119: IF A( I,I ) = 0  THEN  40
120: IF  SMALL  = VA(I) THEN  40
121: SMALL = VA(I)
122: RE = I
123: 40 NEXT  I
124: ORDER ( STAGE ) = RE
125: VALANCY = VALANCY + SMALL
126: FOR I = 1  TO  N
```

```
127: A( I,RE ) = 0
128: NEXT  I
129: P = 0
130: FOR  J = 1  TO  N
131: IF A( RE,J ) = 0  THEN  60
132: P = P+ 1 : ADJ( P ) = J
133: P1 = P
134: 60 NEXT  J
135: IF P = 1  THEN  75
136: FOR  I = 1  TO  N
137: IF A( I,I ) = 0  THEN  70
138: FOR  L = 1  TO P1 - 1: U = ADJ( L )
139: FOR S = L+1  TO P1 : V = ADJ( S )
140: IF I = U THEN 65
141: IF I = V THEN 60
142: IF A ( I,U ) = 0  THEN  65
143: IF A( I,V ) = 0  THEN  60
144: P = P+1: ADJ( P ) = I
145: 60 NEXT  S
146: 65 NEXT  L
147: 70 NEXT  I
148: FOR I = 1 TO P1 - 1 : U = ADJ( I )
149: FOR J = I +1 TO P1: V= ADJ( J )
150: A( U,V ) = 1: A( V,U ) = 1
151: NEXT J,I
152: 75 FOR I = 1 TO P: U = ADJ( I)
```

```
153: C = 0 : VA( U ) = 0
154: FOR J = 1  TO  N
155: IF U = J  THEN  80
156: IF A( U,J ) = 0  THEN  80
157: C = C + 1 : OFD( C) = J
158: 80  NEXT  J
159: IF  C = 1 THEN  100
160: FOR L = 1  TO  C - 1: K = OFD( L )
161: FOR S = L+ 1  TO C:M = OFD( S )
162: IF A( K,M )  0  THEN  90
163: VA( U ) = VA( U ) + 1
164: 90 NEXT S,L
165: NEXT I
166: 100 GO TO 35
167: 110 PRINT "ORDER"
168: FOR I = 1  TO  N
169: PRINT ORDER ( I ) ;
170: NEXT  I
171: PRINT
172: INPUT "ORDERED SEQUENCE FILE ORD.I"; F$
173: CREATE F$ AS 6: FOR I = 1 TO NB:PRINT #6; ORDER ( I )
174: NEXT I : CLOSE 6
175: PRINT "TOTAL VALANCY= "; VALANCY
176: PRINT
177: DIM OYBUSR (NB,NB), OYBUSI ( NB, NB )
178: INPUT "ORDERED BUS PRINT REQRD. YES=1 NO=0";PP
```

```
179: IF PP = 0  THEN  503
180: FOR I=1 TO NB:NN=ORDER(I):FOR J=1 TO NB
181: MM=ORDER(J):GYBUSR(I,J)=YBUSR(NN,MM):GYBUSI(I,J)=YBUSI(NN,MM)
182: NEXT  J,I
183: PRINT "ORDERED R - BUS": PRINT:FOR I=1 TO NB
184: PRINT "ROW=";I:PRINT:FOR J=1 TO NB:PRINT GYBUSR(I,J);:NEXT J
185: PRINT : NEXT  I
186: PRINT "J-BUS": PRINT : FOR I=1 TO NB
187: PRINT "ROW="; I: PRINT
188: FOR J=1 TO NB:PRINT GYBUSI(I,J);:NEXT J:PRINT:NEXT I
189: INPUT "ORDERED BUS FILE REQRD. ? YES = 1 NO=0";PB
190: IF PB = 0  THEN  503
191: INPUT "GYBUSR FILE NAME = GXR"; C$
192: CREATE C$ AS 3:FOR I=1 TO NB:FOR J=1 TO NB
193: PRINT #3; GYBUSR( I,J ): NEXT J,I: CLOSE 3
194: INPUT "GYBUSI FILE NAME = GXI"; D$
195: CREATE D$ AS 4: FOR I = 1  TO  NB
196: FOR J=1 TO NB:PRINT #4; GYBUSI( I,J ): NEXT J,I: CLOSE 4
197: 503 END
NO ERRORS DETECTED
CONSTANT AREA :        8
CODE SIZE  :        5996
DATA STMT AREA :       0
VARIABLE AREA  :     384
A
```

A5.2 OPTIMAL ORDERING OF NODES BY DYNAMIC PROGRAMMING ALGORITHM

```
COMPILE DORDEL $E
HIBASIC COMPILER V2.0
   1: REM DYNAMIC PROGRAMMING ORDERING ALGORITHM
   2: LPRINTER WIDTH 80
   3: INPUT "NO OF BUSES"; N
   4: DIM AX( N,N )
   5: INPUT "AXIJ FILE NAME"; A$
   6: OPEN A$ AS 1
   7: FOR I = 1 TO N:FOR  J=1  TO  N
   8: READ #1; AX( I,J ):NEXT J,I: CLOSE  1
   9: DIM ORDER(N),A(N,N),NODE(N),VALANCY(N),CV(N,N),LROW(N),OPD(N)
  10: FOR STAGE = 1  TO  N
  11: IF STAGE = 1  THEN  120
  12: FOR ST = 1 TO N: CHECK = 0
  13: IF CV(ST,STAGE - 1)= -N THEN  106
  14: NODE (STAGE) =  ST: TOV=N * N: PROW = 0
  15: FOR PR = 1  TO  N
  16: IF PR = NODE (STAGE) THEN 100
  17: NODE ( STAGE - 1 ) = PR
  18: IF STAGE = 2 THEN 10
  19: IF CV( PR,STAGE -1)= -N THEN 100
  20: FOR K = 2 TO STAGE - 1:V = STAGE - K
  21: U = NODE ( V + 1 )
  22: NOD = NODE (V+1) - CV( U,V+1 )
  23: IF NOD = NODE (STAGE) THEN 100
```

```
24: NODE (V) = NOD: NEXT K
25: 10 FOR J=1 TO N: FOR I=1 TO N:A( I,J ) = AX( I,J )
26: NEXT I,J: VA=0: FOR K=1 TO STAGE
27: R= NODE(K):P=0:FOR C=1 TO N: FOR I=1 TO K
28: RE = NODE ( I )
29: IF C = RE THEN 20
30: NEXT I
31: IF A ( R,C ) = 0 THEN 20
32: P = P+1 : OFD(P) = C
33: 20 NEXT C
34: IF P<=1 THEN 35
35: FOR L = 1 TO P - 1: FOR M = L+1 TO P
36: I = OFD( L ): J = OFD ( M )
37: IF A ( I,J ) <>0 THEN 30
38: VA = VA+1:A( I,J ) = 1:A( J,I ) = 1
39: 30 NEXT M,L
40: 35 NEXT K
41: CHECK = CHECK + 1
42: IF TOV = VA THEN 100
43: TOV = VA : PROW = PR
44: 100 NEXT PR
45: IF CHECK<>0 THEN 106
46: VALANCY (ST)= -1:CV(ST,STAGE) = -N: GO TO 110
47: 106 VALANCY (ST)=TOV:CV(ST,STAGE)=ST-PROW
48: GO TO 110
49: 106 CV( ST, STAGE ) = -N
```

```
 50: 110 NEXT ST:PRINT "STAGE"; STAGE:PRINT: FOR I=1 TO N
 51: PRINT "VALANCY"; VALANCY (I);"CV"; CV(I,STAGE)
 52: NEXT I: PRINT
 53: 120 NEXT STAGE
 54: SMALL = N * N: P=0
 55: FOR I=1  TO  N
 56: IF VALANCY ( I ) = -1  THEN  140
 57: IF SMALL <=  VALANCY ( I ) THEN  140
 58: SMALL = VALANCY ( I )
 59: 140 NEXT I: FOR I = 1  TO  N
 60: IF  SMALL <>VALANCY ( I ) THEN  150
 61: P = P+1: LROW ( P ) = I
 62: 150 NEXT  I
 63: FOR I = 1  TO  P
 64: ORDER ( N ) = LROW( I ):  FOR J = 1  TO  N - 1
 65: C = N -J:RO=ORDER(C+1):ORDER(C)=ORDER(C+1) - CV(RO,C+1)
 66: NEXT J: PRINT "ORDER";I: PRINT
 67: FOR J = 1  TO N: PRINT ORDER ( J );: NEXT  J
 68: PRINT : NEXT I: PRINT
 69: END
NO ERRORS DETECTED
CONSTANT AREA :        8
CODE SIZE  :        1819
DATA STMT AREA  :      0
VARIABLE AREA   :    248
A
```

A6.3 GAUSS SEIDEL LOAD FLOW WITH OPTIMALLY ORDERED NODES

```
COMPILE OPTLF $E
MBASIC COMPILER V2.0
   1: LPRINTER WIDTH 80
   2: INPUT "NO. OF BUSES & NO. OF LINES"; NB,LPQ
   3: DIM ORDER (NB),Y(LPQ,9),YBUSR(NB,NB),YBUSI (NB,NB)
   4: INPUT "YBUSR FILE NAME = GYR.I";A$
   5: OPEN A$ AS 1: FOR I=1 TO NB: FOR J=1 TO NB
   6: READ #1; YBUSR(I,J): NEXT J,I: CLOSE 1
   7: INPUT "YBUSI FILE NAME = GYI.I"; B$
   8: OPEN B$ AS 2: FOR I=1 TO NB: FOR J=1 TO NB
   9: READ #2; YBUSI (I,J): NEXT J,I: CLOSE 2
  10: INPUT "YA.I FILE NAME"; A$
  11: OPEN A$ AS 1: FOR I=1 TO LPQ: FOR J=1 TO 7
  12: READ #1; Y(I,J):  NEXT J,I: CLOSE 1
  13: INPUT "ORDER FILE NAME ORD.I"; A$: OPEN A$ AS 1
  14: FOR I=1 TO NB:READ #1; ORDER (I): NEXT I: CLOSE 1
  15: FOR I=1 TO LPQ: FOR J=1 TO NB
  16: IF Y(I,1)<>ORDER(J) THEN 420
  17: JJ=Y(I,1):Y(I,1)= J:Y(I,8)= JJ:GO TO 7
  18: 420 NEXT J
  19: 7 FOR J=1  TO NB
  20: IF Y(I,8)<>ORDER(J) THEN 421
```

```
21: JJ=Y(I,2): Y(I,8)= J:Y(I,9)= JJ: GO TO 8
22: 421 NEXT J
23: 8 NEXT I
24: INPUT "LINE ADMITTANCE PRINT ? YES = 1 NO = 0"; BB
25: IF BB = 0 THEN 422
26: PRINT "LINE ADMITTANCE": PRINT
27: PRINT "GP GQ YR YI LCR LCI TTC P Q": PRINT
28: FOR I= 1 TO LPQ: FOR J=1 TO 9: PRINT Y(I,J);
29: NEXT J: PRINT: NEXT I
30: 422 INPUT "SLACK BUS = J1="; J1
31: DIM NBUSR(NB,2), NBUSI(NB,2), P(NB,4),PA(NB),PR(NB)
32: DIM LPR(NB, LPI(NB), KLP(NB), KLI(NB)
33: DIM KLPR(NB,NB),KLPI(NB,NB),NBUSRA(NB,2),NBUSIA(NB,2)
34: FOR I=1 TO NB
35: IF I=J1 THEN 54
36: NBUSR(I,1) = 1: NBUSI( I,1 ) = 0
37: NBUSRA(I,1)= 1: NBUSIA(I,1) = 0
38: 54 NEXT I
39: INPUT "SLACK BUS ACTIVE & REACTIVE VOLTAGES="; VSVR, VSVI
40: NBUSR(J1,1)= VSVR: NBUSI( J1,1 ) = VSVI
41: PRINT "PARAMETERS OF VOLTAGE EQUATIONS": PRINT
42: INPUT "GENERATION & LOAD POWER FILE = GL ="; B$
43: OPEN B$ AS 2: FOR I=1 TO NB: FOR J=1 TO 4
44: READ #2; P(I,J): NEXT J,I: CLOSE 2
45: INPUT "G - L PRINT ? YES = 1 NO = 0"; BB
```

```
46: IF BB = 0 THEN 2
47: PRINT"GENERATION-ACTIVE & REACTIVE:LOAD-ACTIVE & REACTIVE"
48: FOR I=1 TO NB: FOR J=1 TO 4: PRINT P(I,J),: NEXT J,I
49: 2 INPUT "MVA BASE"; BASE
50: FOR I = 1  TO NB
51: PA(I)=(P(I,1)-P(I,3))/BASE: PR(I)= -(P(I,2)-P(I,4))/BASE
52: D=YBUSR(I,I) * YBUSR(I,I)+ YBUSI(I,I) * YBUSI(I,I)
53: LPR(I)= YBUSR(I,I)/D:LPI(I)= -YBUSI(I,I)/D
54: NEXT  I
55: FOR I = 1  TO NB
56: IF I = J1  THEN 57
57: KLR(I)= PA(I) * LPR(I) + PR(I) * LPI(I)
58: KLI(I)= PA(I) * LPI(I) - PR(I) * LPR(I)
59: 57 NEXT I
60: KLR( J1 ) = 0: KLI( J1 ) = 0
61: INPUT " BUS PAR. PRINT ? YES = 1 NO = 0"; BB
62: IF BB = 0  THEN  3
63: PRINT "BUS PARAMETERS": PRINT
64: FOR I=1 TO NB: PRINT "BUS=" ; ORDER (I): PRINT
65: PRINT "KLR(I),KLI(I)=";KLR(I),KLI(I): PRINT : NEXT I
66: 3 FOR I = 1 TO NB: FOR J = 1  TO NB
67: KLPR( I,J ) = 0: KLPI( I,J ) = 0: NEXT J,I
68: FOR I = 1  TO NB
69: FOR J = 1  TO NB
70: IF J = I  THEN  60
```

```
71: KLPR(I,J) =YBUSR(I,J)* LPR(I) -YBUSI(I,J)* LPI(I)
72: KLPI(I,J) =YBUSR(I,J)* LPI(I) +YBUSI(I,J)* LPR(I)
73: GO NEXT J
74: NEXT  I
75: INPUT "LINE PAR. PRINT ? YES = 1 NO =0";BB
76: IF BB = 0  THEN  4
77: PRINT " LINE PARAMETERS": PRINT
78: FOR I=1 TO NB: PRINT "ROW=";ORDER(I): FOR J=1 TO NB
79: PRINT KLPR(I,J), KLPI(I,J) : NEXT J
80: PRINT : NEXT I: PRINT
81: 4 DIM BLR(NB,2),BLI(NB,2),BBUSCIA(NB,2)
82: DIM BLRA(NB,2),BLIA(NB,2)
83: REM BBUSCIA STANDS FOR CONJUGATE FOR VOLTAGE
84: K=1: FOR I=1 TO NB:BBUSRA(I,1)= BBUSR(I,1)
85: BBUSIA(I,1) = BBUSI(I,1): NEXT I
86: INPUT "ACCLERATION FACTOR ALPHA=";  ALPHA
87: INPUT "TOLERANCE = TL = ";  TL
88: INPUT "HOW MANY BUSES ARE VOLTAGE CONTROLLED ?";  L
89: DIM PV( L,4 )
90: INPUT "VCB.I FILE NAME";  D$
91: OPEN D$ AS 6: FOR I=1 TO L: FOR J=1 TO 4
92: READ #6; PV(I,J):NEXT J,I:CLOSE  6 : FOR I=1  TO L
93: LL = PV(I,1): BBUSR( LL,K )= PV( I,2 )
94: BBUSI( LL,K ) = 0 : NEXT  I
95: GO BBUSRA ( J1,K+1) = BBUSRA( J1,K )
96: BBUSIA (J1, K+1) = BBUSIA ( J1, K )
```

```
 97: MBUSR ( J1,K+1)= MBUSR ( J1,K )
 98: MBUSI ( J1,K+1 )= MBUSI ( J1,K )
 99: GOSUB 107
100: FOR I = 1  TO  NB
101: IF I = J1  THEN  61
102: DLRA ( I, 1 ) = 0: DLIA ( I,1 ) = 0
103: DLR ( I,1 ) = 0 : DLI ( I,1 ) = 0
104: 61 NEXT  I
105: FOR  I = 1  TO  NB
106: IF  I = J1  THEN  90
107: MBUSCIA ( I,K ) = -1.0* MBUSIA ( I, K )
108: 90  NEXT  I
109: FOR I = 1  TO  NB
110: IF  I = J1  THEN  62
111: D=MBUSRA(I,K)* MBUSRA(I,K)+ MBUSCIA(I,K)* MBUSCIA(I,K)
112: ER1 =(KLR(I)* MBUSRA(I,K)+ KLI(I)* MBUSCIA(I,K))/D
113: MI1 =(KLI(I)* MBUSRA(I,K)- KLR(I)* MBUSCIA(I,K))/D
114: IF  I=1  THEN  64
115: ER2 = 0: MI2=0: FOR  J=1  TO  I-1
116: ER2=ER2+KLPR(I,J)* MBUSRA(J,K+1)-KLPI(I,J)* MBUSIA(J,K+1)
117: MI2=MI2+KLPI(I,J)*MBUSRA(J,K+1)+KLPR(I,J)* MBUSIA(J,K+1)
118: NEXT  J
119: 64 ER3=0
120: MI3 = 0
121: IF I=NB THEN 700
122: FOR  J = I+ 1  TO  NB
```

```
123: ER3=ER3+KLPR(I,J) * MBUSRA(J,K) - KLPI(I,J) * MBUSIA(J,K)
124: MI3=MI3+KLPI(I,J) * MBUSRA(J,K)+ KLPR(I,J) * MBUSIA(J,K)
125: NEXT J
126: 700 IF I = 1 THEN 111
127: GO TO 112
128: 111 ER2 = 0
129: MI2 = 0
130: 112 MBUSR( I,K+1 ) = ER1 - ER2 - ER3
131: MBUSI ( I,K+1 ) = MI1 - MI2 - MI3
132: MBUSRA (I,K+1)=MBUSRA(I,K)+ALPHA*(MBUSR(I,K+1)-MBUSRA(I,K))
133: MBUSIA (I,K+1)=MBUSIA(I,K)+ALPHA*(MBUSI(I,K+1)-MBUSIA(I,K))
134: 62 NEXT I
135: PRINT "ITERATION ITN="; ITN: PRINT
136: INPUT "ITERATION  PRINT REQD. ? YES = 1 NO = 0"; BB
137: IF BB = 0  THEN  5
138: PRINT "BUS VOLTAGES & ACCELERATED BUS VOLTAGES": PRINT
139: FOR I=1 TO NB:PRINT "BUS NO.="; ORDER (I): PRINT
140: PRINT MBUSR(I,K+1),MBUSI(I,K+1),MBUSRA(I,K+1),MBUSIA(I,K+1)
141: NEXT I : PRINT
142: 5 FOR I = 1 TO NB
143: IF I = J1  THEN  66
144: DLR( I,K+1) = MBUSR (I,K+1) - MBUSR( I,K )
145: DLRA(I,K+1) = MBUSRA(I,K+1) - MBUSRA(I,K)
146: DLI (I,K+1) = MBUSI ( I,K+1) - MBUSI ( I,K )
147: DLIA(I,K+1) = MBUSIA(I,K+1) - MBUSIA(I,K)
```

```
148: 66 NEXT  I
149: DMAXR = DLR( 1,K+1 ): DMAXI = DLI ( 1,K+1 )
150: IF DMAXR < 0  THEN  701
151: GO TO 702
152: 701 DMAXR = - DMAXR
153: 702 IF DMAXI < 0  THEN 703
154: GO TO 704
155: 703 DMAXI = - DMAXI
156: 704 FOR I = 2  TO NB
157: IF  I = J1  THEN  70
158: IF  DLR ( I,K+1 ) < 0  THEN  706
159: DEL = DLR ( I,K+1 )
160: GO TO 706
161: 706 DEL = - DLR ( I,K+1 )
162: 706 IF ( DMAXR - DEL ) > 0  THEN  70
163: DMAXR = DEL
164: 70 NEXT  I
165: FOR I = 2 TO NB
166: IF I = J1  THEN  73
167: IF DLI ( I,K+1 ) < 0  THEN  707
168: DELL = DLI  ( I,K+1 )
169: GO TO 708
170: 707 DELL = - DLI  ( I,K+ 1 )
171: 708 IF ( DMAXI - DELL ) > 0  THEN  73
172: DMAXI = DELL
```

```
173: 73 NEXT I
174: IF ( DMAXR - DMAXI )> 0 THEN 68
175: DL = DMAXI
176: GO TO 71
177: 68 DL = DMAXR
178: 71 IF ( DL - TL )< 0 THEN 75
179: ITN = ITN + 1
180: FOR I = 1 TO NB
181: IF I = J1 THEN 88
182: NBUSR(I,K)=NBUSR(I,K+1): NBUSI(I,K)= NBUSI(I,K+1)
183: NBUSRA(I,K)=NBUSRA(I,K+1): NBUSIA(I,K)=NBUSIA(I,K+1)
184: 88 NEXT I
185: GO TO 80
186: 75 K1 = K+1
187: PRINT "LINEFLOWS & POWER SL BUS": PRINT: PRINT
188: DIM PQ( LPQ,6), QP( LPQ,6 )
189: FOR K = 1 TO NB : FOR I = 1 TO LPQ
190: IF Y( I,1 )<>K THEN 201
191: J = Y ( I,2 )
192: P1=NBUSRA(K,K1) * (NBUSRA(K,K1) - NBUSRA(J,K1))
193: P2= - NBUSCIA(K,K1) *(NBUSIA(K,K1) - NBUSIA( J,K1 ))
194: P3= NBUSRA(K,K1) * (NBUSIA(K,K1) - NBUSIA(J,K1))
195: P4= NBUSCIA(K,K1) *(NBUSRA(K,K1) - NBUSRA(J,K1))
196: P1 = P1 + P2 : P2 = P3+P4
197: PW = ( P1 * Y( I,3) - P2 * Y( I,4 )) * BASE
198: Q3=NBUSRA(K,K1) *NBUSRA(K,K1): Q4=NBUSIA(K,K1) *NBUSIA(K,K1)
```

```
199: Q3 = ( Q3 + Q4 ) * Y ( I,6 )
200: Q =(P1* Y(I,4)+P2 * Y(I,3) +Q3) * BASE
201: PQ(I,5)=X(I,8):PQ(I,6)=X(I,9):PQ(I,1)=X(I,1):PQ(I,2)=X(I,2)
202: PQ(I,3)= PW : PQ (I,4) = - Q
203: 201 NEXT I
204: NEXT K
205: PRINT " LINE POWER FLOW "
206: PRINT
207: FOR I=1 TO LPQ: PRINT "ROW=";I:PRINT
208: FOR K=1 TO 6:PRINT PQ(I,K),:NEXT K:PRINT:NEXT I:PRINT
209: PRINT "LINE POWER FLOW REVERSED" : PRINT
210: FOR K=1 TO NB: FOR I=1 TO LPQ
211: IF Y(I,2)<>K THEN  301
212: J = Y ( I,1 )
213: P1=EBUSRA(K,K1)*( EBUSRA(K,K1) - EBUSRA(J,K1))
214: P2= - EBUSCIA(K,K1) *(EBUSIA(K,K1)- EBUSIA(J,K1))
215: P3= EBUSRA(K,K1) *(EBUSIA(K,K1) - EBUSIA(J,K1))
216: P4=EBUSCIA(K,K1) *(EBUSRA(K,K1) - EBUSRA(J,K1))
217: P1=P1+P2:P2=P3+P4
218: PW=(P1 * Y(I,3) - P2 * Y(I,4)) * BASE
219: Q3=EBUSRA(K,K1)* EBUSRA(K,K1):Q4=EBUSIA(K,K1)* EBUSIA(K,K1)
220: Q3= (Q3+Q4) * Y(I,6)
221: Q=(P1*Y(I,4)+P2 * Y(I,3)+ Q3) * BASE
222: QP(I,1)=X(I,2):QP(I,2)=X(I,1):QP(I,3)=PW: QP(I,4)= - Q
223: QP(I,5) = Y(I,9):QP(I,6)= Y(I,8)
224: 301 NEXT I
```

```
225: NEXT  K
226: FOR I = 1  TO LPQ
227: PRINT "ROW=";I:PRINT: FOR K=1 TO 6
228: PRINT QP(I,k),: NEXT K:PRINT: NEXT I: PRINT
229: SUMA=0; SUMB=0
230: FOR I=1  TO LPQ
231: IF PQ( I,1 )<>J1  THEN  401
232: SUMA=SUMA+PQ(I,3): SUMB=SUMB+PQ(I,4)
233: 401 NEXT  I
234: PRINT "SLACK BUS ACTIVE POWER ="; SUMA: PRINT
235: PRINT "SLACK BUS REACTIVE POWER= "; SUMB : PRINT
236: DIM EMAG(NB),GAMMA (NB)
237: FOR I=1 TO NB:A=EBUSRA(I,K1)*EBUSRA(I,K1)+EBUSIA(I,K1)*EBUSIA(I,K1)
238: EMAG(I) = SQR(A)
239: GAMMA(I)=-1.0*(180/3.142)* ATN(EBUSIA(I,K1)/EBUSRA(I,K1))
240: PRINT "BUS NO.  VOLTAGE MAGNITUDE ANGLE": PRINT
241: PRINT ORDER (I), EMAG(I), GAMMA (I)
242: PRINT : NEXT  I
243: GO TO 113
244: 107 REM COMPUTATION OF VOLTAGE CONTROLLED BUSES
245: FOR II=1 TO L:LL=PV(II,1):QMIN=PV(II,3)/BASE
246: QMAX=PV(II,4)/BASE: VCELL=PV(II,2)
247: ANGLLL=ATN(EBUSI(LL,K)/EBUSR( LL,K ))
248: TETA= ANGLLL* ( 180/3.142 )
249: EBUSR(LL,K+1)=VCELL*COS(ANGLLL):EBUSI(LL,K+1)=VCELL*SIN(ANGLLL)
250: REM CALCULATION OF REACTIVE POWER AT BUS LL
```

```
251: QR=(RBUSR(LL,K+1) * RBUSR(LL,K+1) * YBUSI (LL,LL)
252: QR=QR+(RBUSI (LL,K+1) * RBUSI (LL,K+1)) * YBUSI (LL,LL)
253: SUM=0: FOR I=1  TO NB
254: IF  I = LL  THEN  108
255: A=RBUSR(I,K) * YBUSR(LL,I)+RBUSI (I,K) * YBUSI (LL,I)
256: A= A  * RBUSI  ( LL, K+1 )
257: B=RBUSI (I,K) * YBUSR(LL,I)-RBUSR(I,K) * YBUSI (LL,I)
258: B= -B * RBUSR( LL,K+1 )
259: SUM = SUM + A + B
260: 108 NEXT  I
261: QR= QR + SUM
262: A = ABS ( QR )
263: IF A  = QRMAX  THEN 109
264: A = QRMAX : GO TO 1111
265: 109 IF A  = QRMIN THEN  110
266: A = QRMIN : GO TO 1111
267: 110 RBUSR (LL,K)= RBUSR(LL,K+1): RBUSI (LL,K)=RBUSI (LL,K+1)
268: REM RECOMPUTE KLR & KLI
269: 1111 QR = A
270: D=YBUSR(LL,LL) * YBUSR(LL,LL)+YBUSI (LL,LL) * YBUSI (LL,LL)
271: LPR(LL) = YBUSR(LL,LL)/D: LPI (LL)= -YBUSI (LL,LL)/D
272: KLR(LL) = PA(LL) * LPR(LL)+ QR * LPI (LL)
273: KLI (LL)= PA(LL)  * LPI (LL) - QR * LPR(LL)
```

```
  274: NEXT  II
  276: RETURN
  276: 113  STOP
  277: END
NO ERRORS DETECTED
CONSTANT AREA :        24
CODE SIZE  :         3653
DATA STMT AREA  :       0
VARIABLE AREA   :     632
A
```