REFERENCE

1. Aly Q gncl Ashour I. Thermodynamic Modelling for Oleochemical Industry, Europpean Symopoium on Computer AidedProcess Engineering-2, pp 195-202
2. Dac R. Yang, Kurt V. Waller, Dale E. Sebory and Duncan A. Mellichamp, Dynamic Structural Tranformation for Distillation Control Configulation, AIChE J., vol 36, no. 9, pp 1391-1402, 1990
3. Edmister, W.C., Hydrocarbon Processing ,vol 47, no. 10, pp 145-149, 1968
4. Ernest J. Henley, J.D. Seader, Equilibrium-Stage Separation Operation in Chemical Engineering, John Wiley & sons Inc., New York, 1981
5. Hasan Orbey and J.H. Vera, Correlation for the Third Virial Coefficient Using Tc, Pc and ω As Parameters, AIChE J., vol 29, no. 1, pp 107-113,
6. Hendric C. Van Nees and Michael M. Abbott, Classical Thermodynamics of Nonelectrolyte Solutions, Mc Graw-Hill Book Company, USA, 1984
7. J. Perregard, Model simplification and Reduction for Simulation and Optimization of Chemical Processes, Computer Chem. Engng., vol 17, no. 516, pp 465-483,1993
8. L. Uric, S. Bottini, E.A. Brigde and J.RA Romarnoli, Thermodynamic Tuning in Separation process Simulation and Design, Computer chem. Engng., vol 15 no. 7, 1991
9. Marahiro Kinoshita, Simple Model for Dynamic Simulation of Stage Separation Processes with very Volatile Component, AIChE J., vol 32, no. 5, pp 872-874, 1986
10. Nelson W.L., Petroleum Refinery Engineering, Mc Graw-Hill Book Company, USA, 1982

11. Peng, D.Y., and D.B. Robinson, A new Two-constant Equation of state, Ind. Eng. Chem. Fundam, vol 15, pp 59-64, 1976

12. R. E. Fitymouis, Improving Distillation Column Design Usig Thermodnamic Aailability Analysis, AIChE J., vol 26, no. 2, pp 265-273, 1980

13. Roberto De Santis and Biagio Grande, An Equation for Predicting Third Virial Coefficent of Non polar Gases, AIChE J., vol 25, no. 6, pp 931-938, 1979

14. Robert H. Perry and Don Green, Perry's Chemical Engineers, Handbook, sixth edition, Mcgrow-Hill International Editions, 1984

15. Tomio Umeda, Kayuo Niida and Katsuo Shiroko, Thrmodynamic Appoach to Heat Integration in Distillaion Systems, AIChE J.,vol 29, no. 3, pp 423-429, 1979

16. V. R. Dhole and B. Linnhoff, Distillation Column Targets, Computer chem Engng, vol 17, no. 516, pp 549-560, 1993

17. West, E.w., and J. H. Earbar, An Evaluation of Four Methods of Predicting Thermodynamic Properties of Light Hydrocarbon System, Paper presented at the 52 nd Annual Meeting of NPGA, Dallas, Texas, March 26-28, 1973

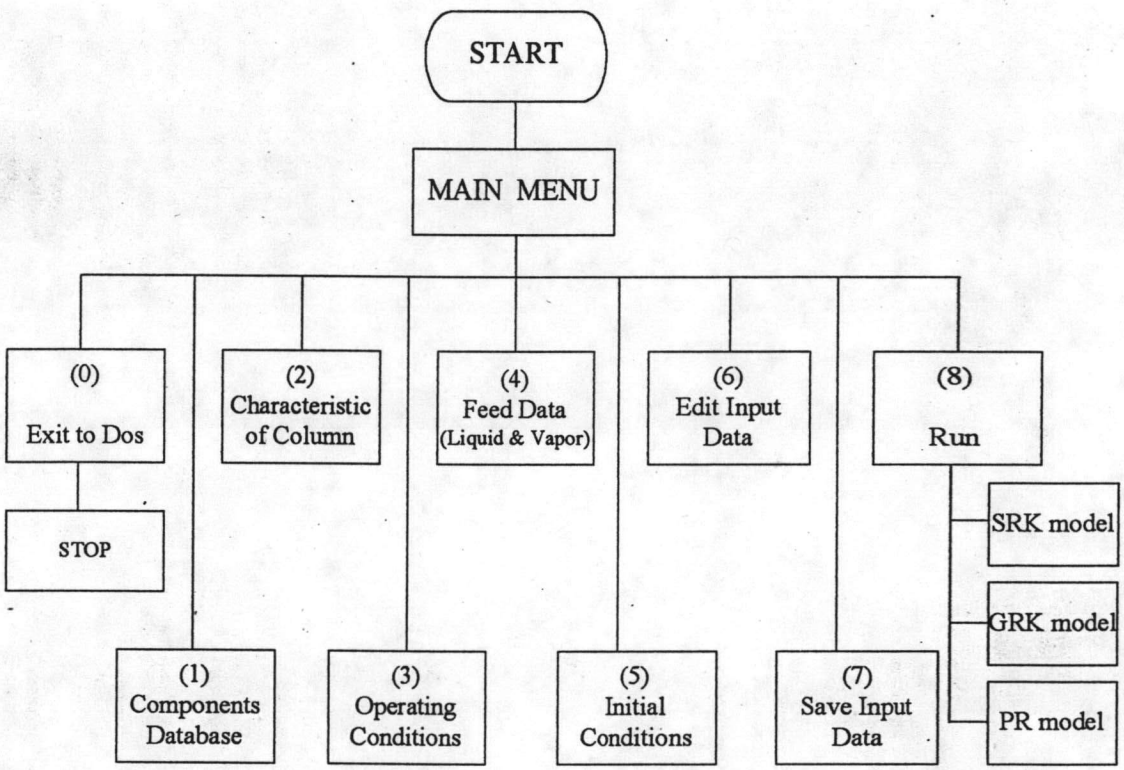
18. William L. Luyben, Process Modeling, Simulation and Control for Chemical Engineers, Mc Graw-Hill publishing company, Second Edition,1990

19. Yuji Naka, Masayuki Terashita and Tekuichiro Takamatsu, A Thermodynamic Approch to Multicomponent Distillation System Synthesis, AIChE J., vol 28, no. 5, pp 812-820, 1982

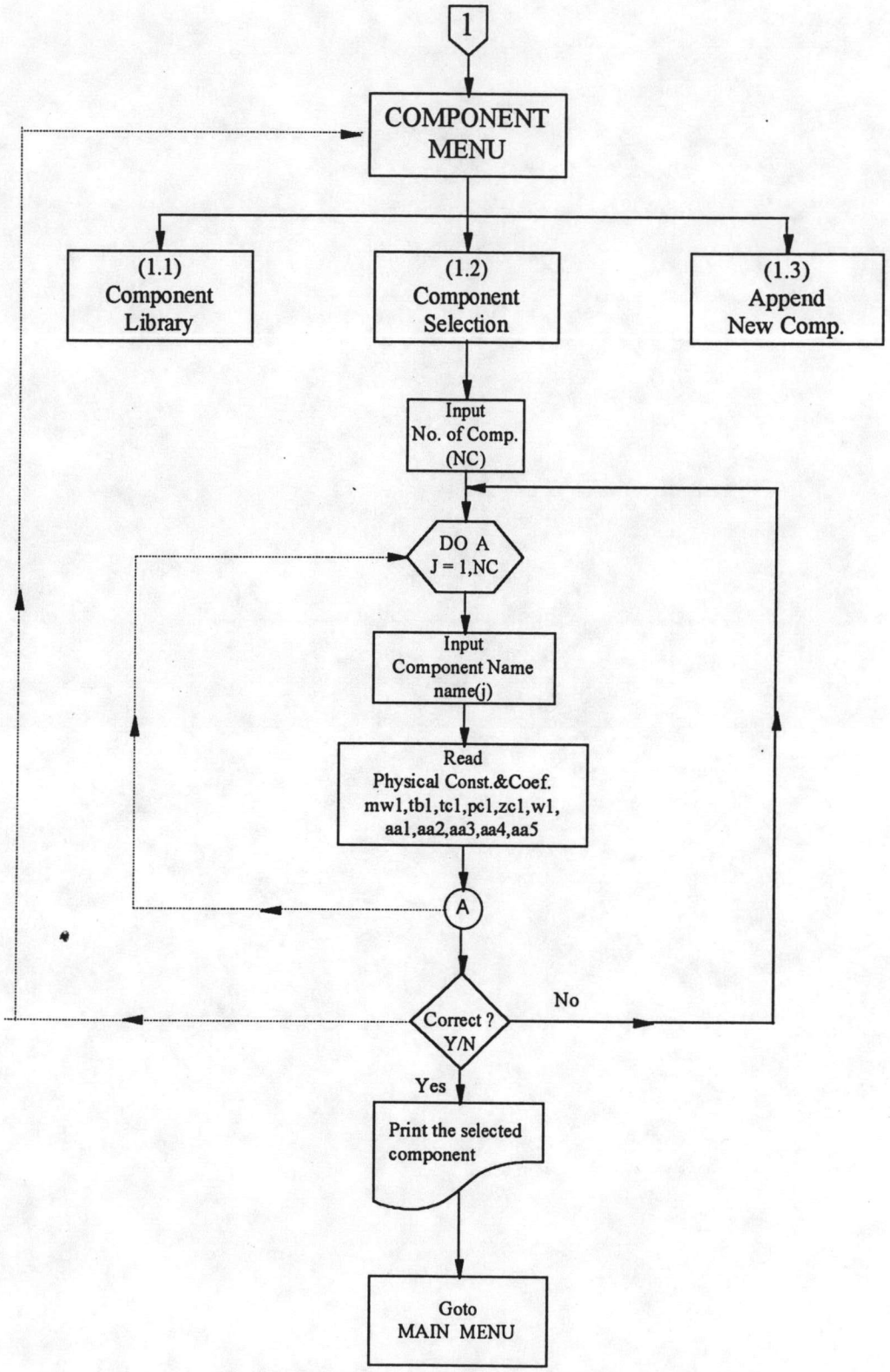
20. Z. T. Fidkowski, M. F. Doherty, and M. F. Malone, Feasibility of Separations for Distillation of Nonideal Ternary Mixtures, AIChE J., vol 39, no. 8, pp 1303-1321, 1993

Appendix A

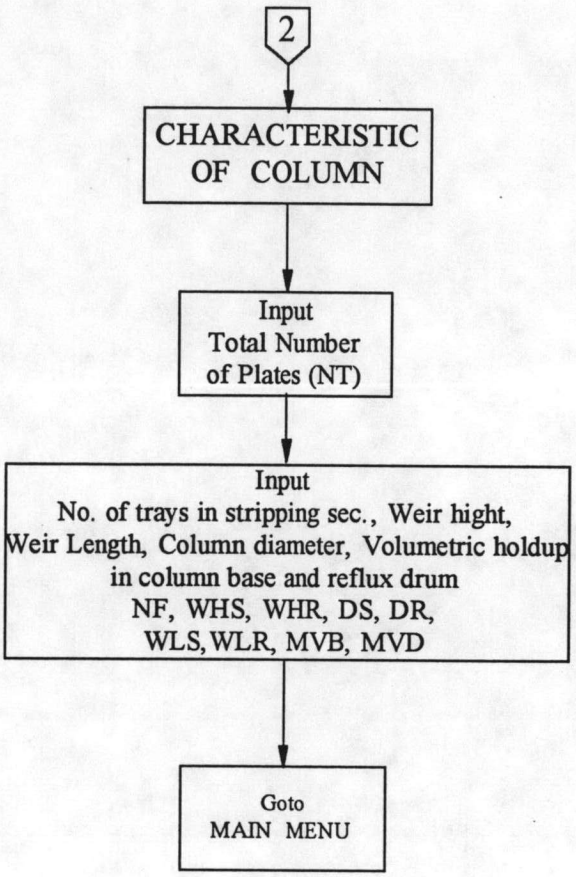
Computer Program Flow Chart



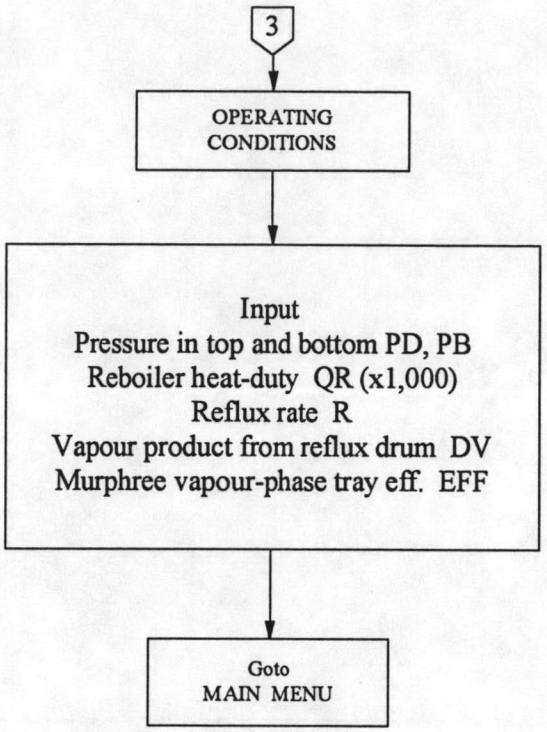
A) Program Structure



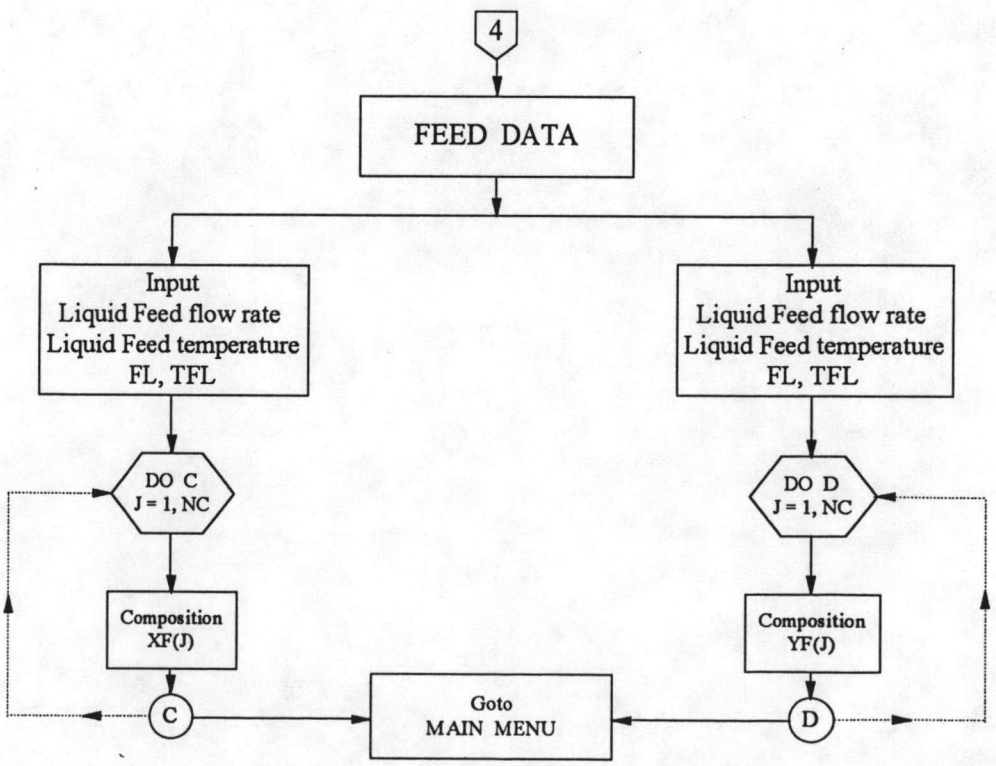
1) Component Database



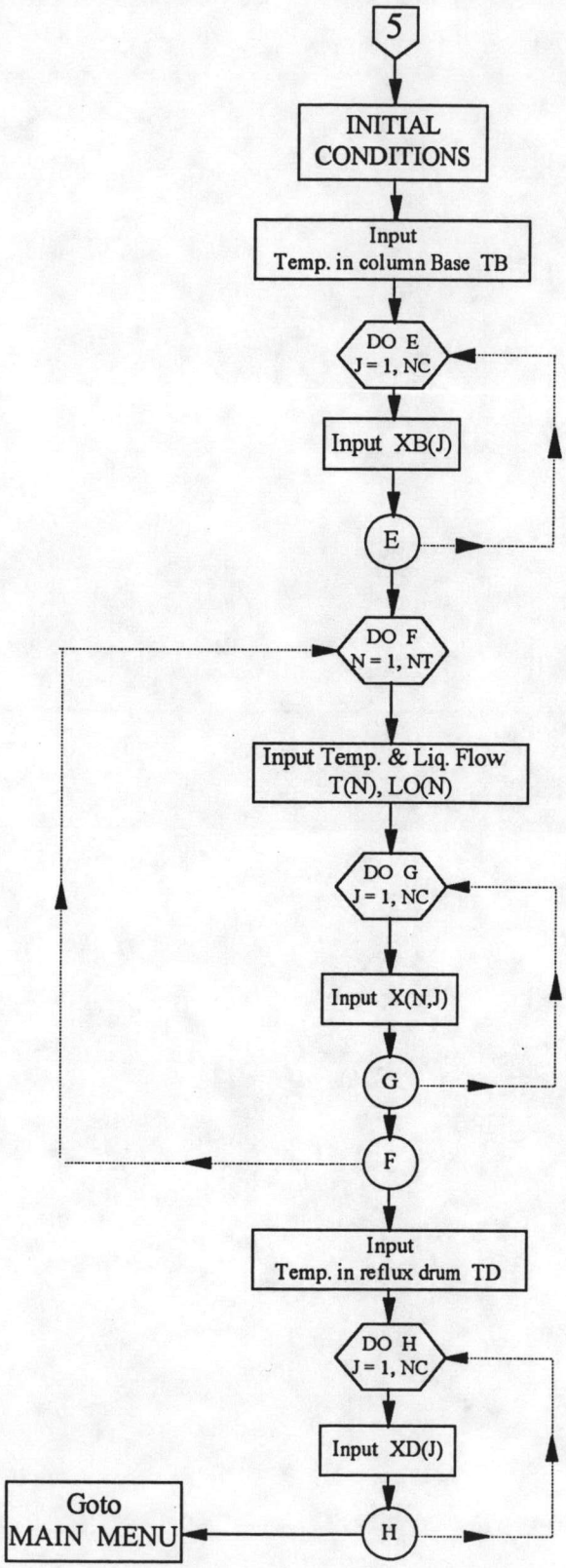
2) Characteristic of Column



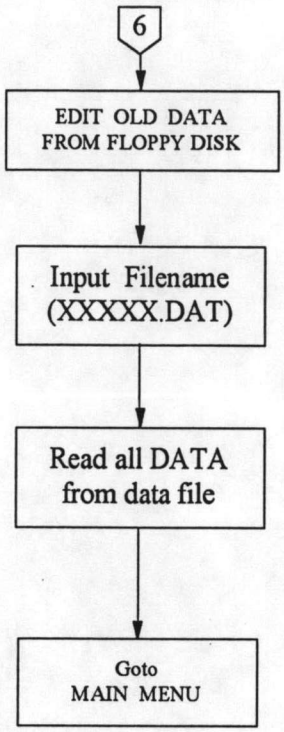
3) Operating Conditions



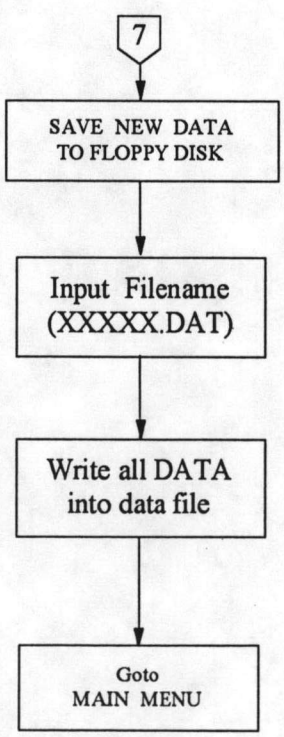
4) Feed Data



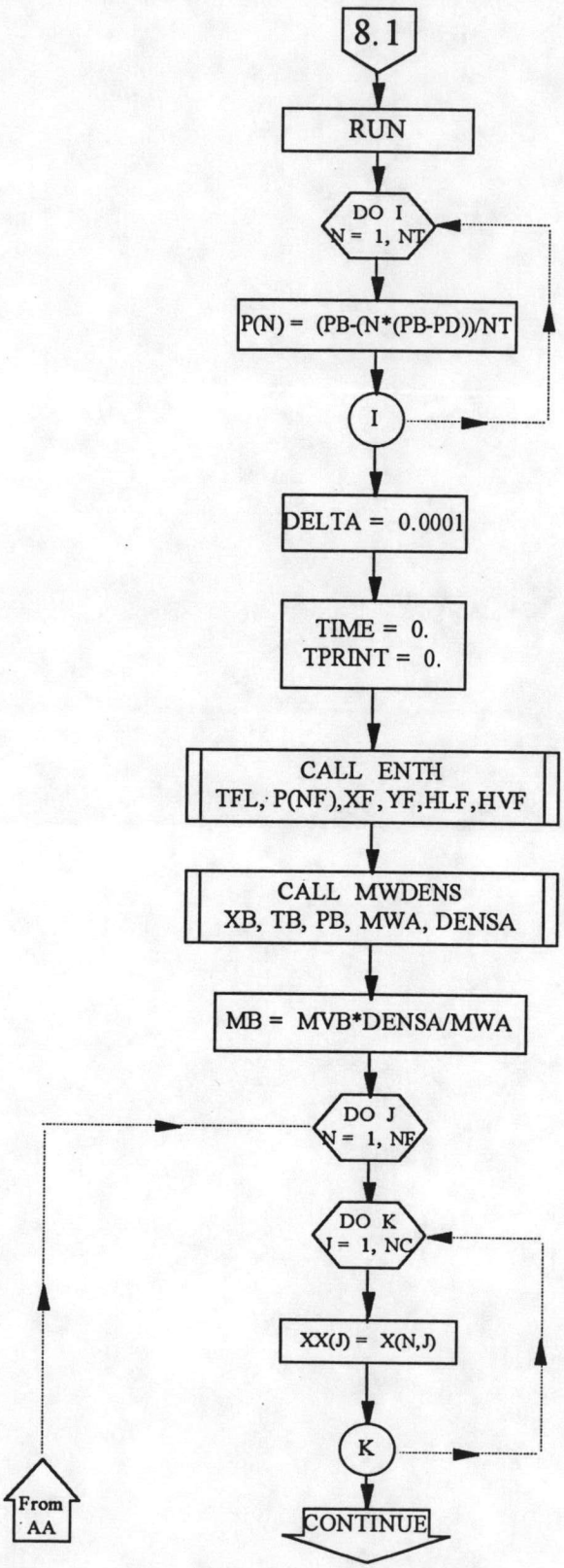
5) Initial Conditions



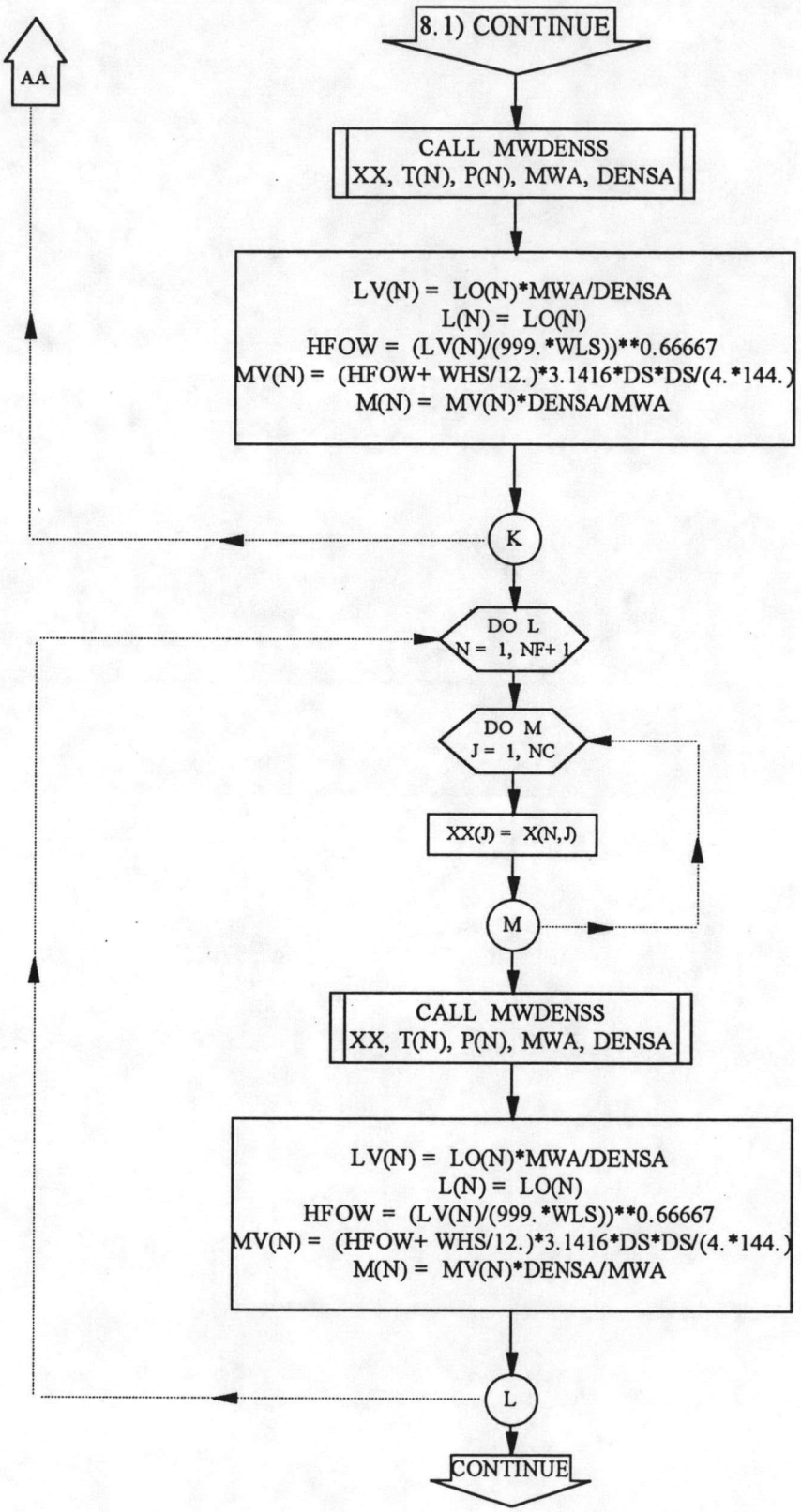
6) Edit Input Data



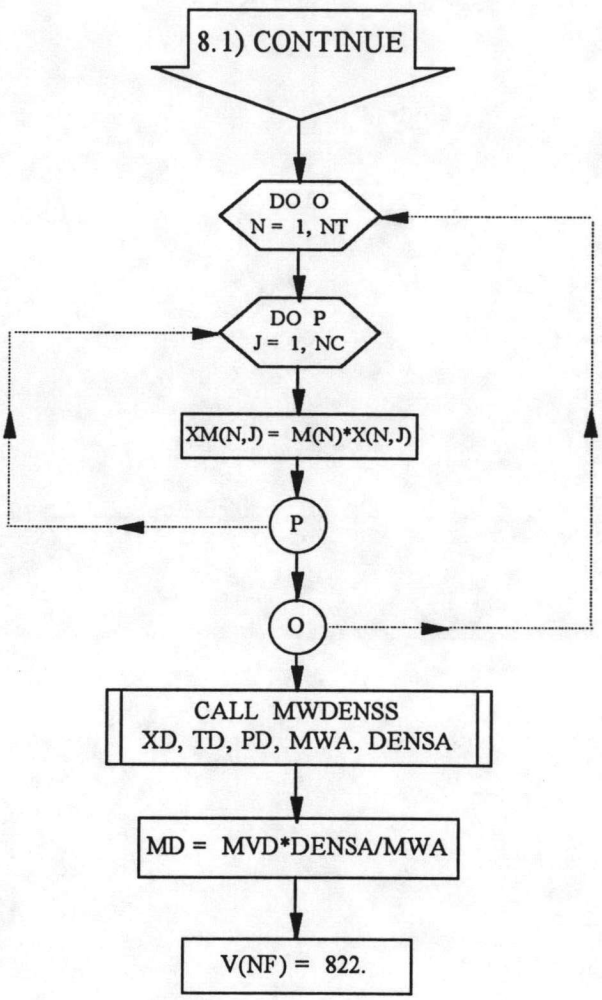
7) Save Input Data



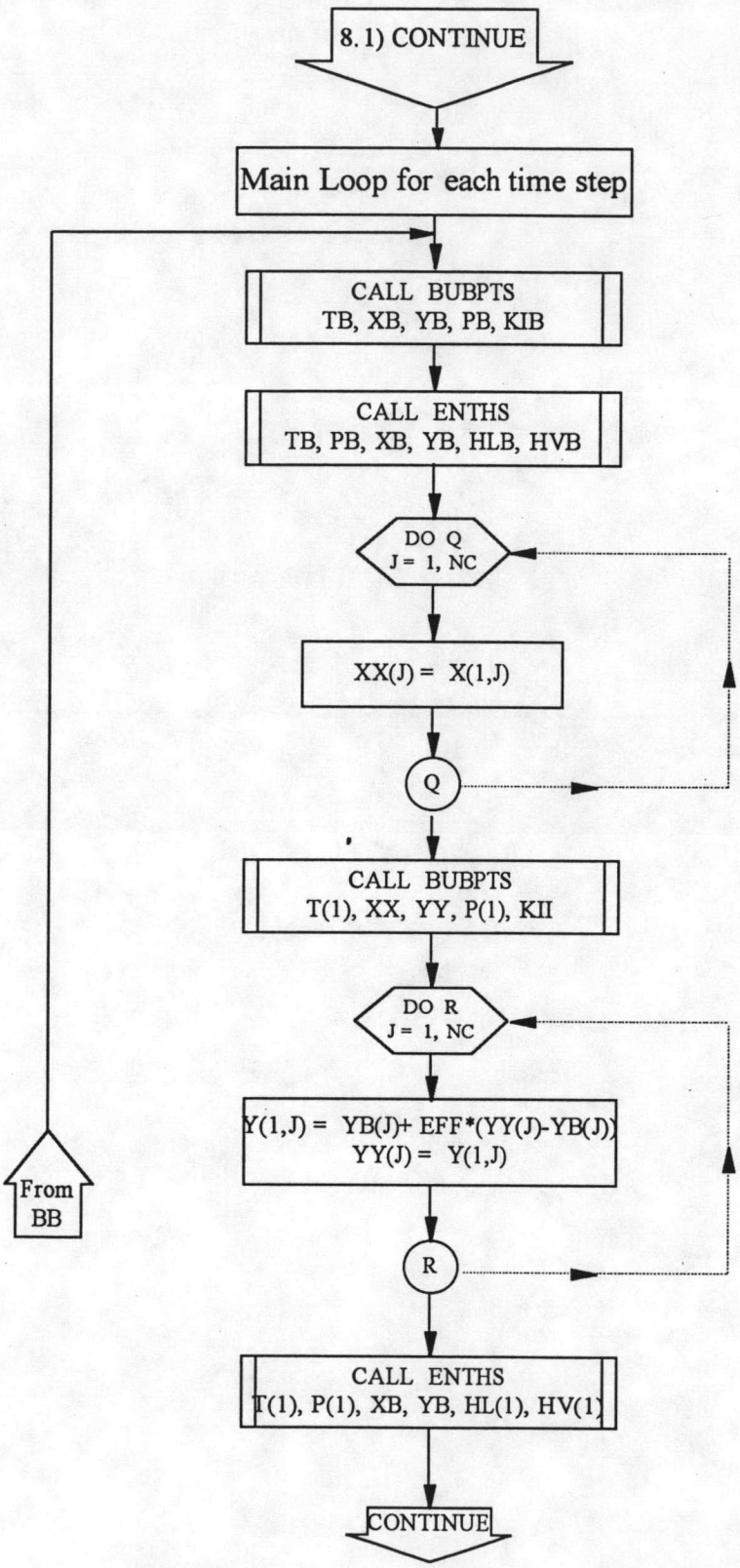
8.1) Running SRK model



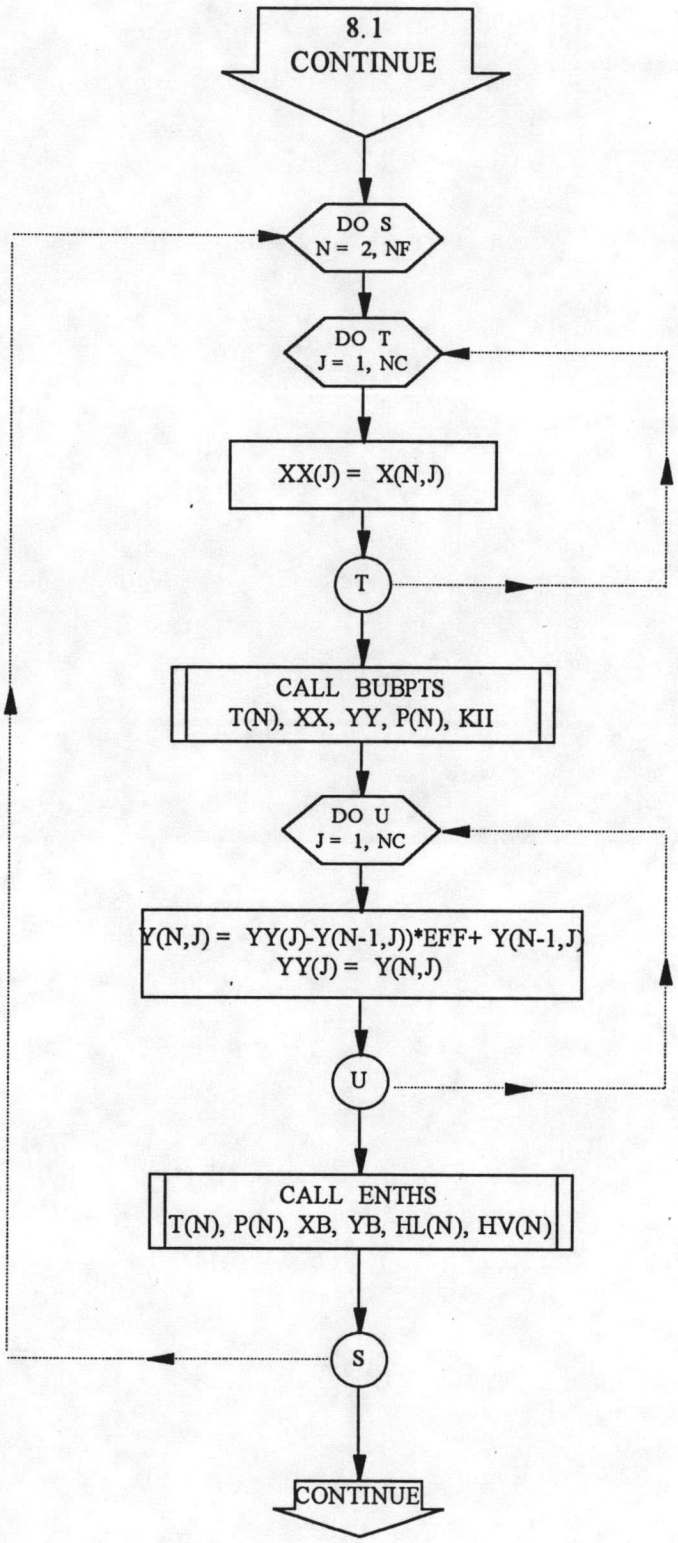
8.1) Running SRK model (Continue)



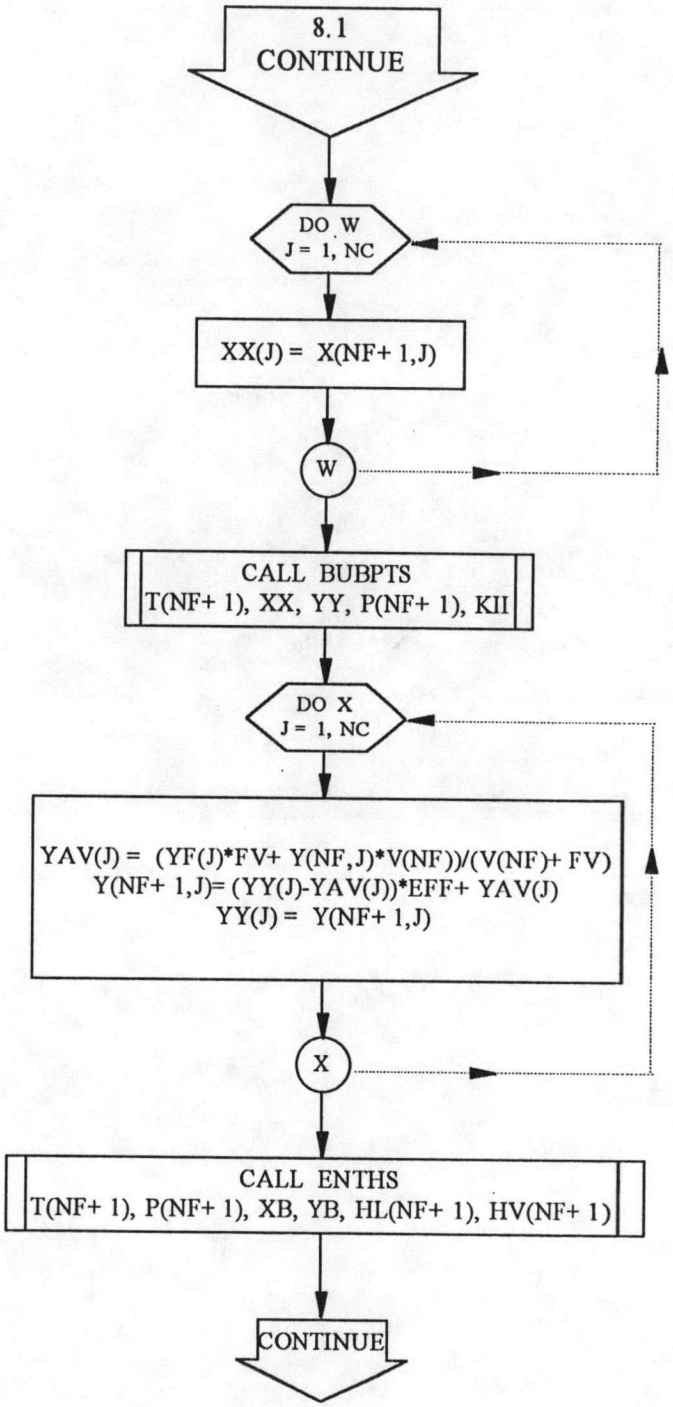
8.1) Running SRK model (Continue)



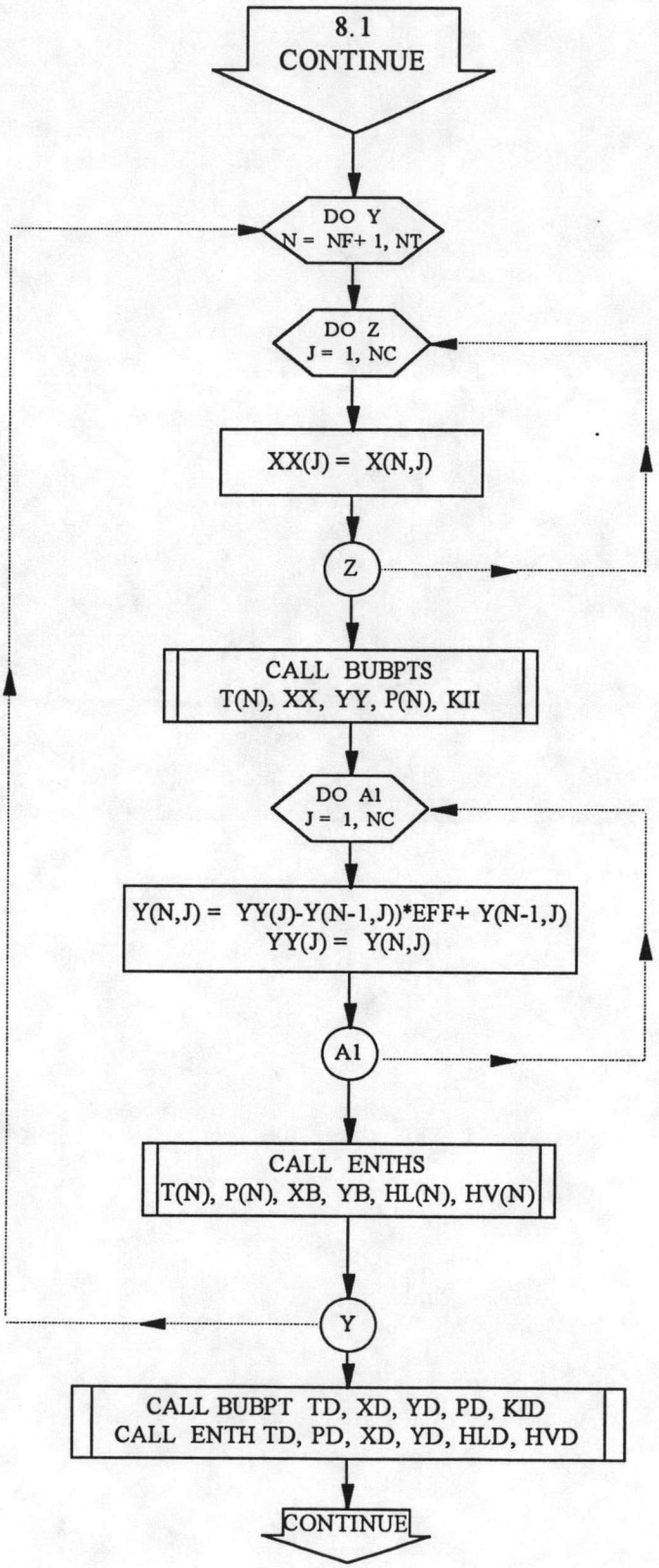
8.1) Running SRK model (Continue)



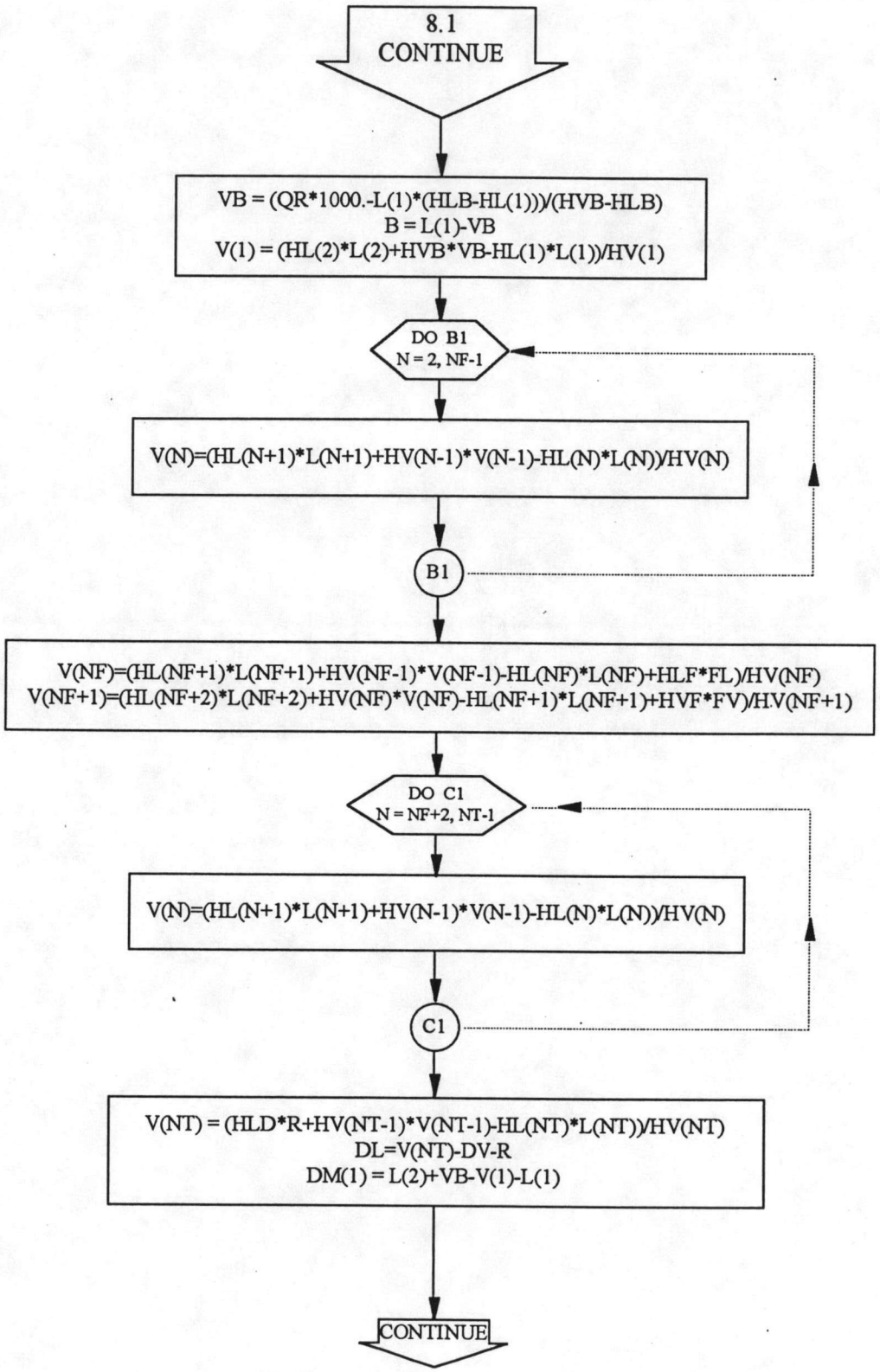
8.1) Running SRK model (Continue)



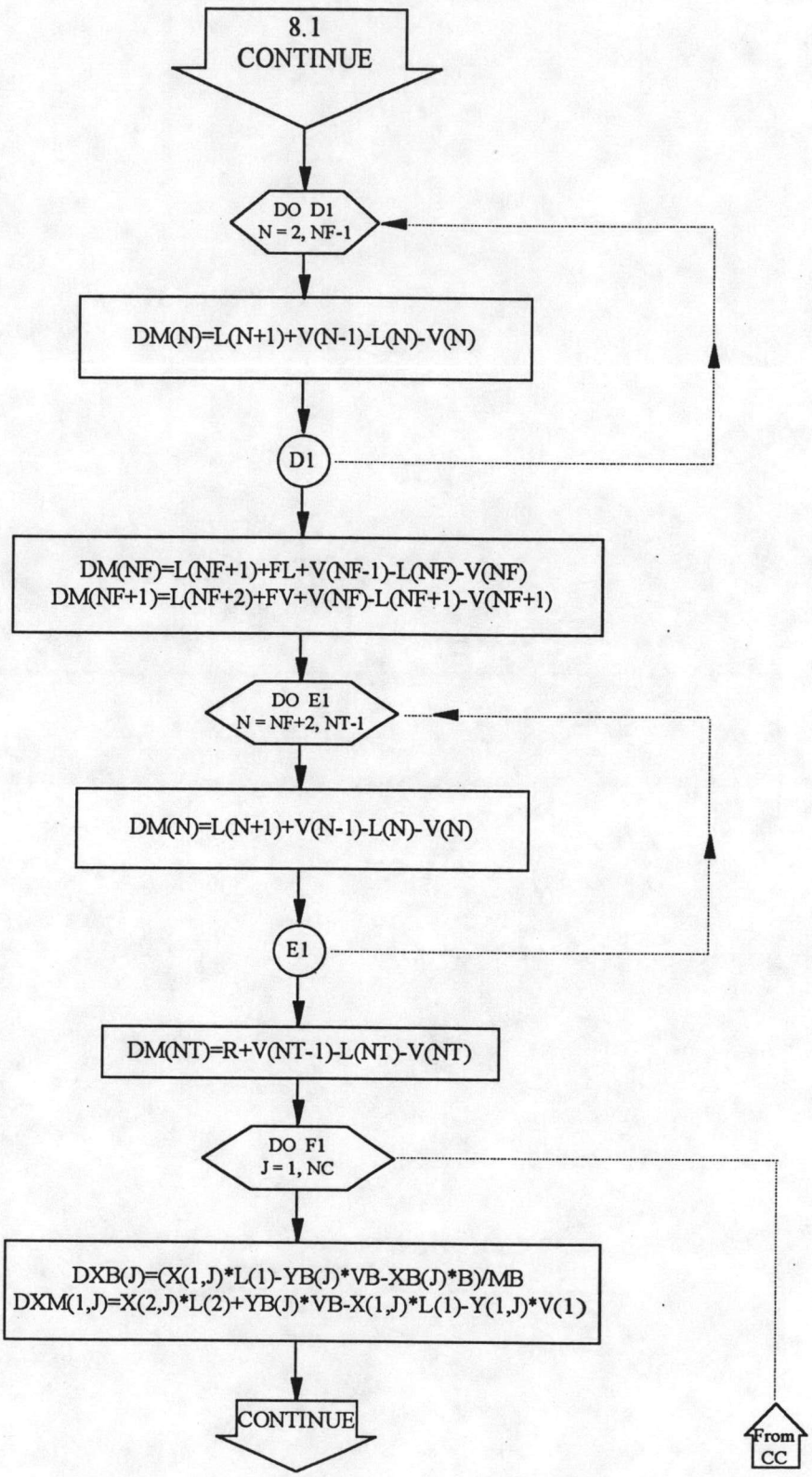
8.1) Running SRK model (Continue)



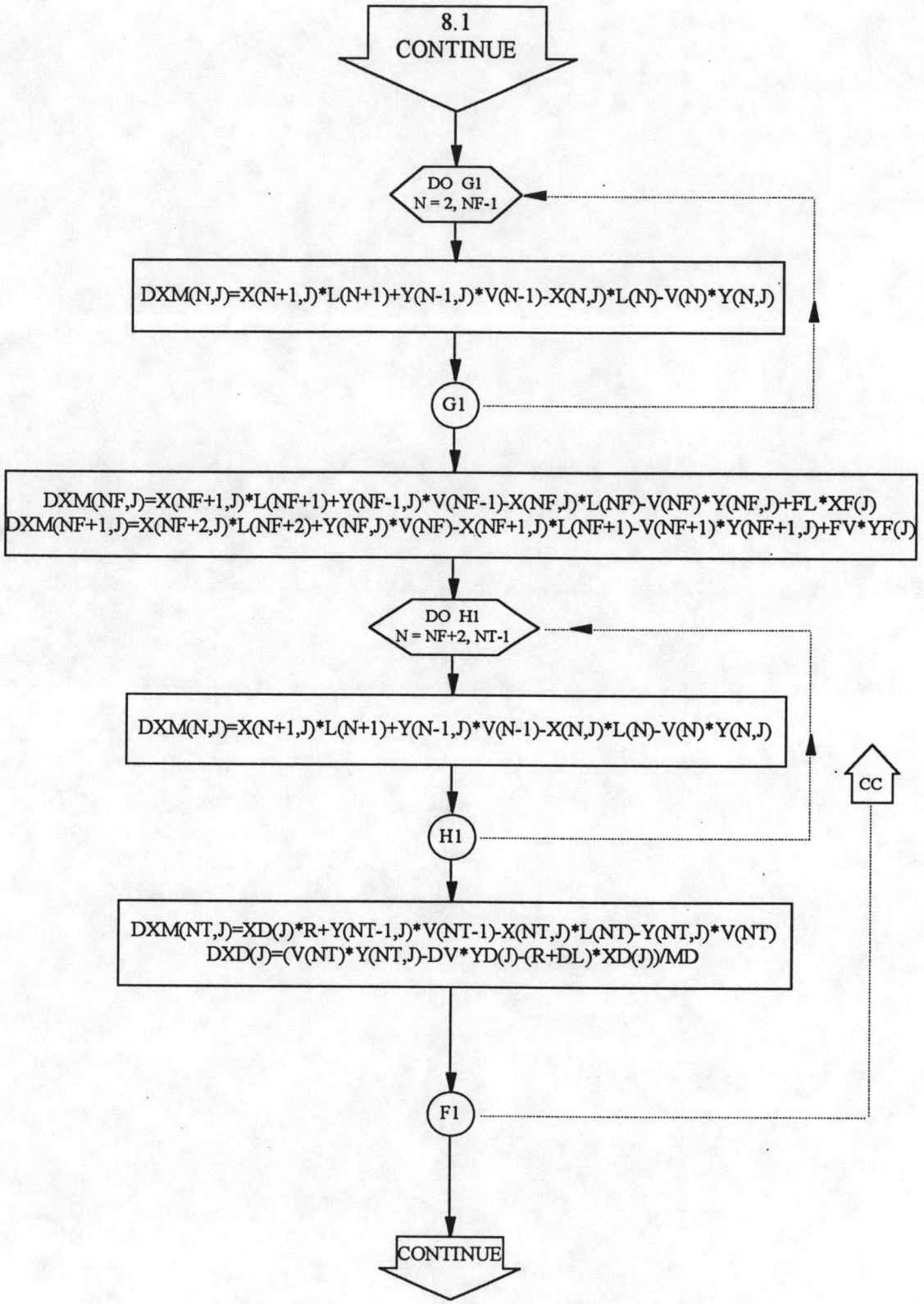
8.1) Running SRK model (Continue)



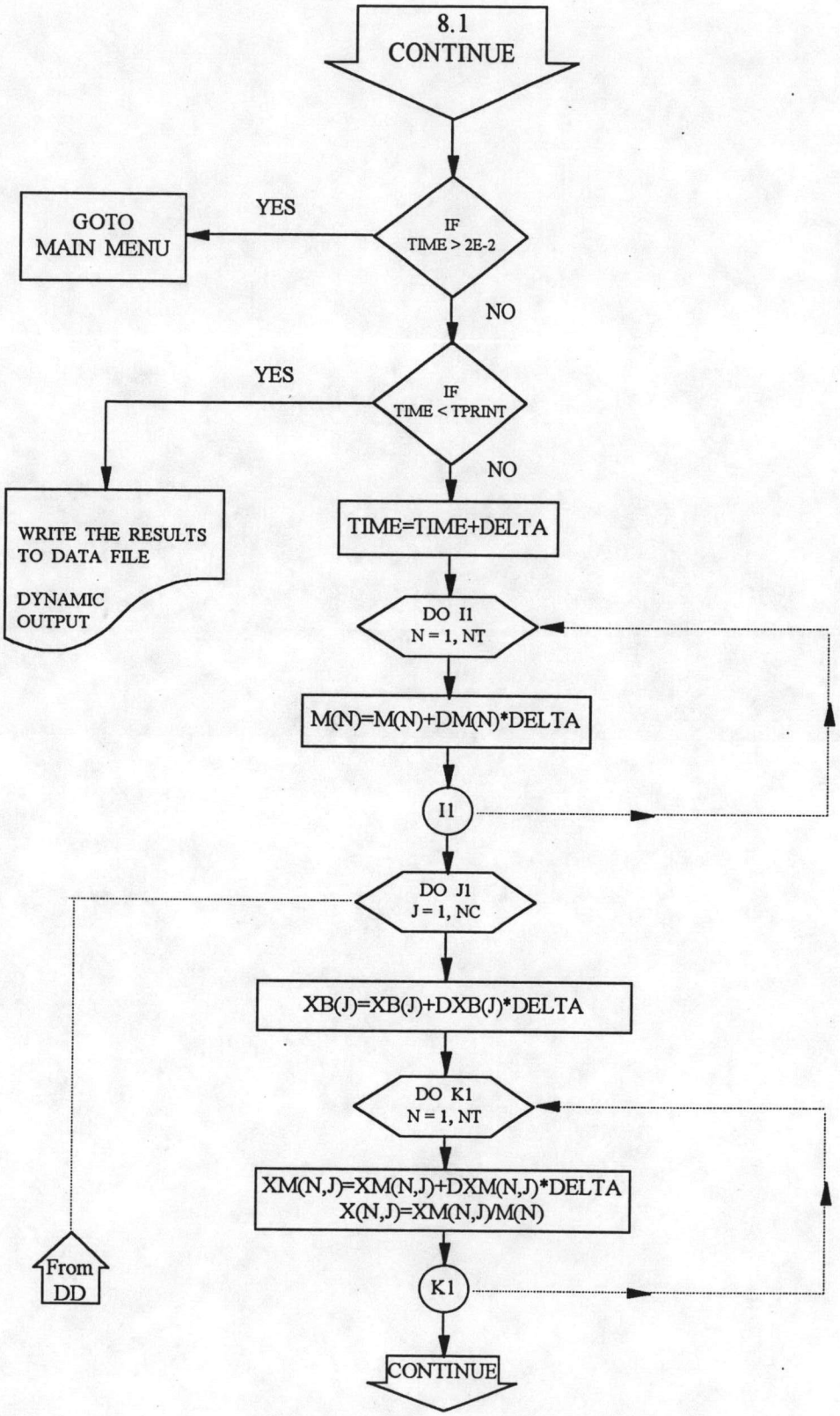
8.1) Running SRK model (Continue)



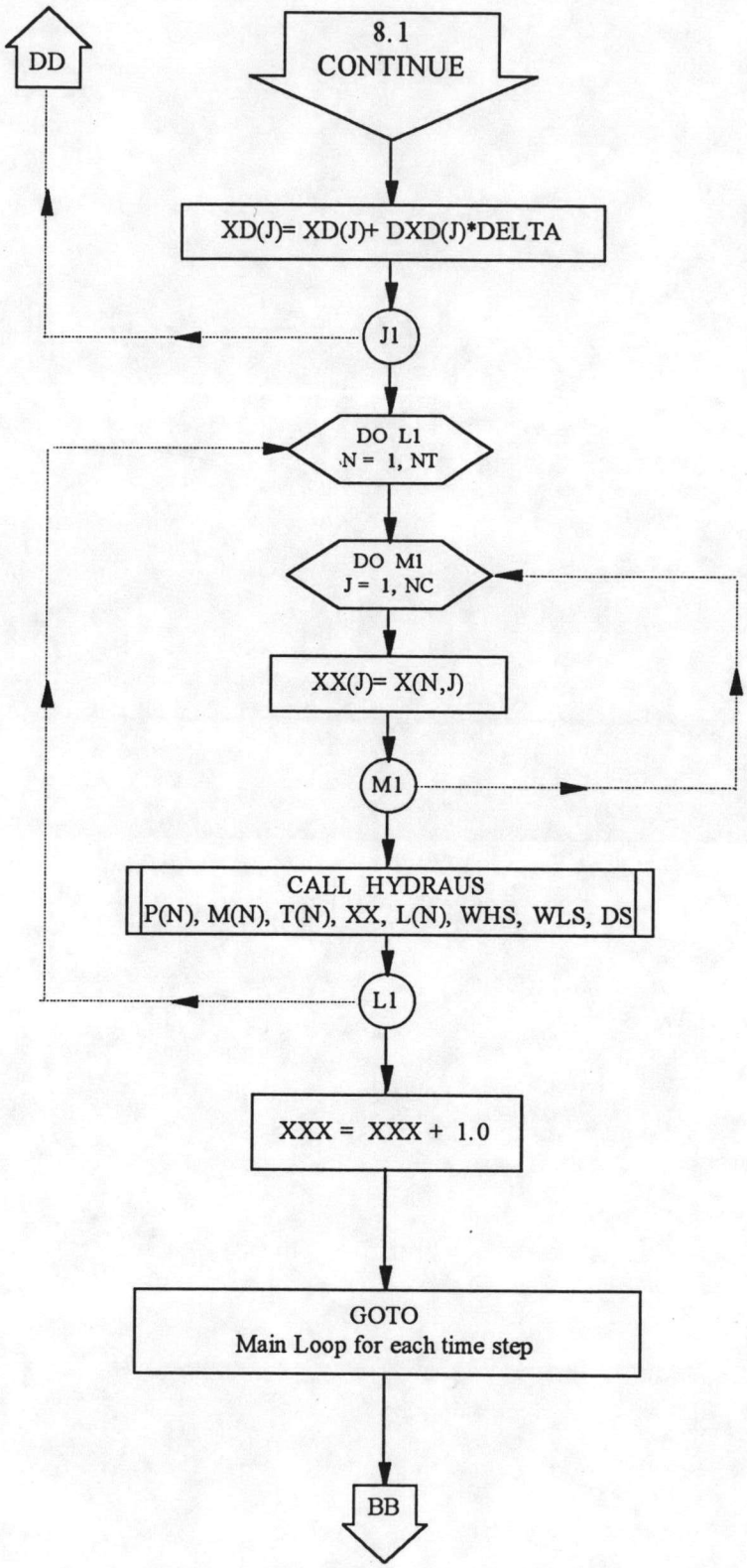
8.1) Running SRK model (Continue)



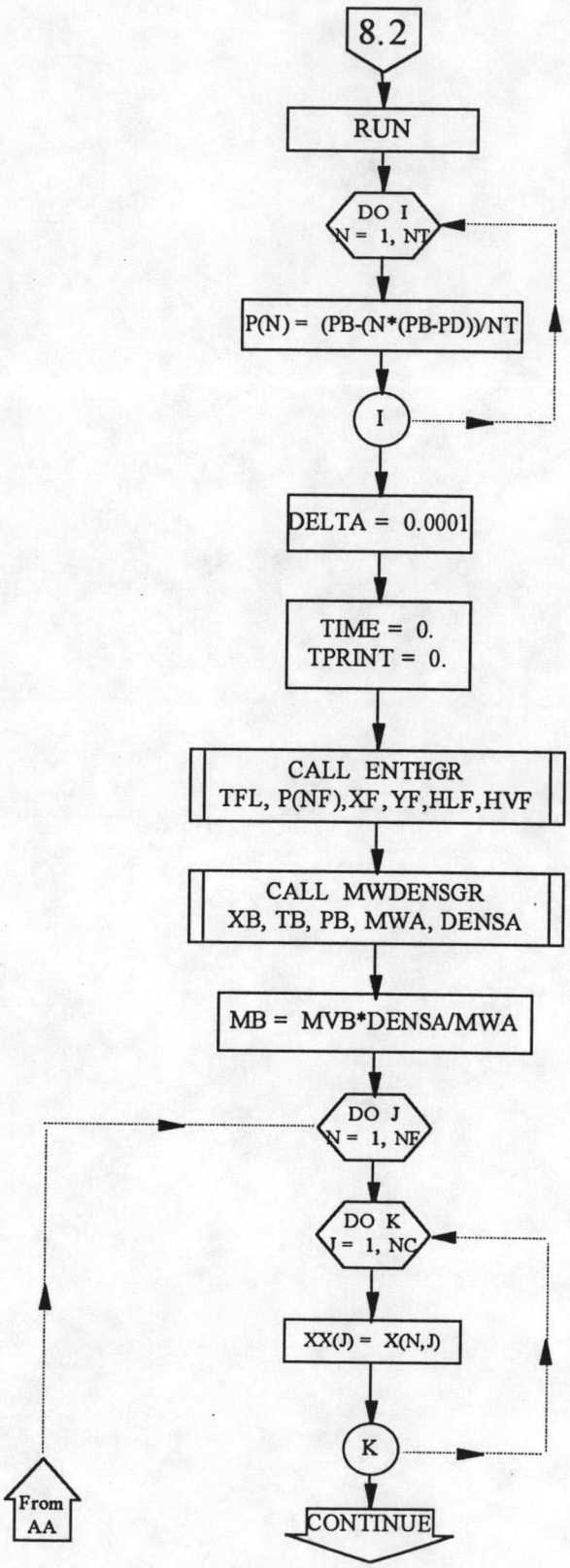
8.1) Running SRK model (Continue)



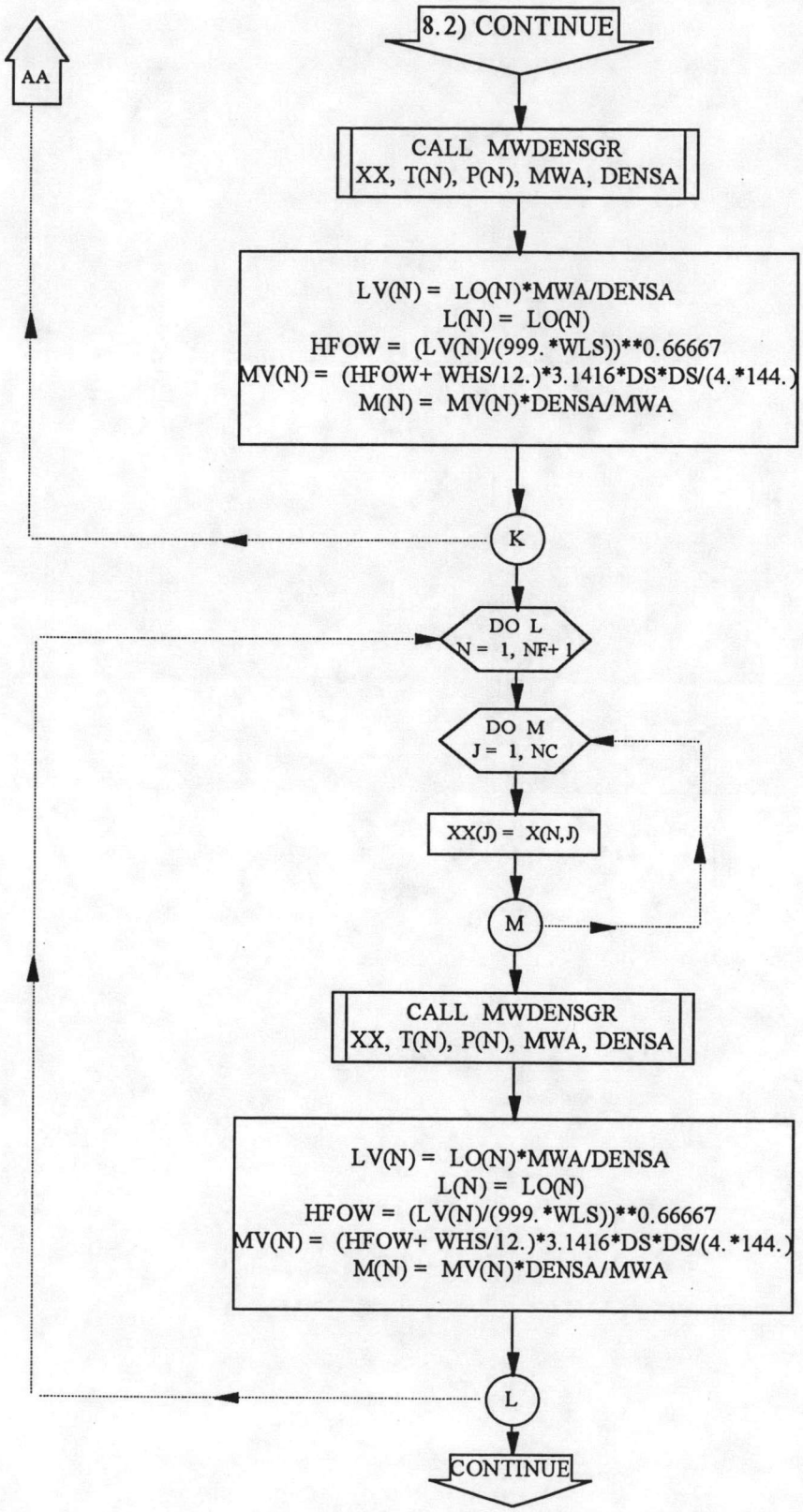
8.1) Running SRK model (Continue)



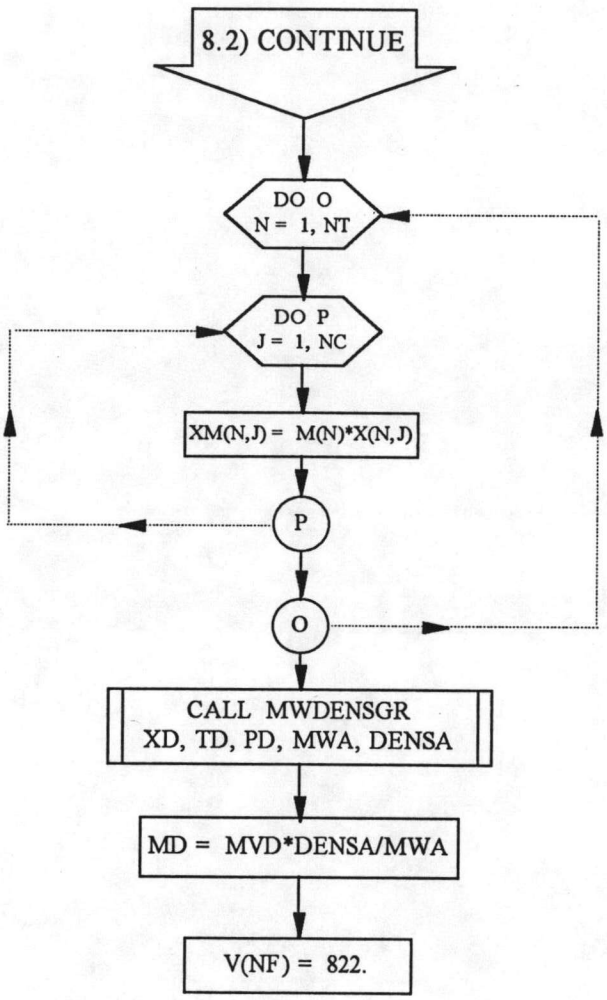
8.1) Running SRK model (Continue)



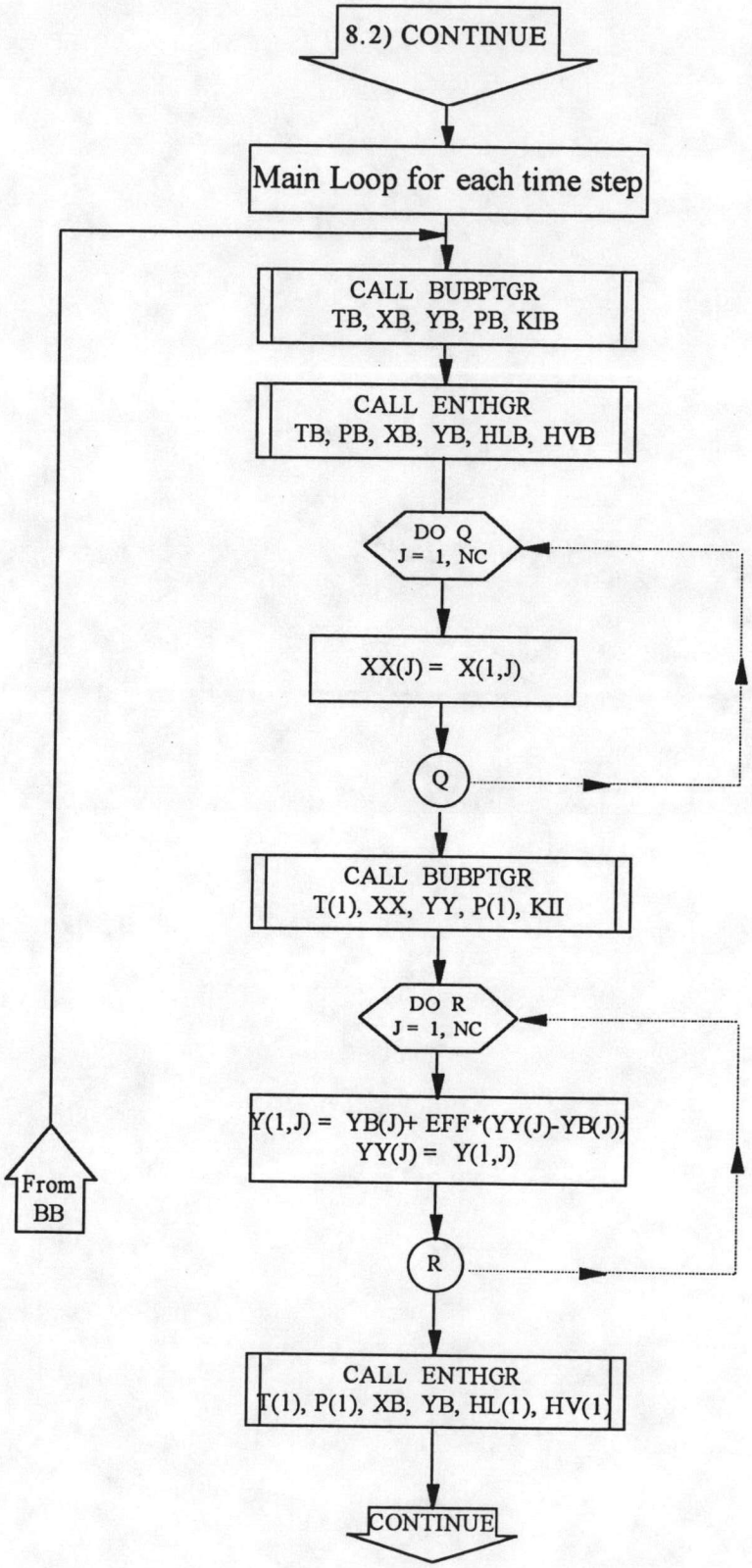
8.2) Running GRK model



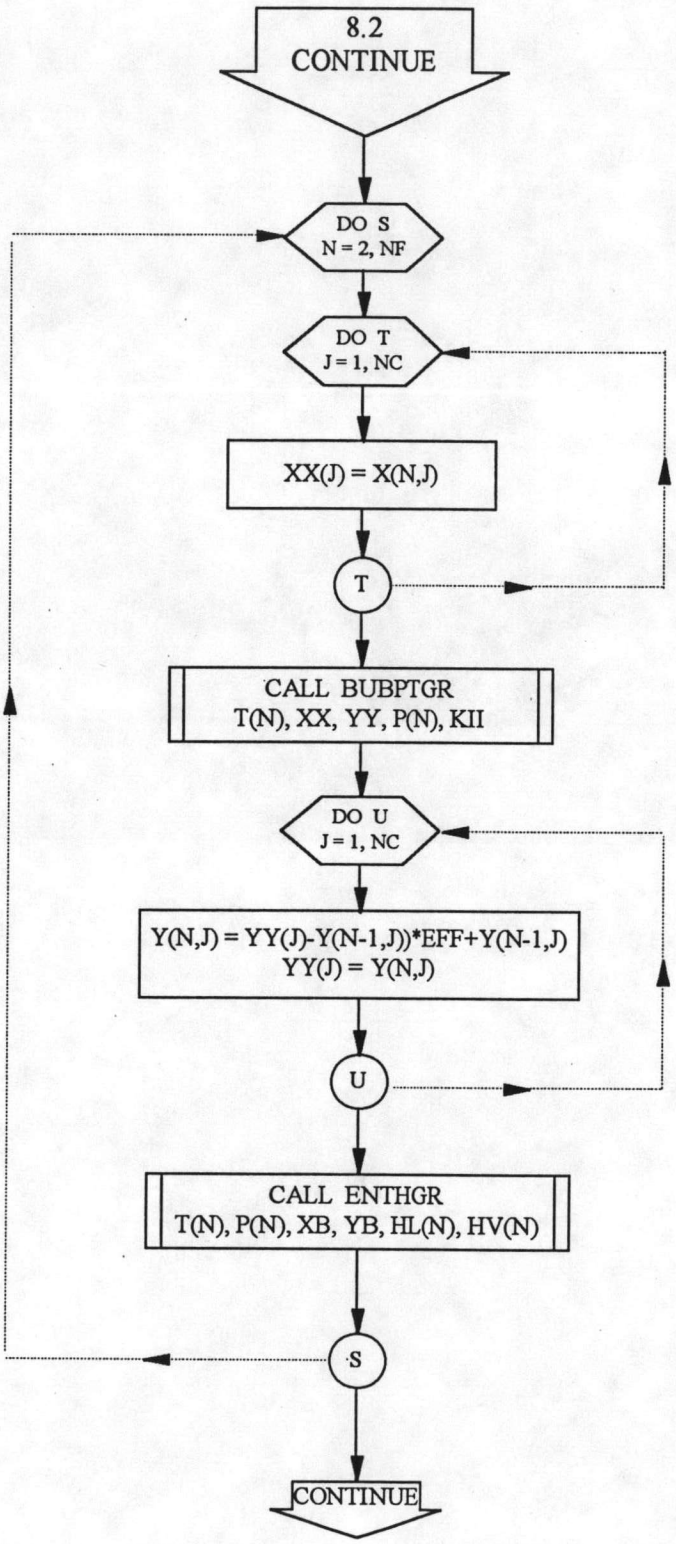
8.2) Running GRK model (Continue)



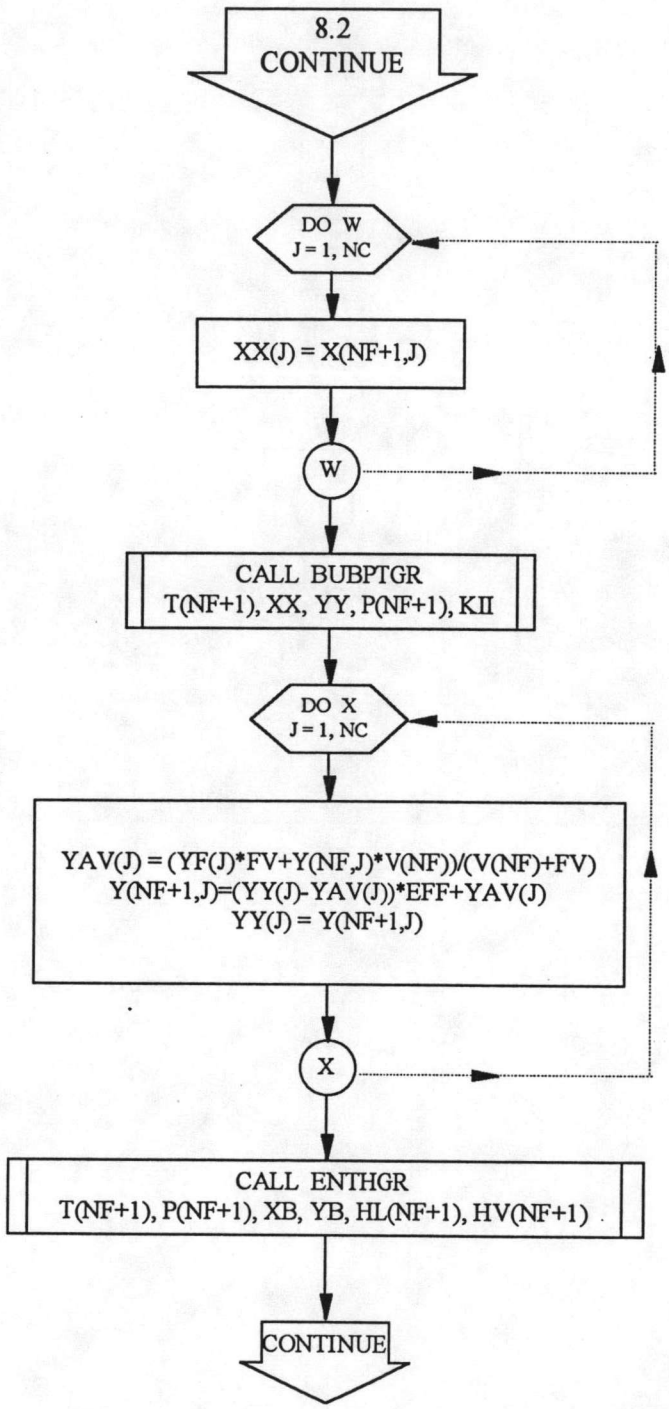
8.2) Running GRK model (Continue)



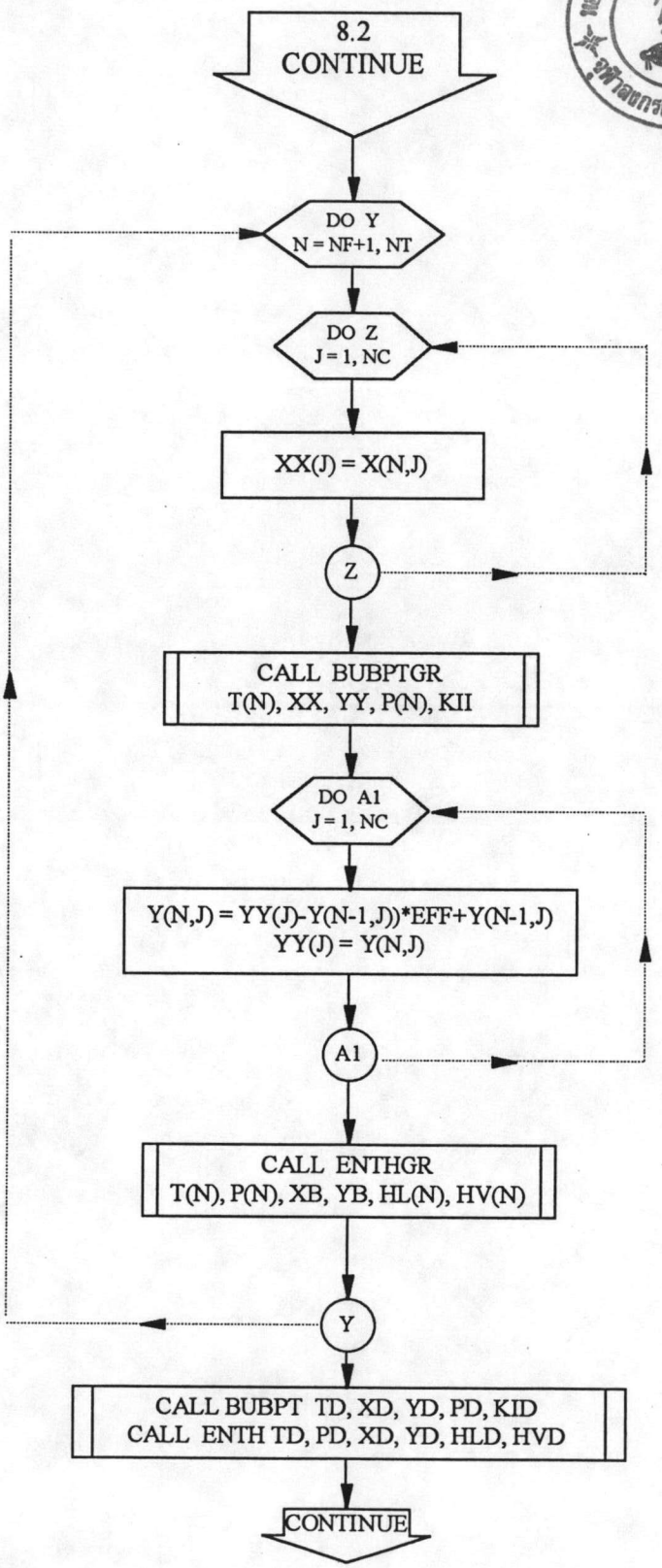
8.2) Running GRK model (Continue)



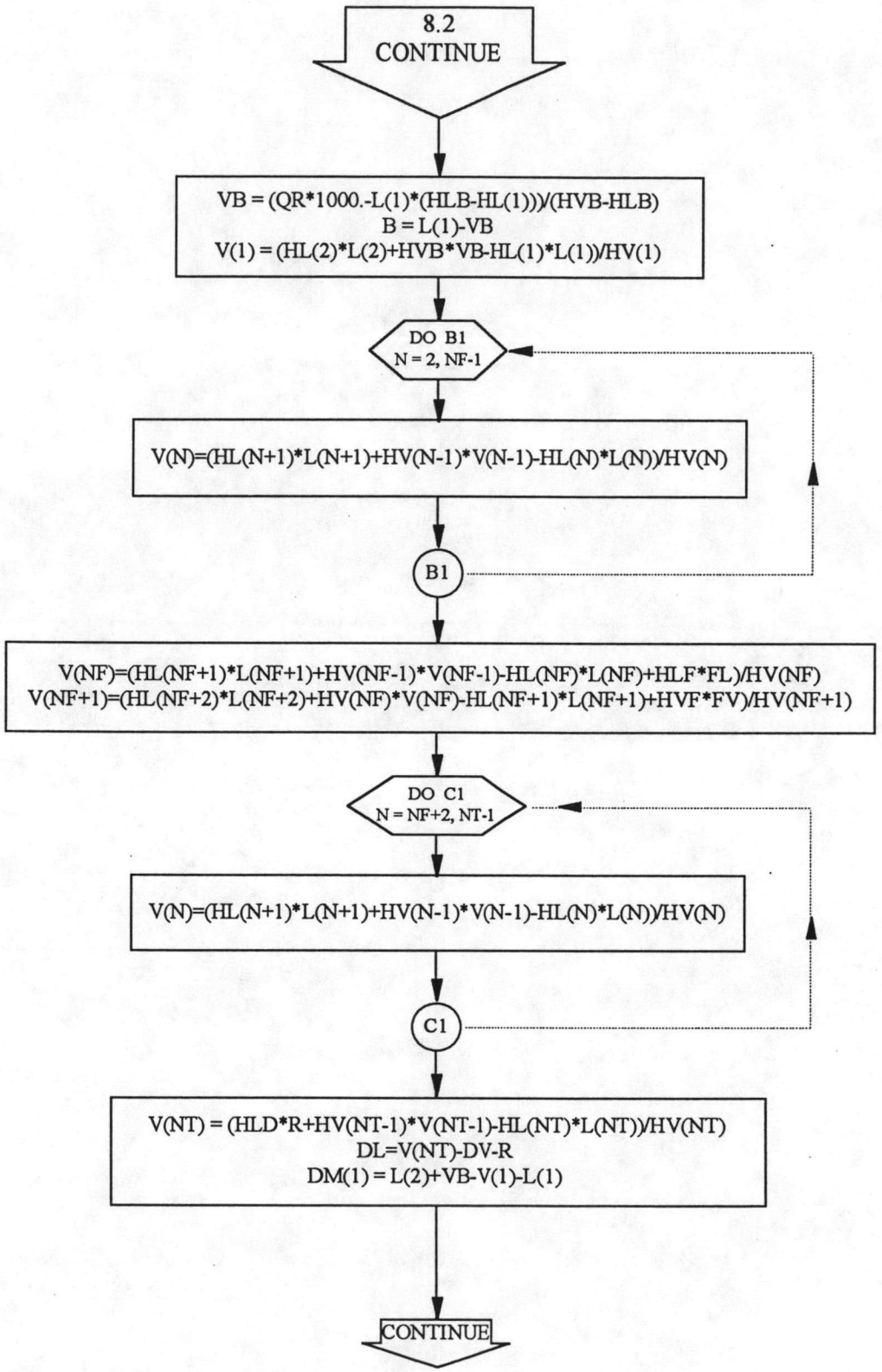
8.2) Running GRK model (Continue)



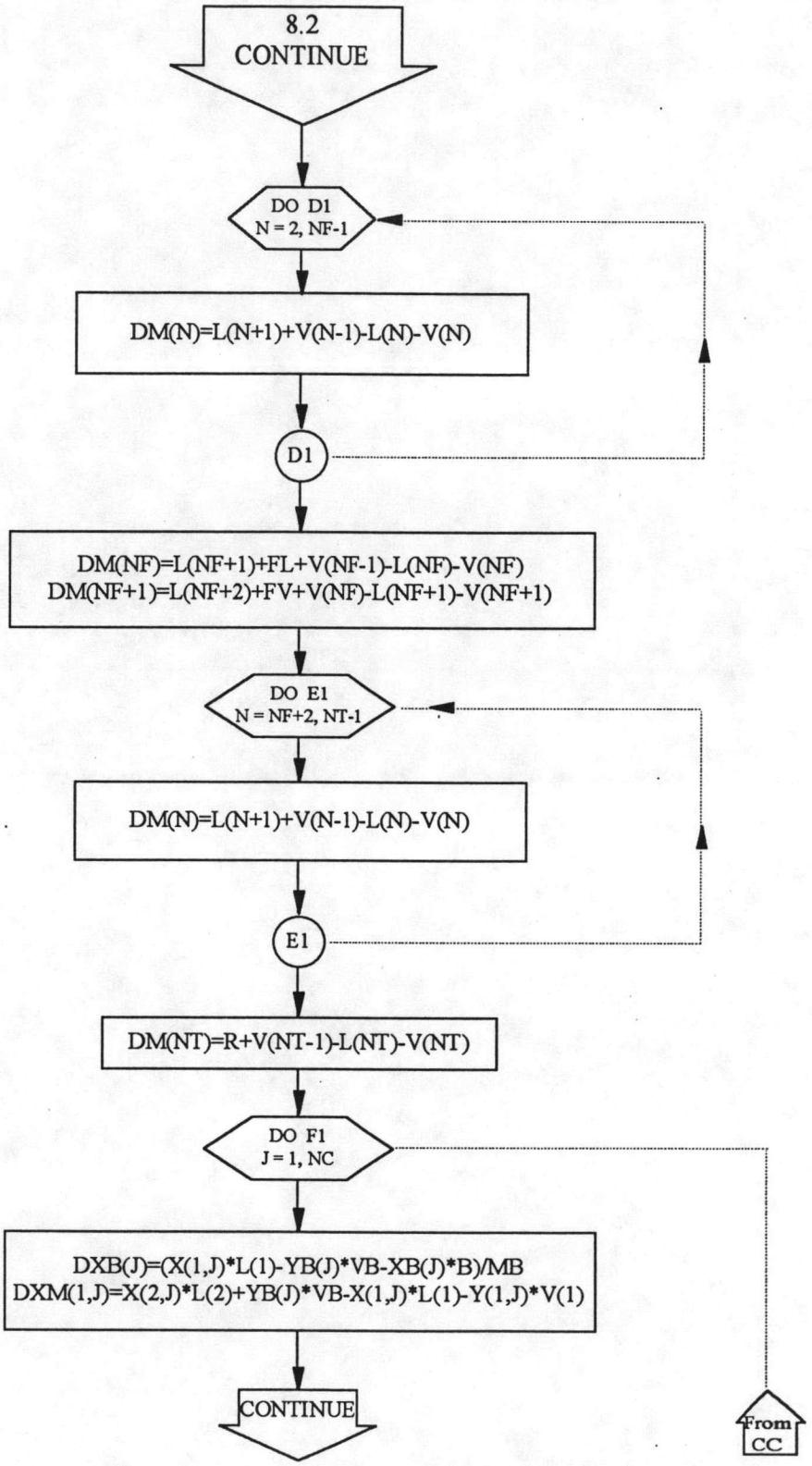
8.2) Running GRK model (Continue)



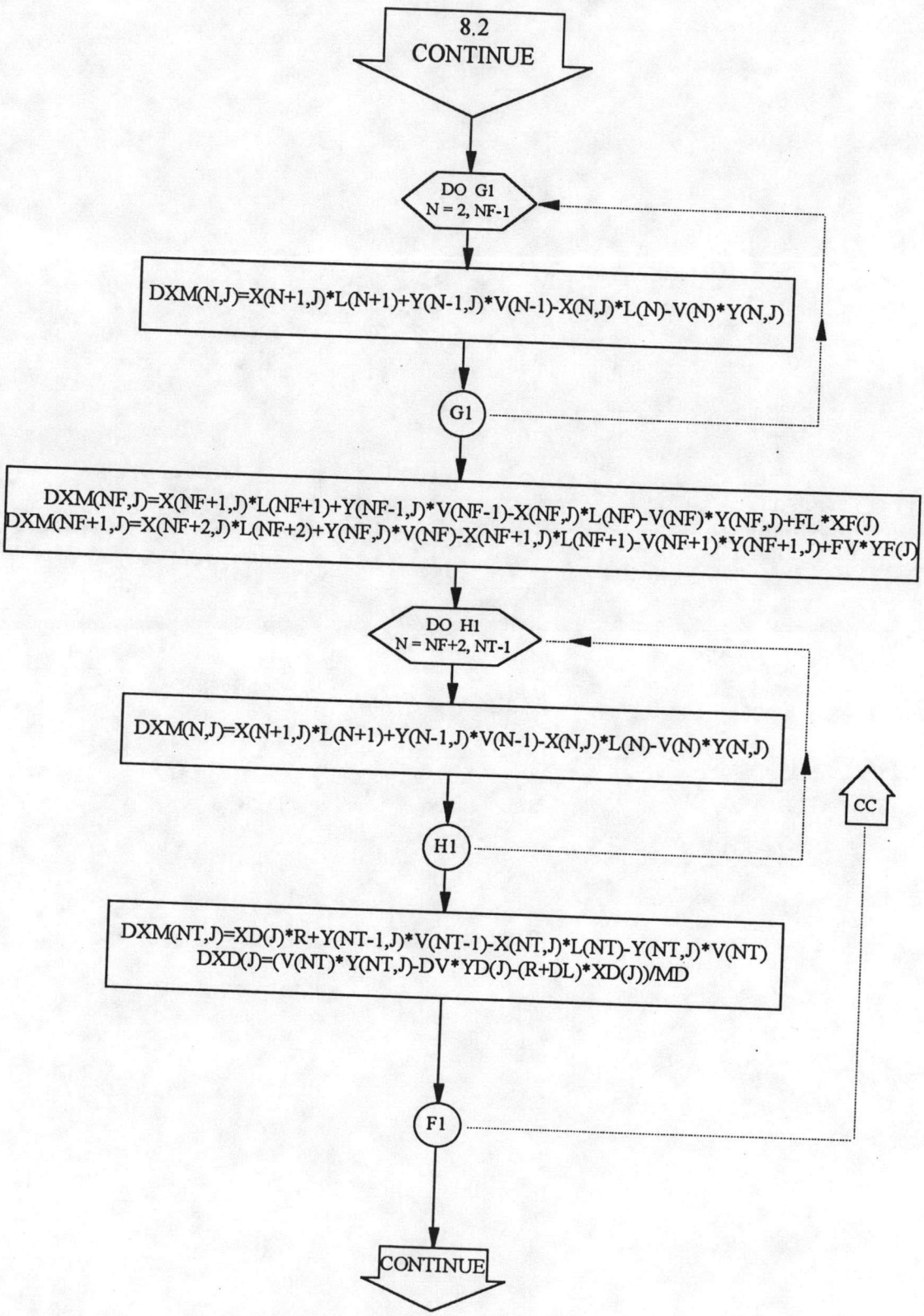
8.2) Running GRK model (Continue)



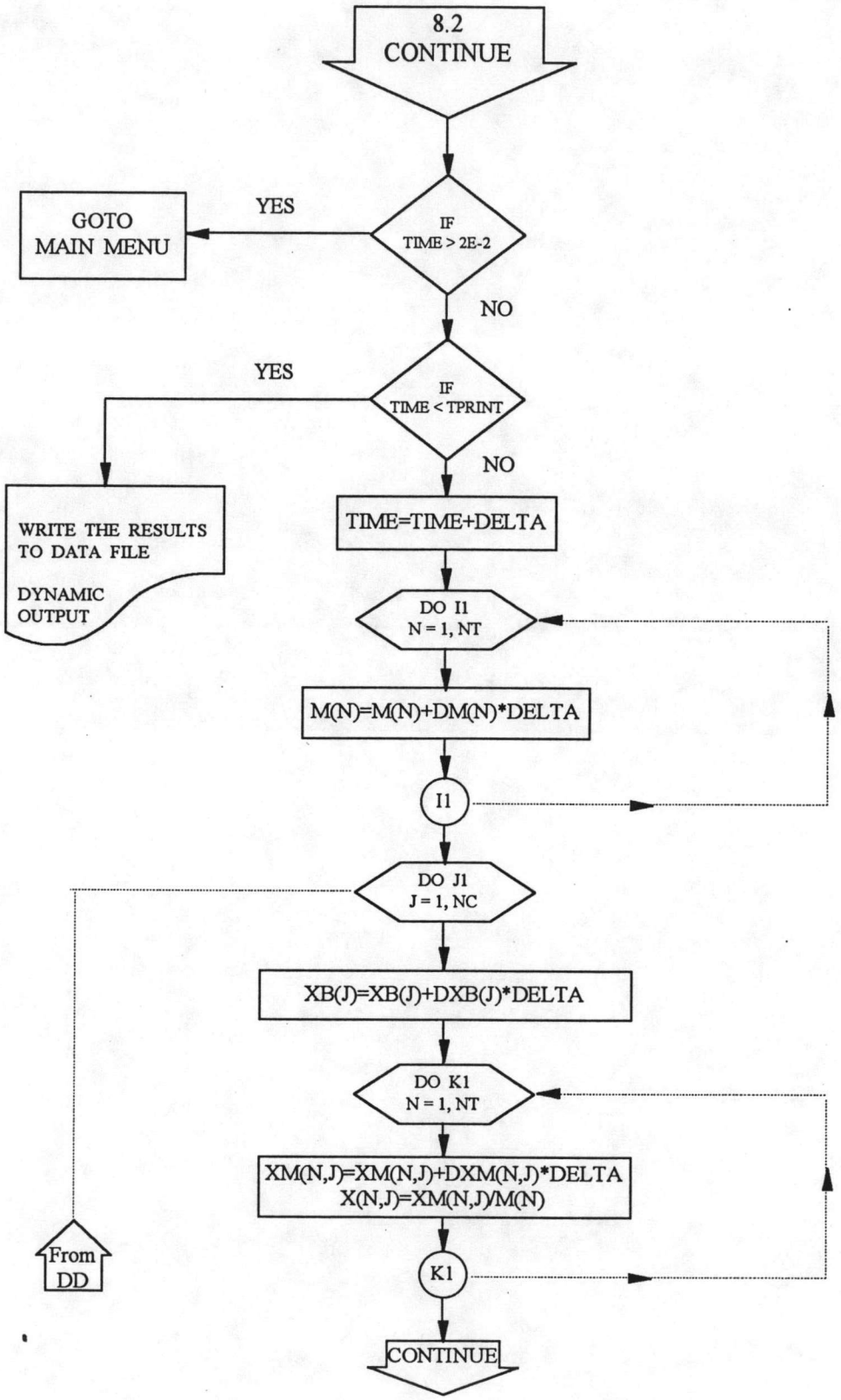
8.2) Running GRK model (Continue)



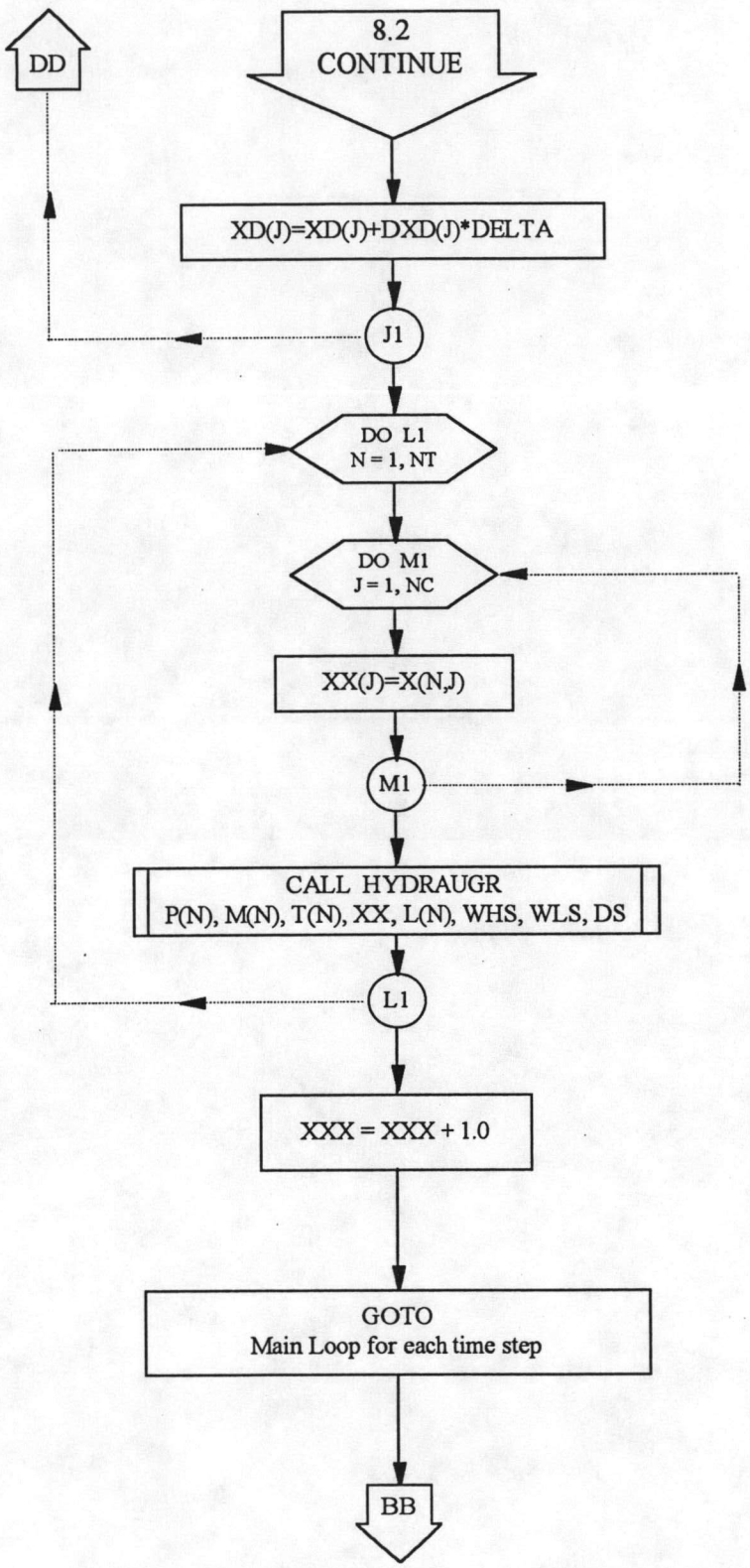
8.2) Running GRK model (Continue)



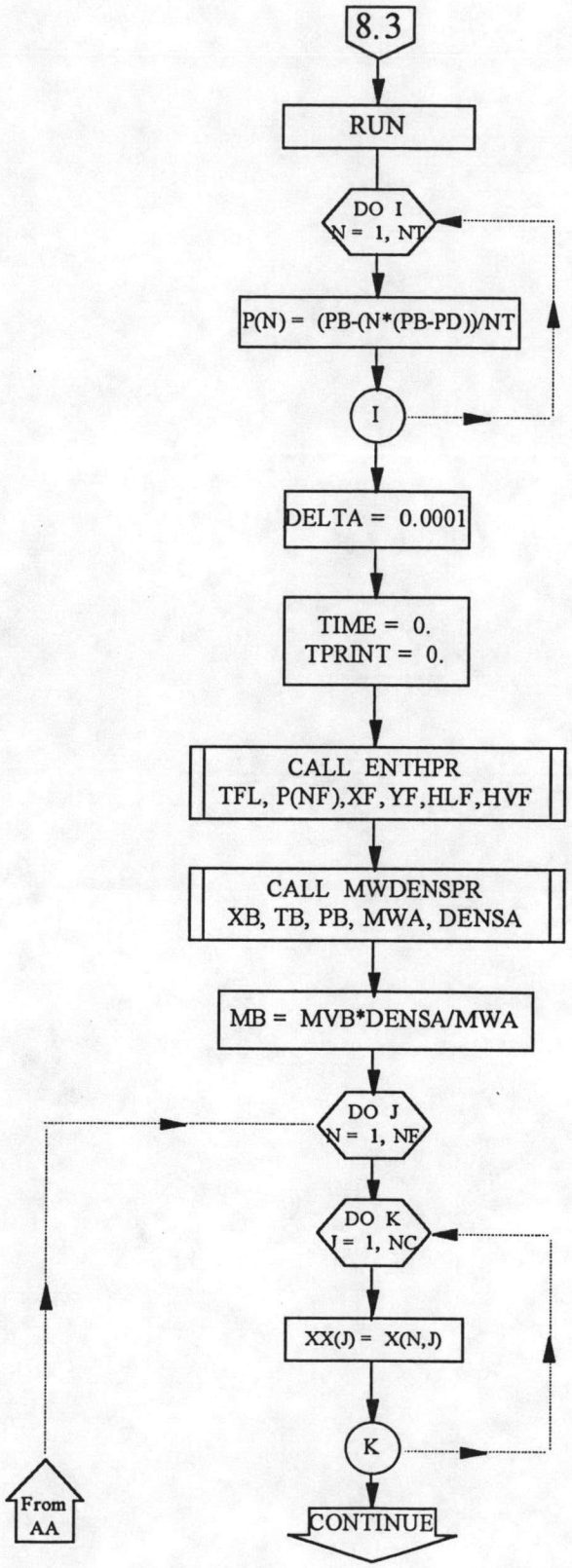
8.2) Running GRK model (Continue)



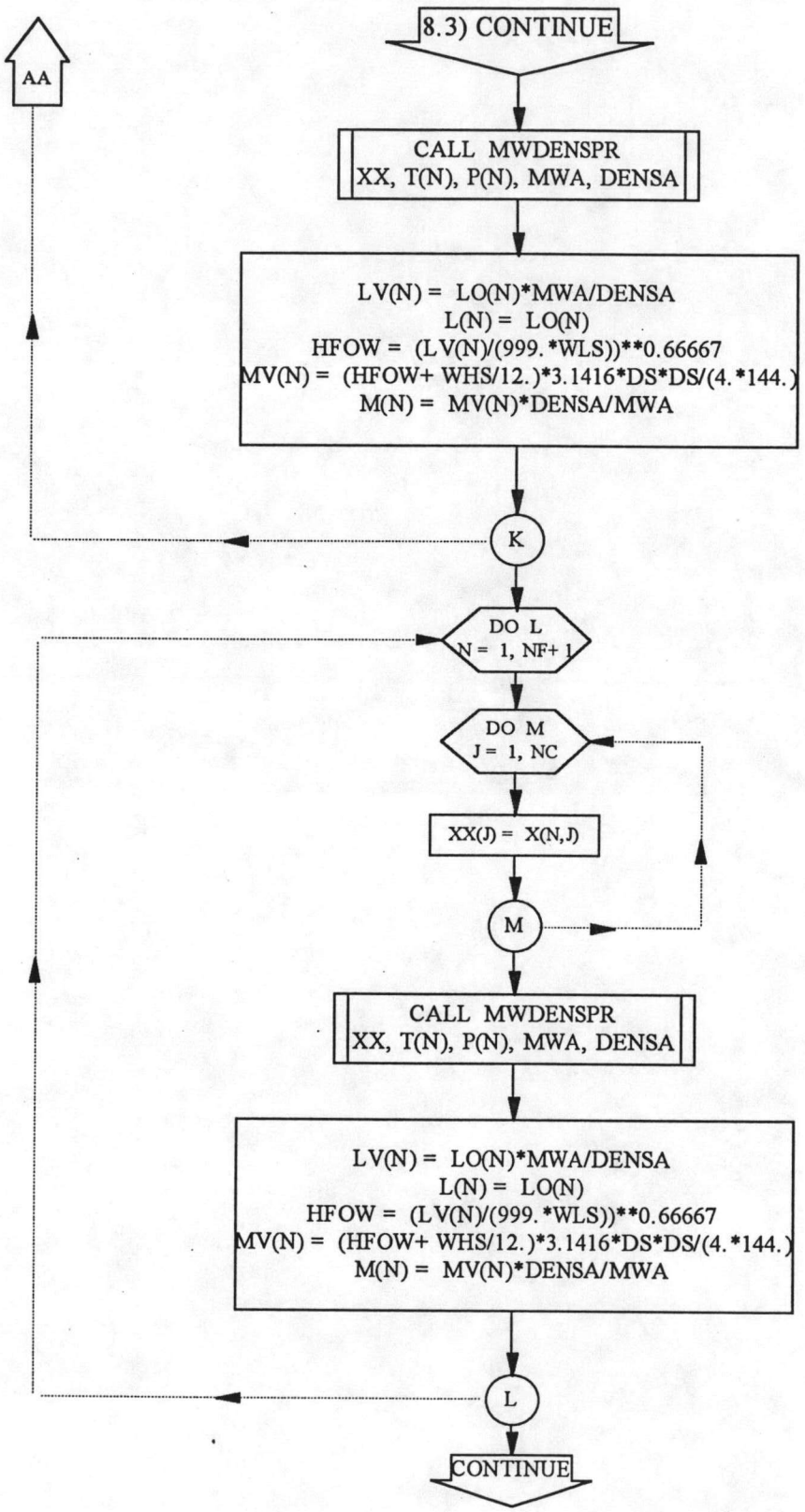
8.2) Running GRK model (Continue)



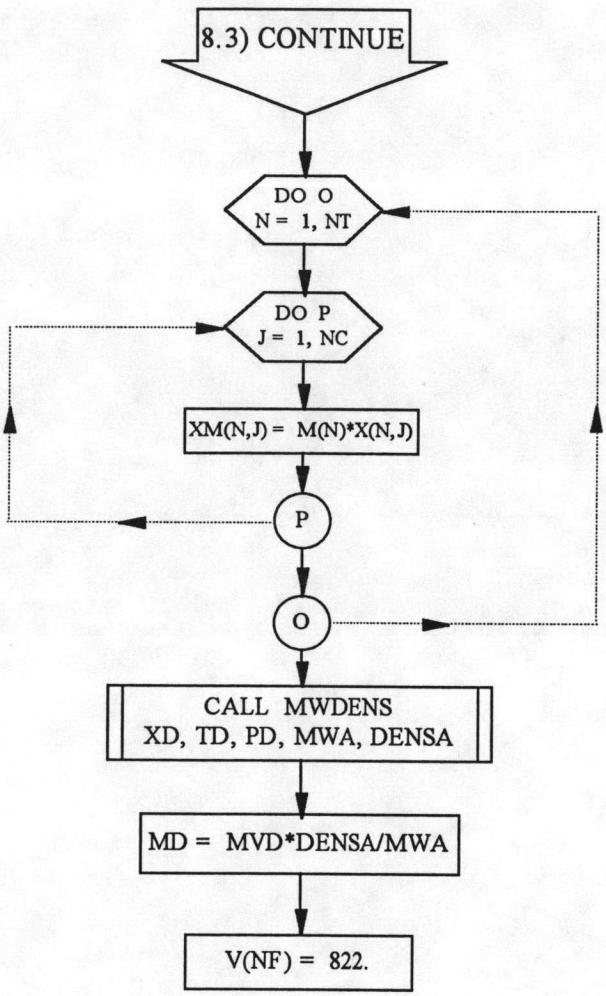
8.2) Running GRK model (Continue)



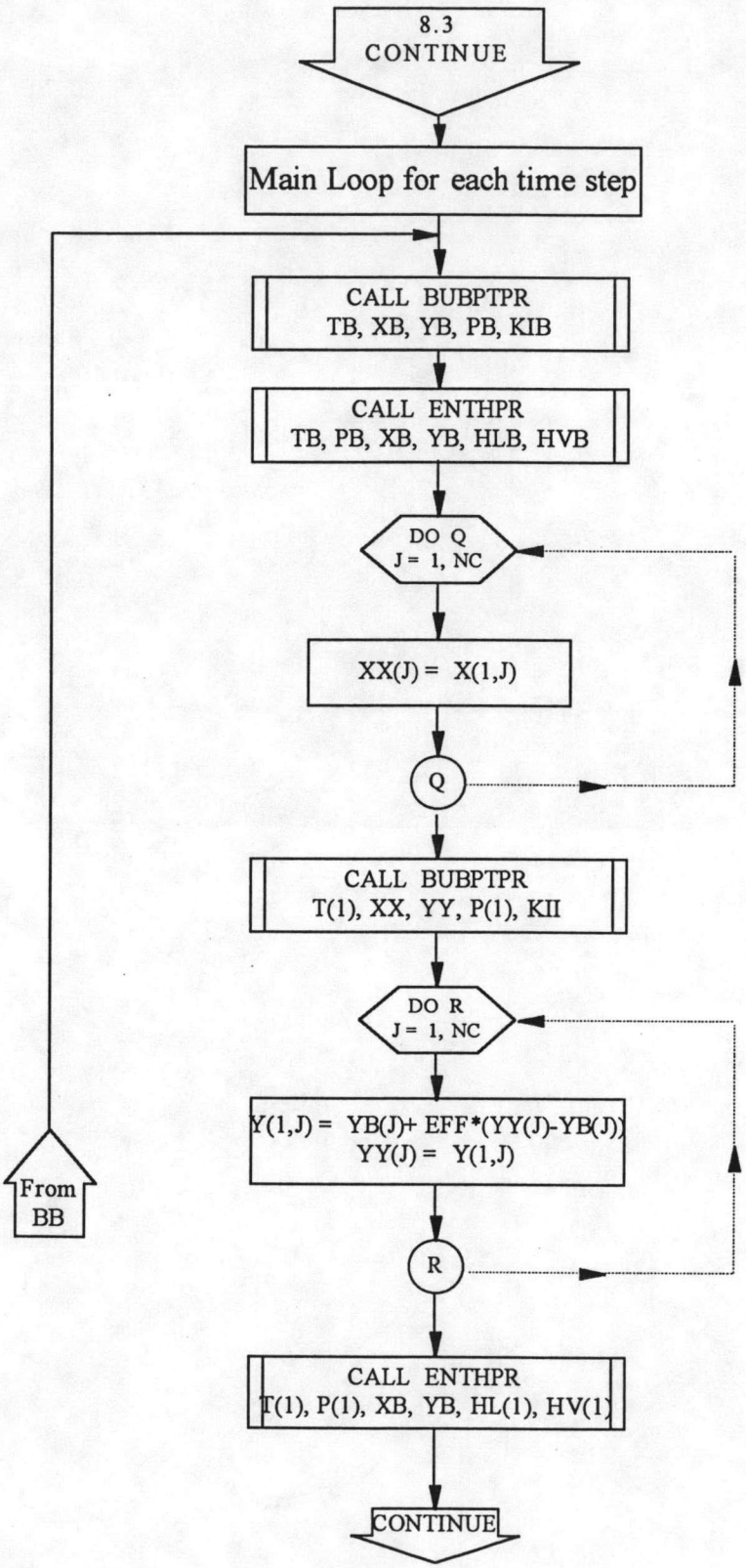
8.3) Running PR model



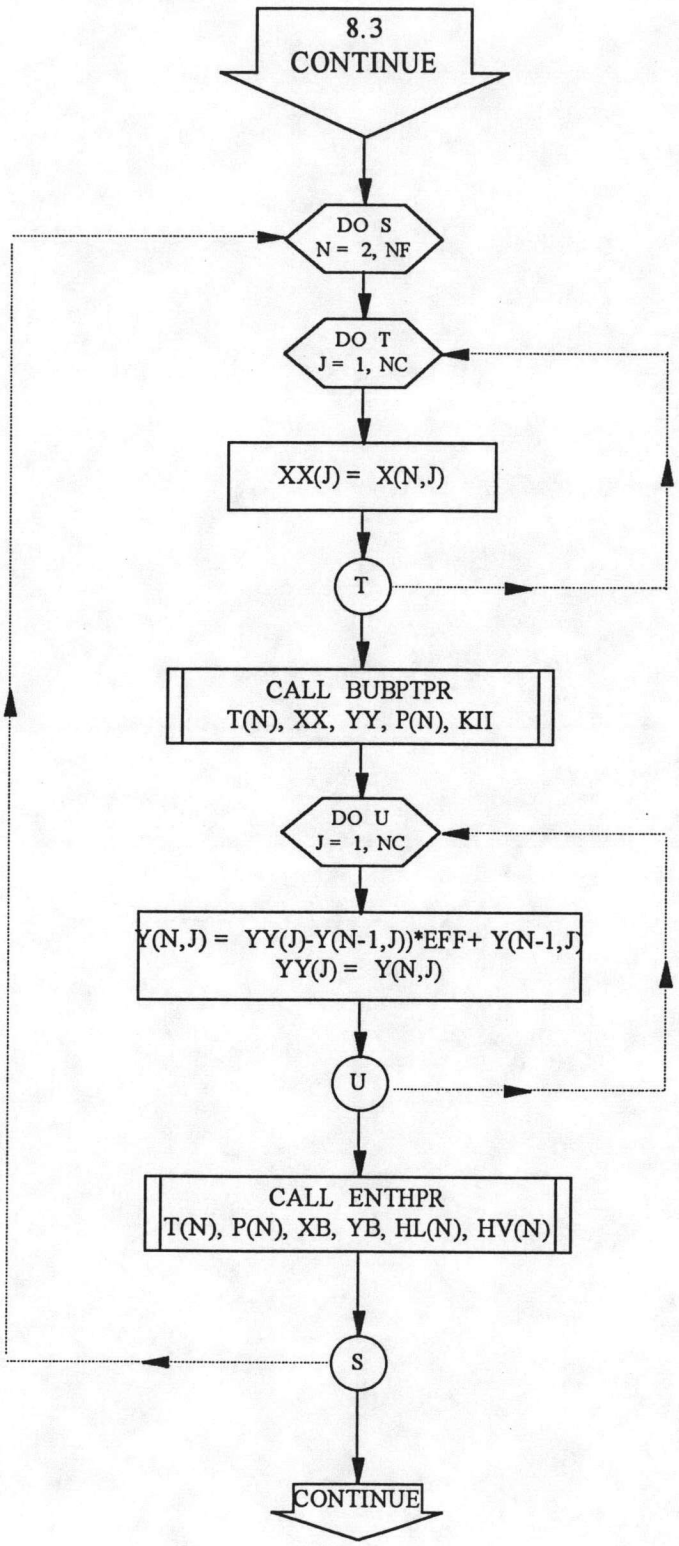
8.3) Running PR model (Continue)



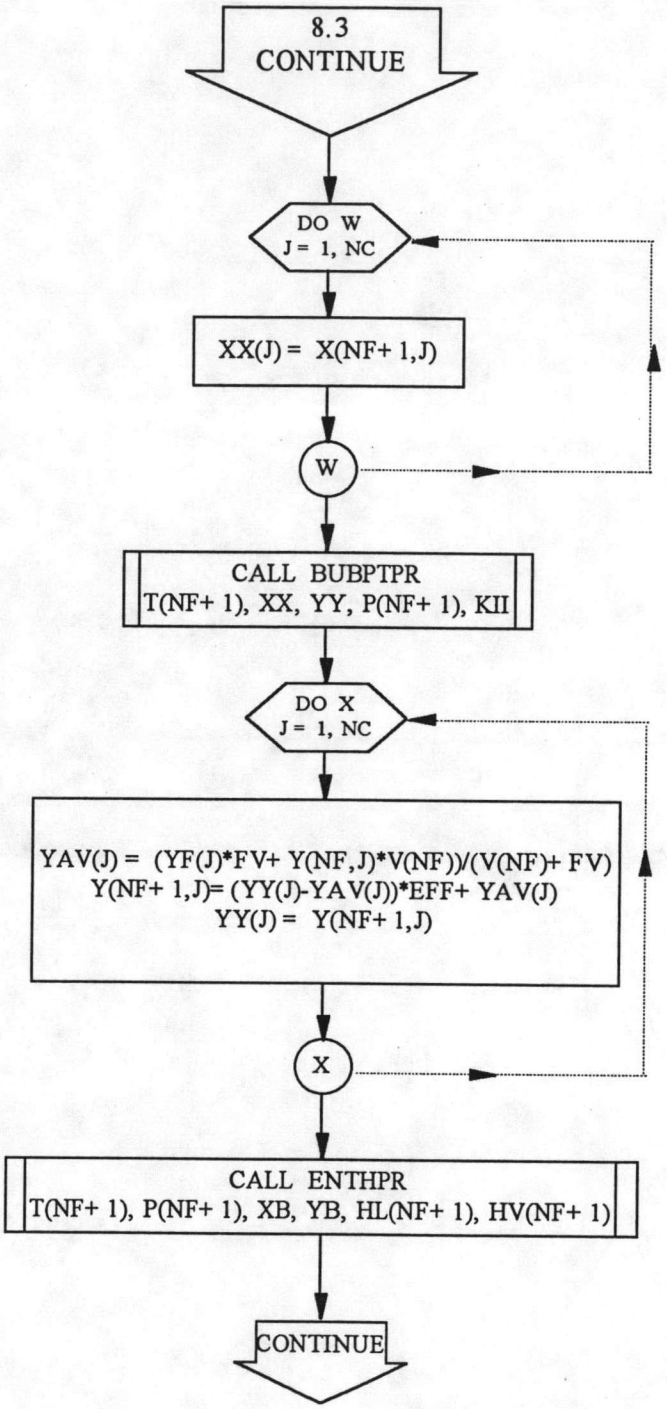
8.3) Running PR model (Continue)



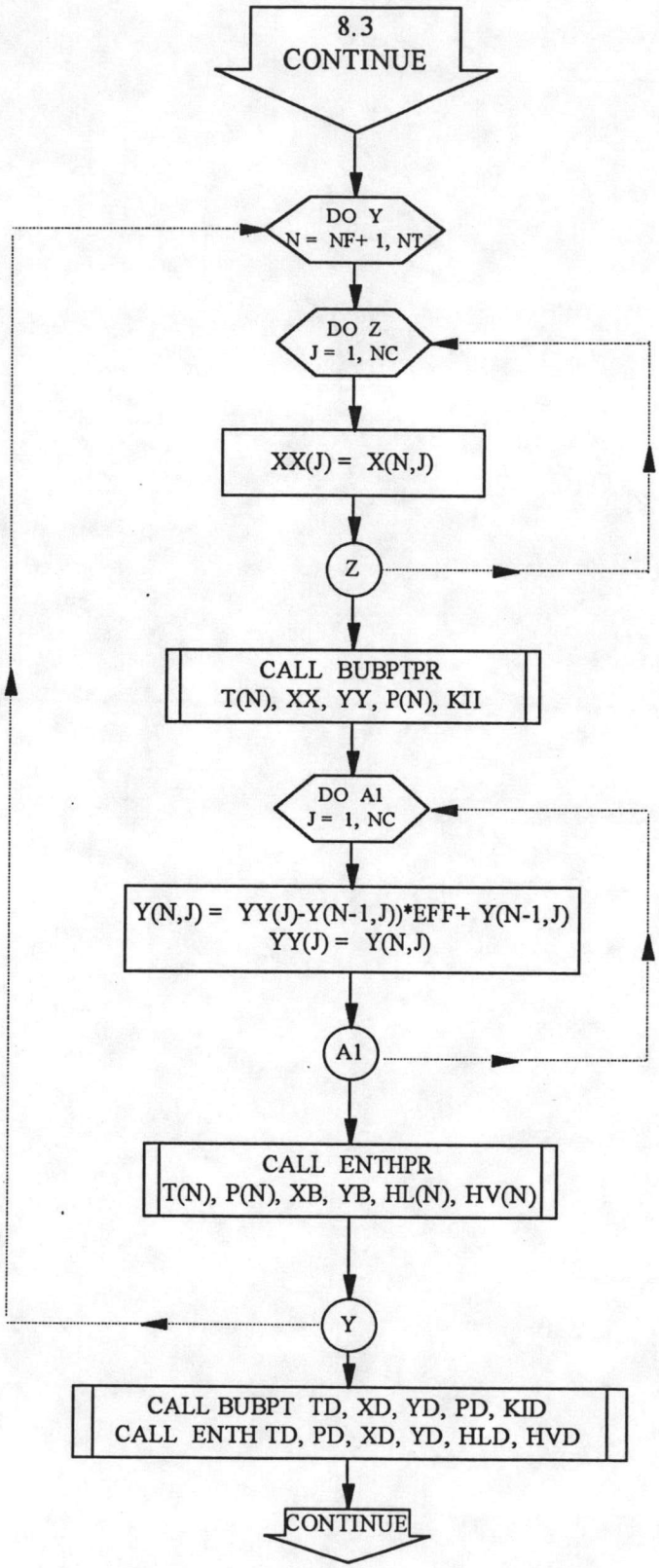
8.3) Running PR model (Continue)



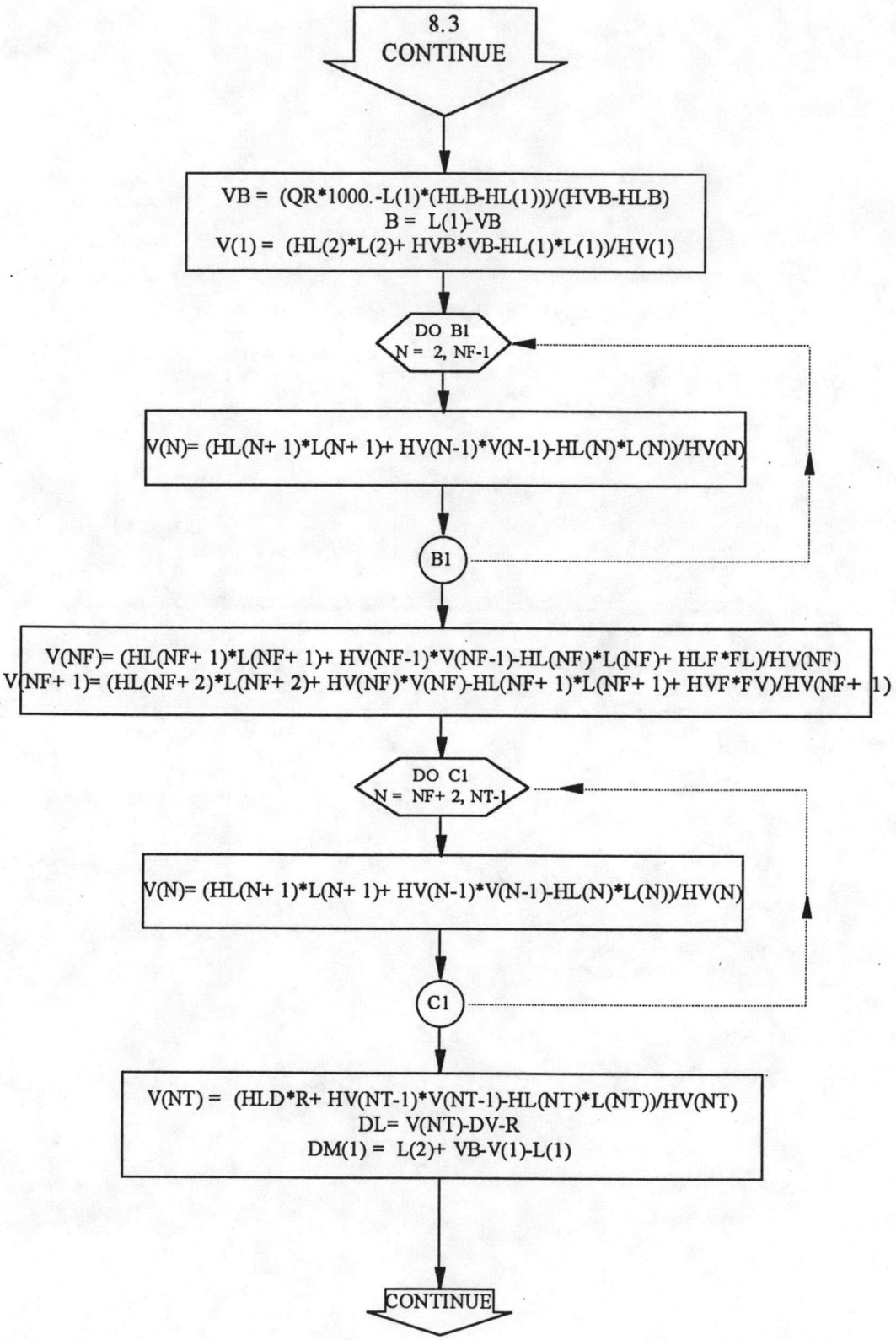
8.3) Running PR model (Continue)



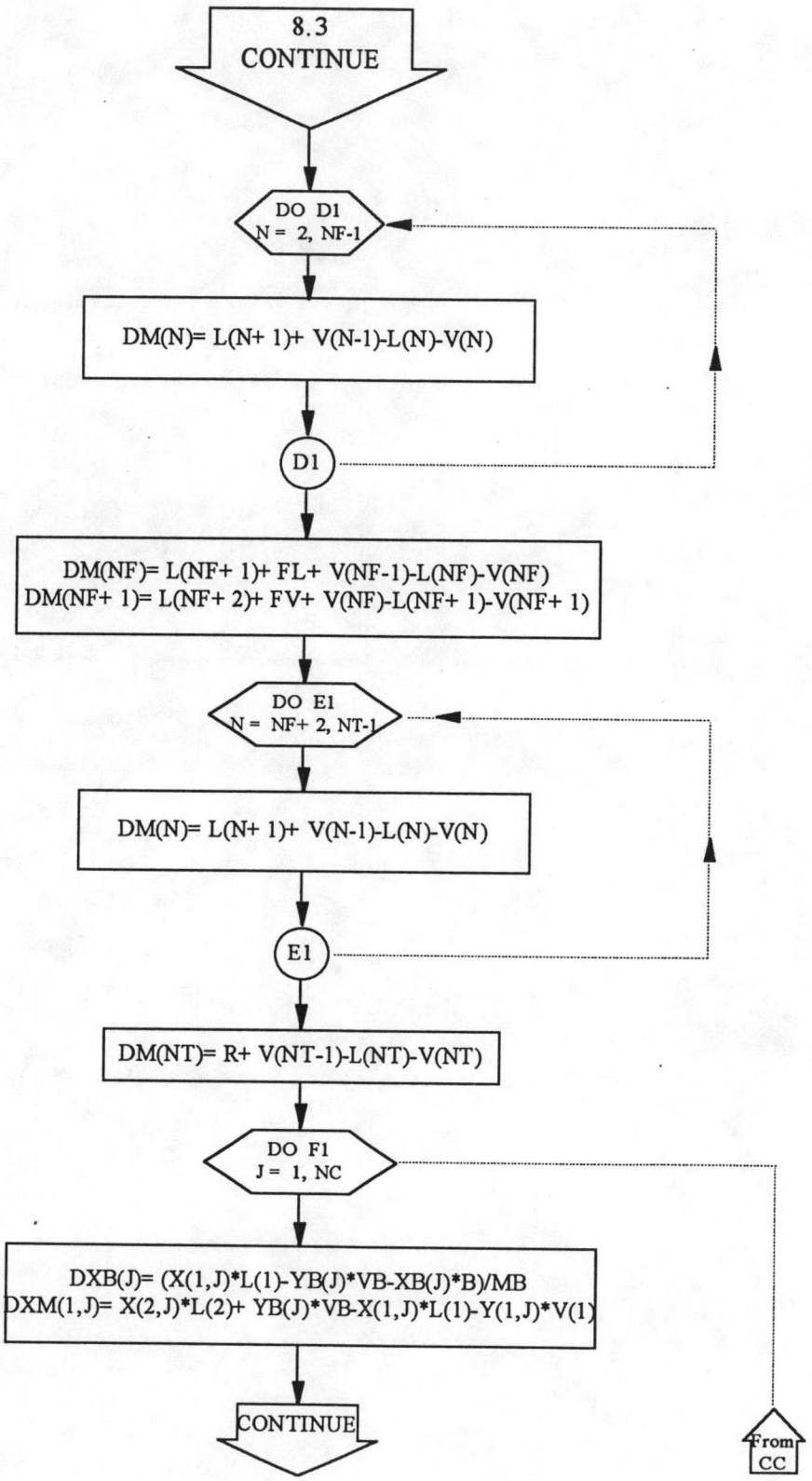
8.3) Running PR model (Continue)



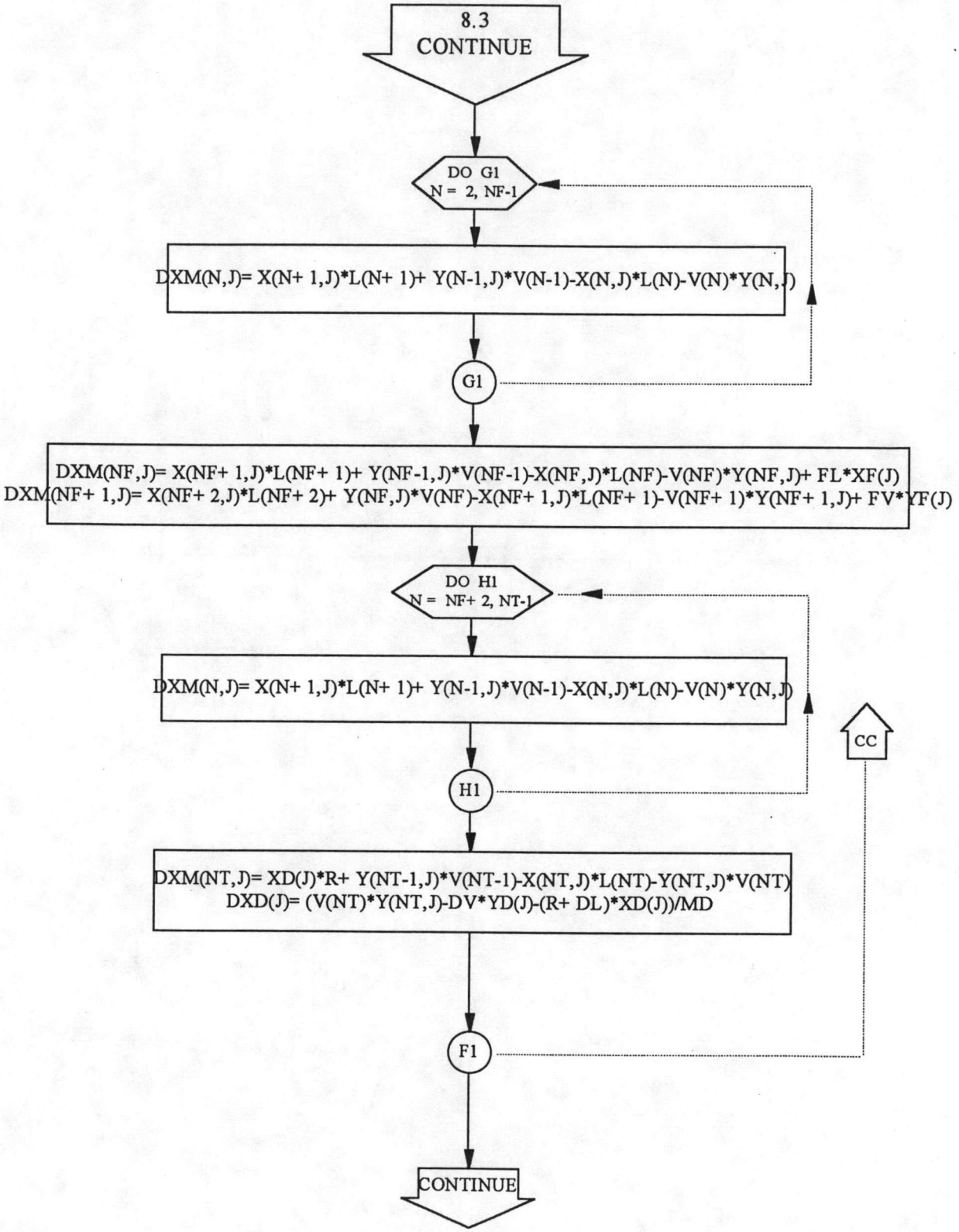
8.3) Running PR model (Continue)



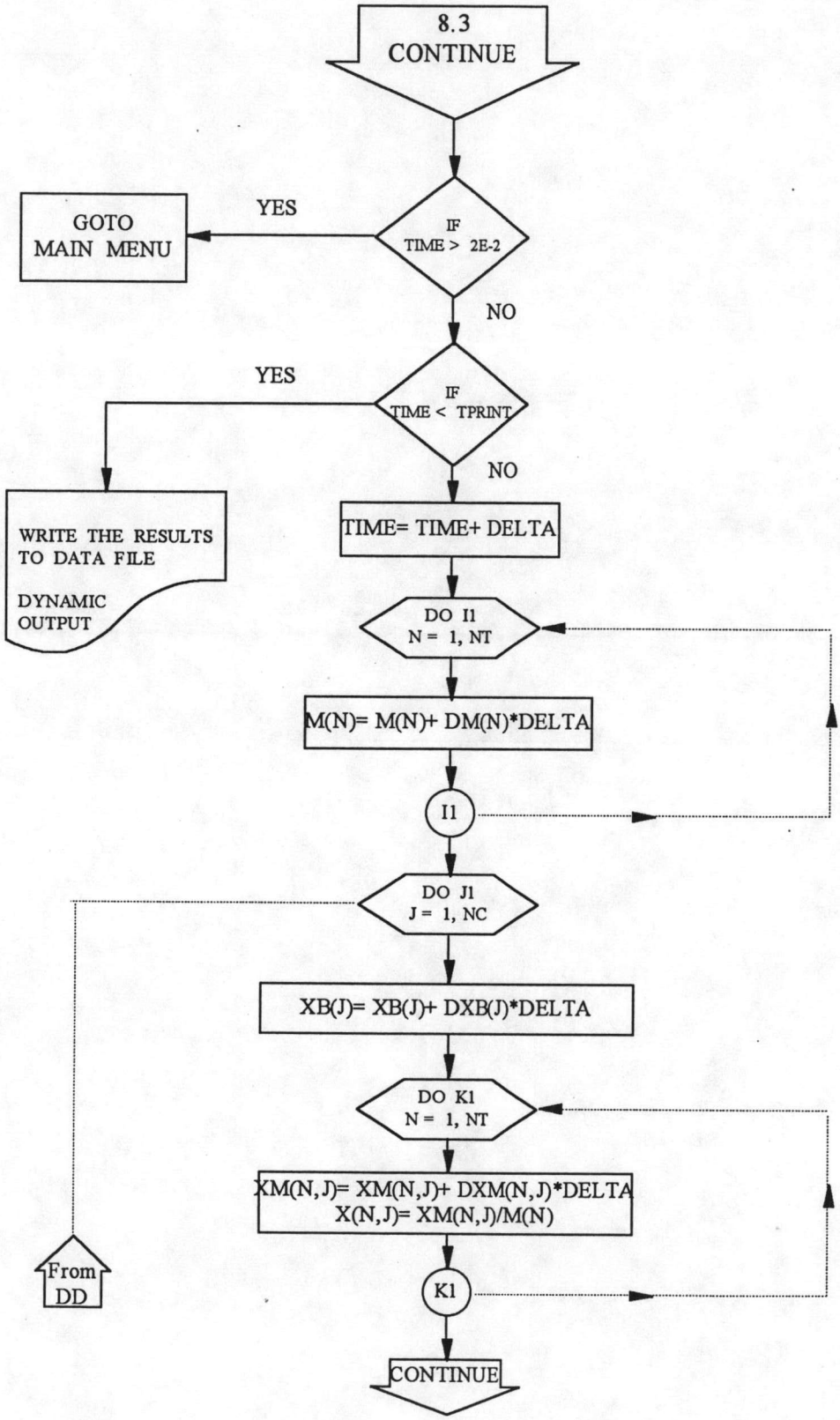
8.3) Running PR model (Continue)



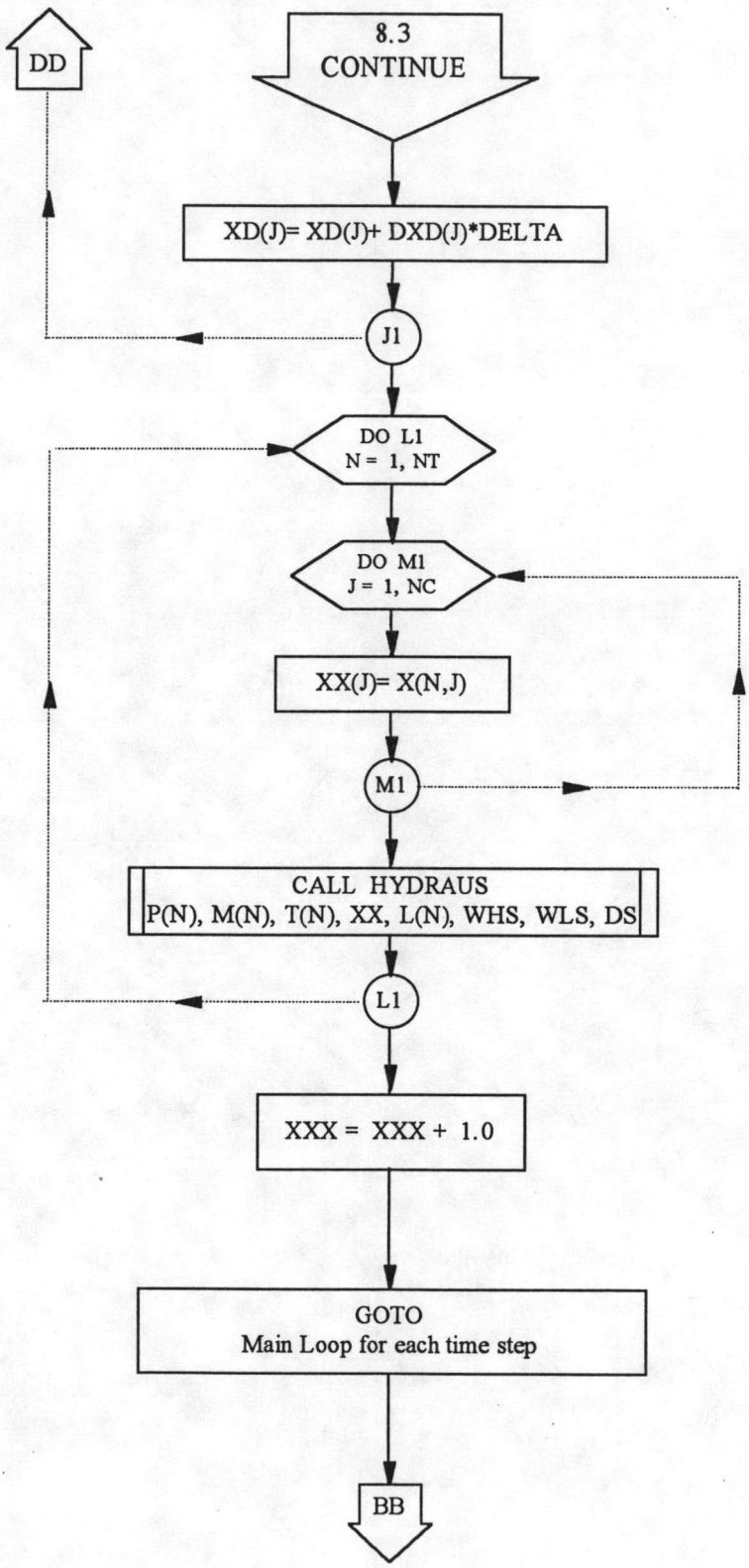
8.3) Running PR model (Continue)



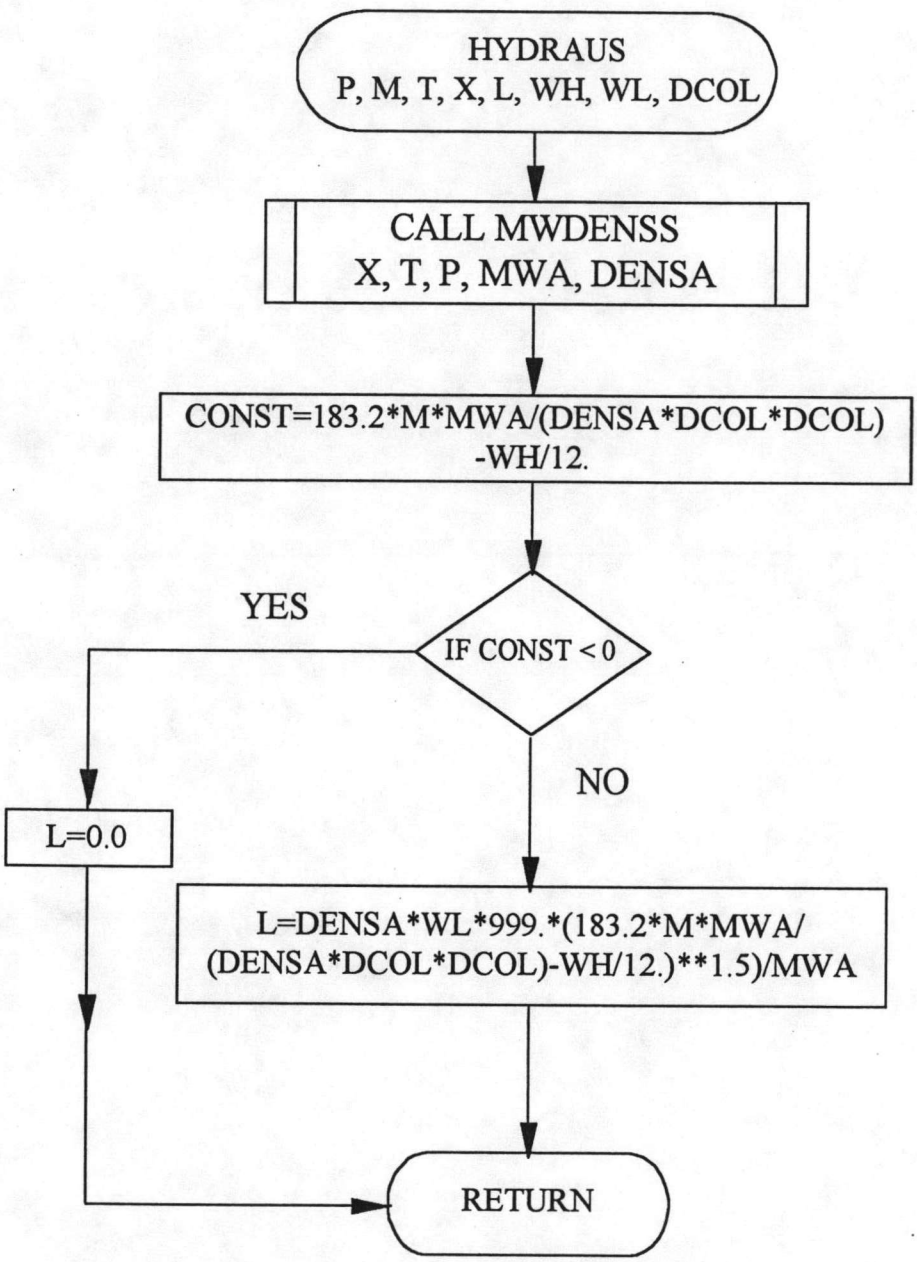
8.3) Running PR model (Continue)



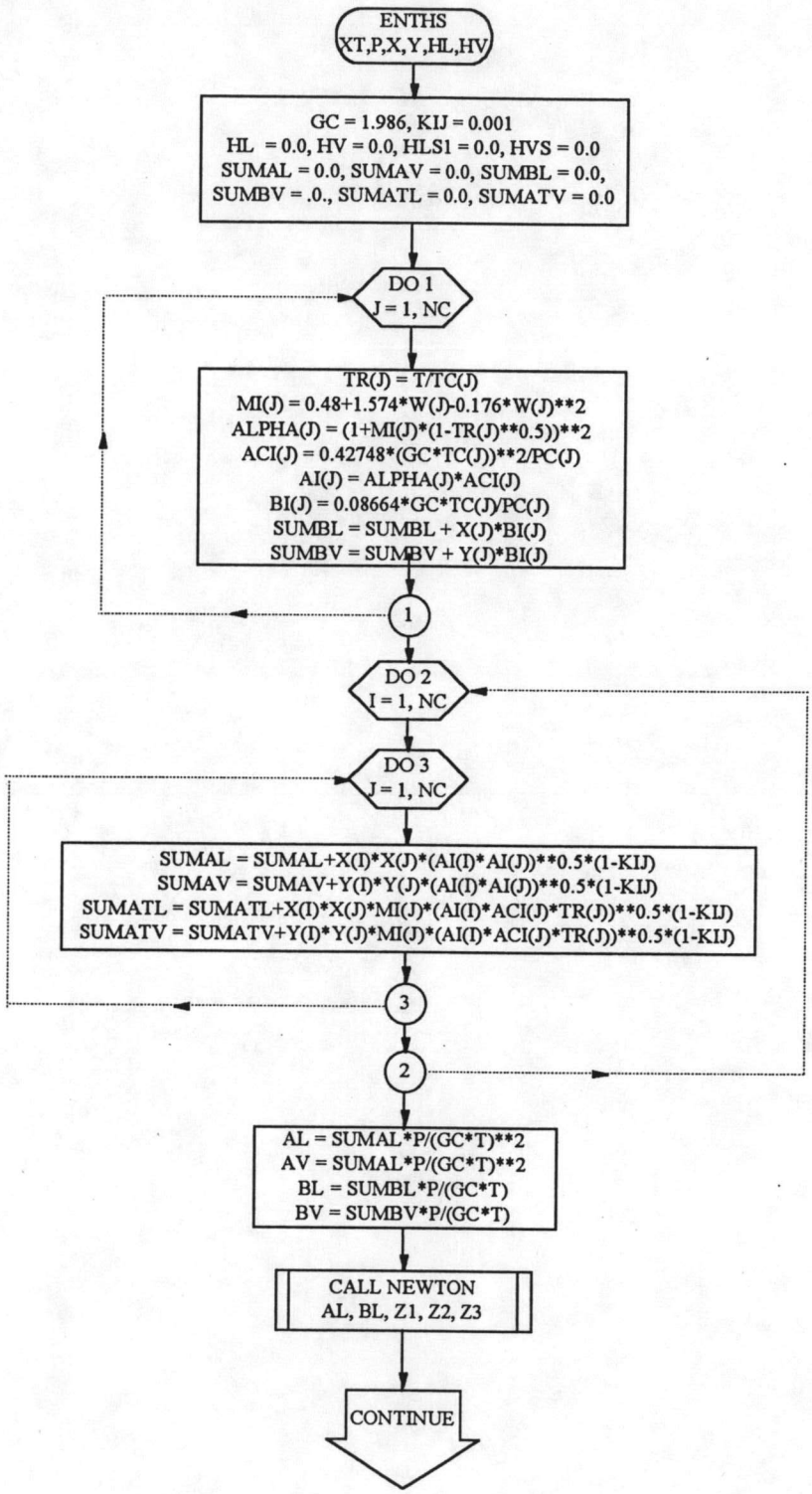
8.3) Running PR model (Continue)



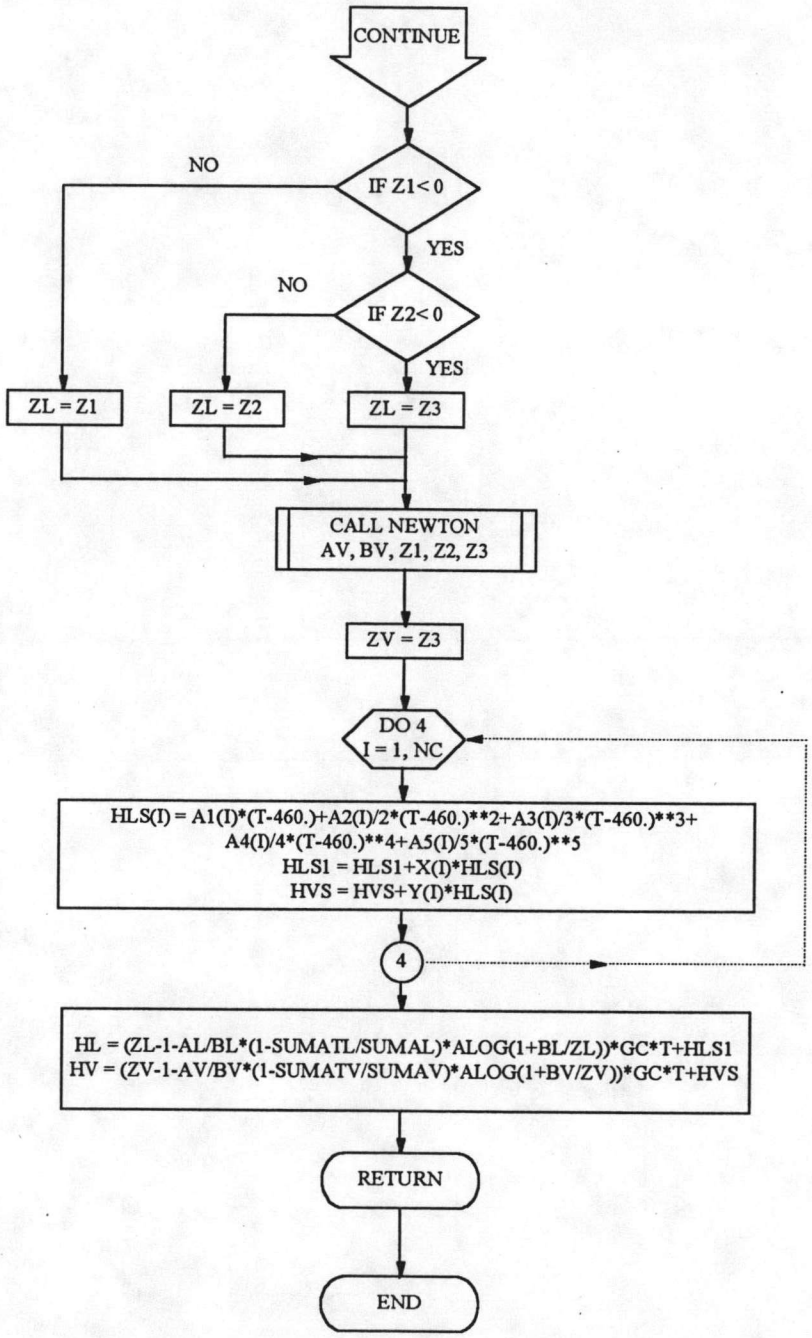
8.3) Running PR model (Continue)



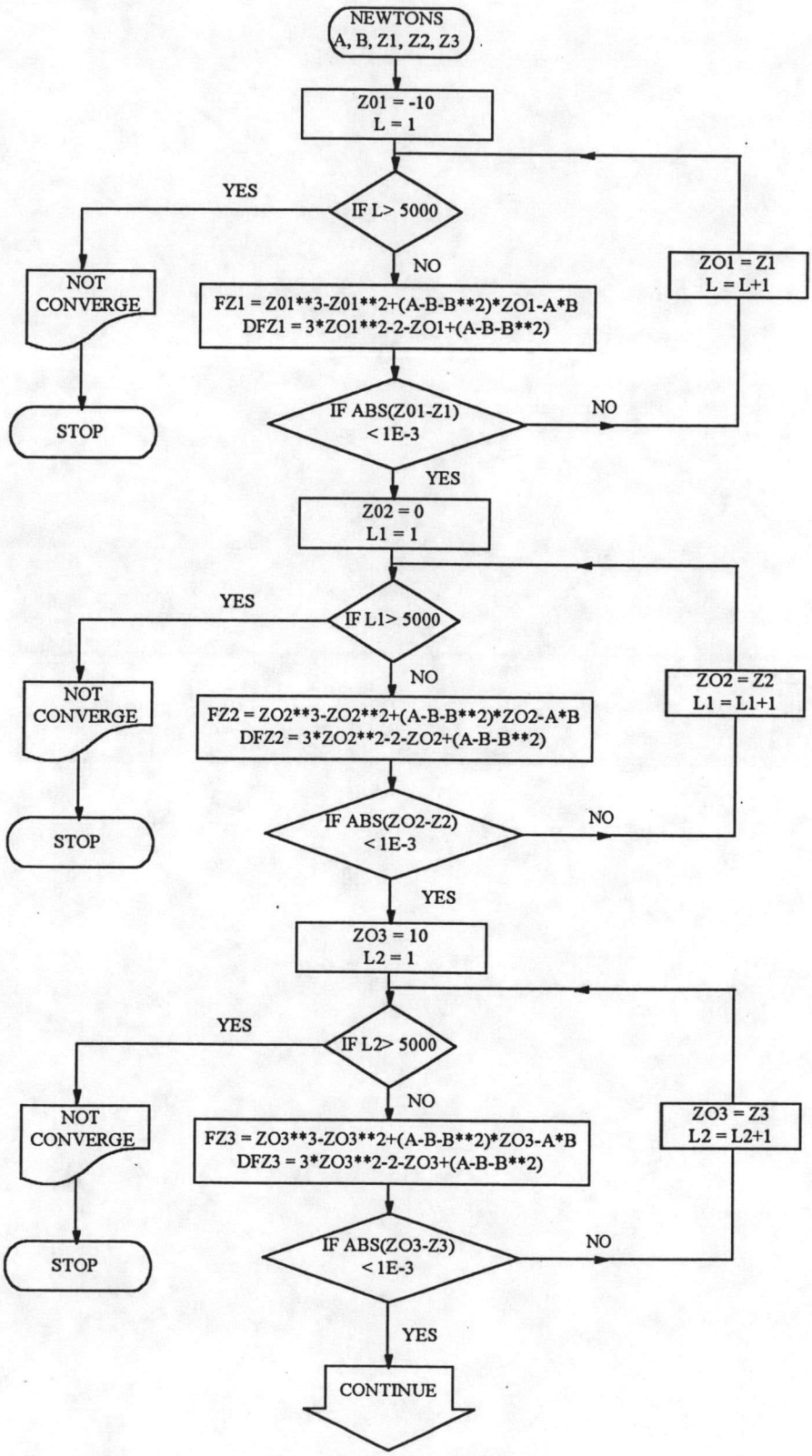
SUBPROGRAM HYDRAUS



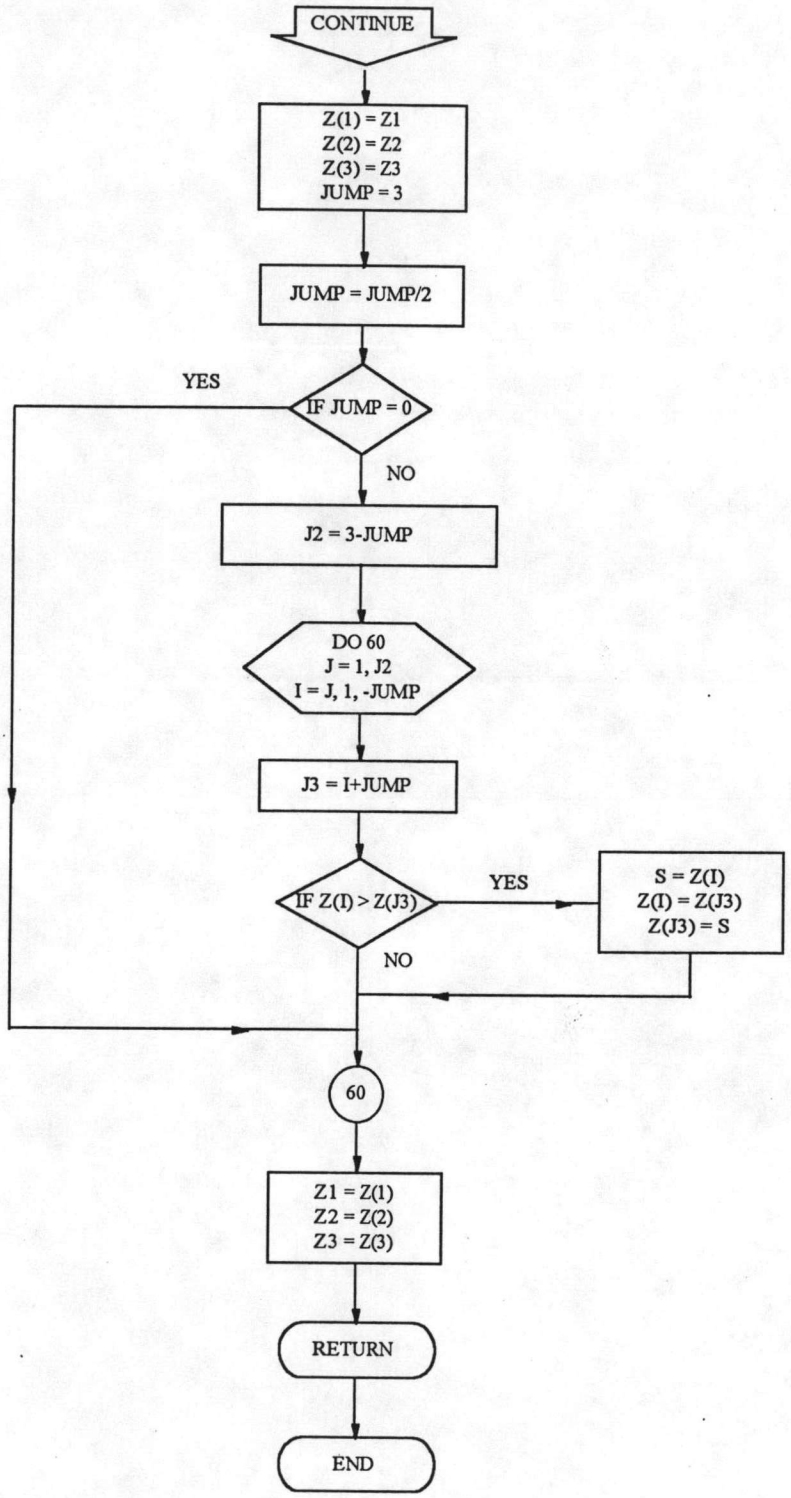
SUBROUTINE ENTHS



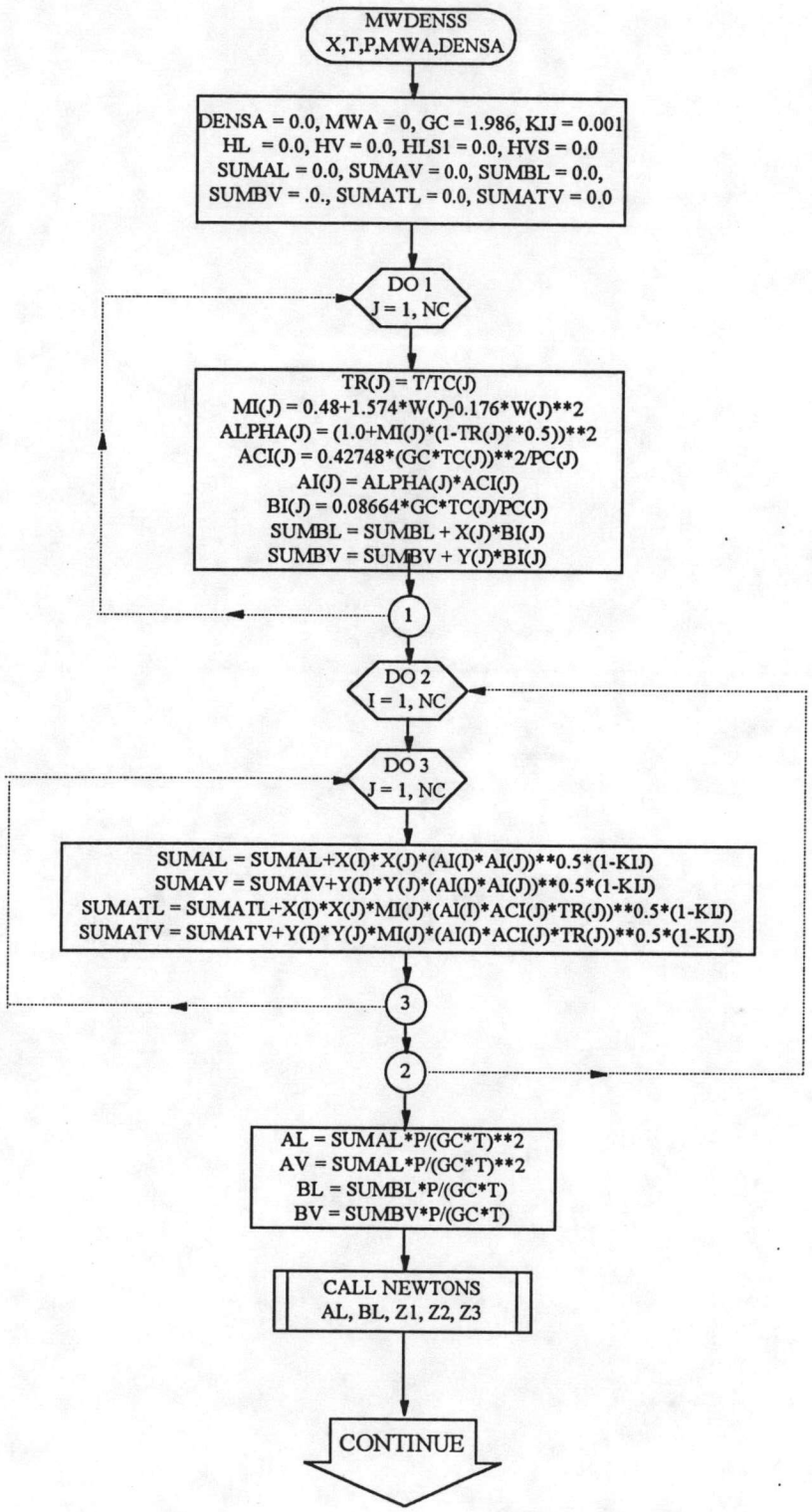
SUBROUTINE ENTHS (Continue)



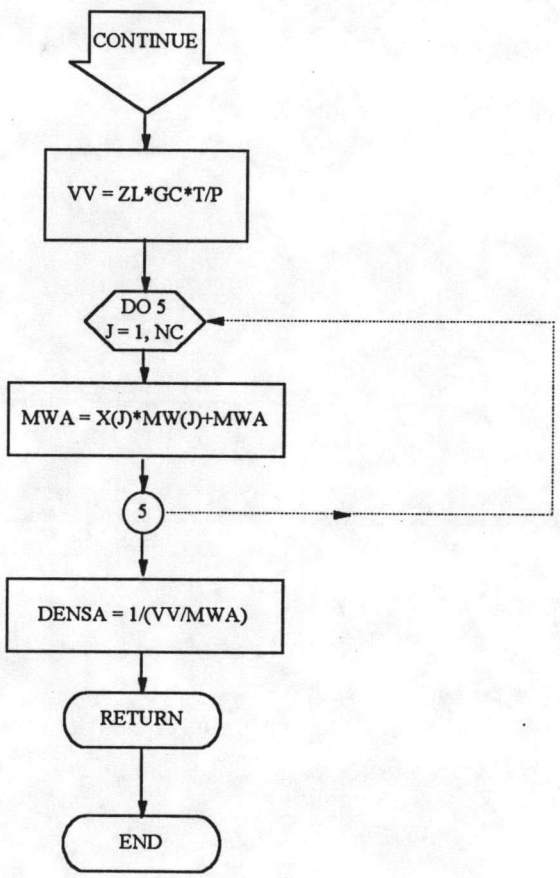
SUBROUTINE NEWTONS



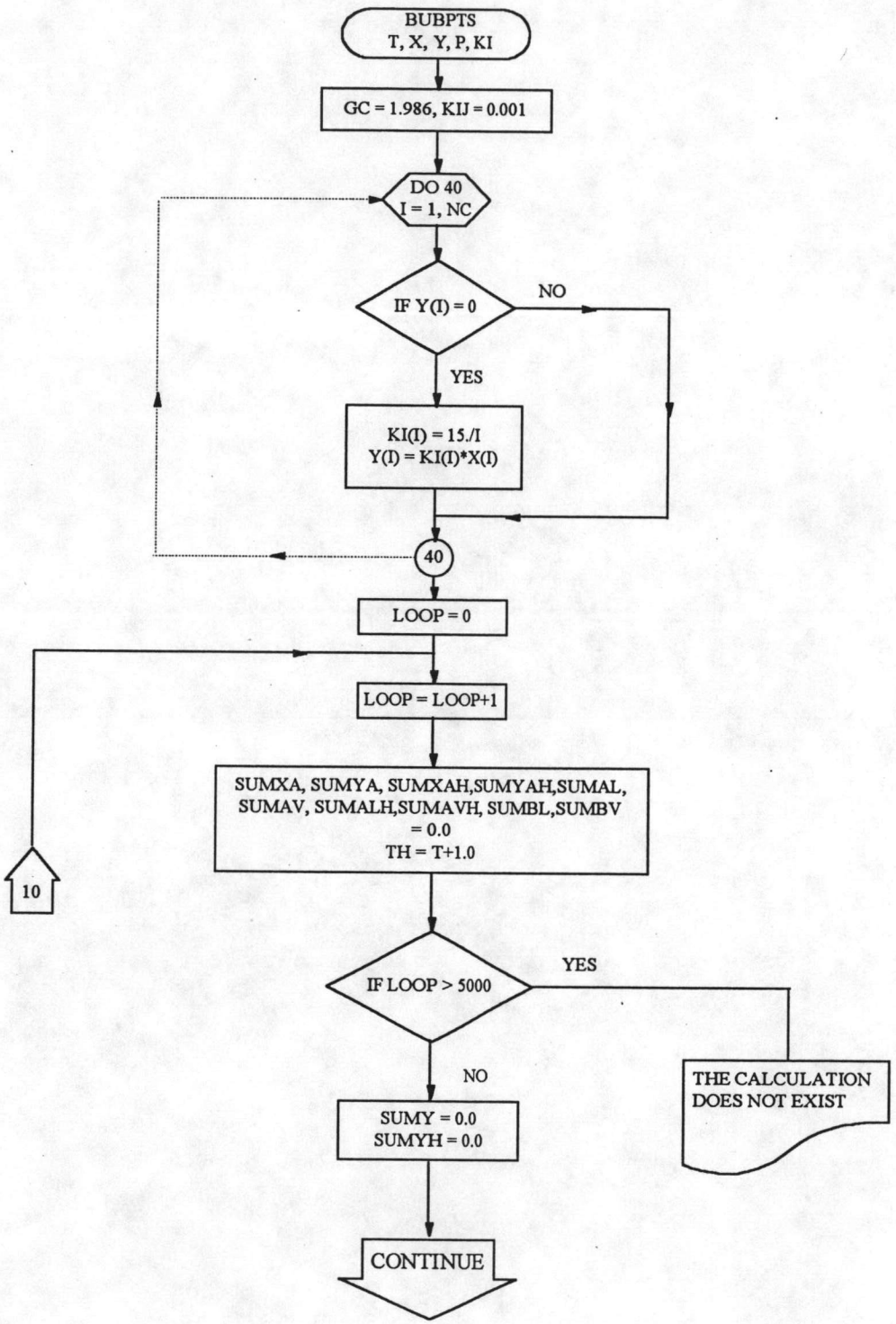
SUBROUTINE NEWTONS (Continue)



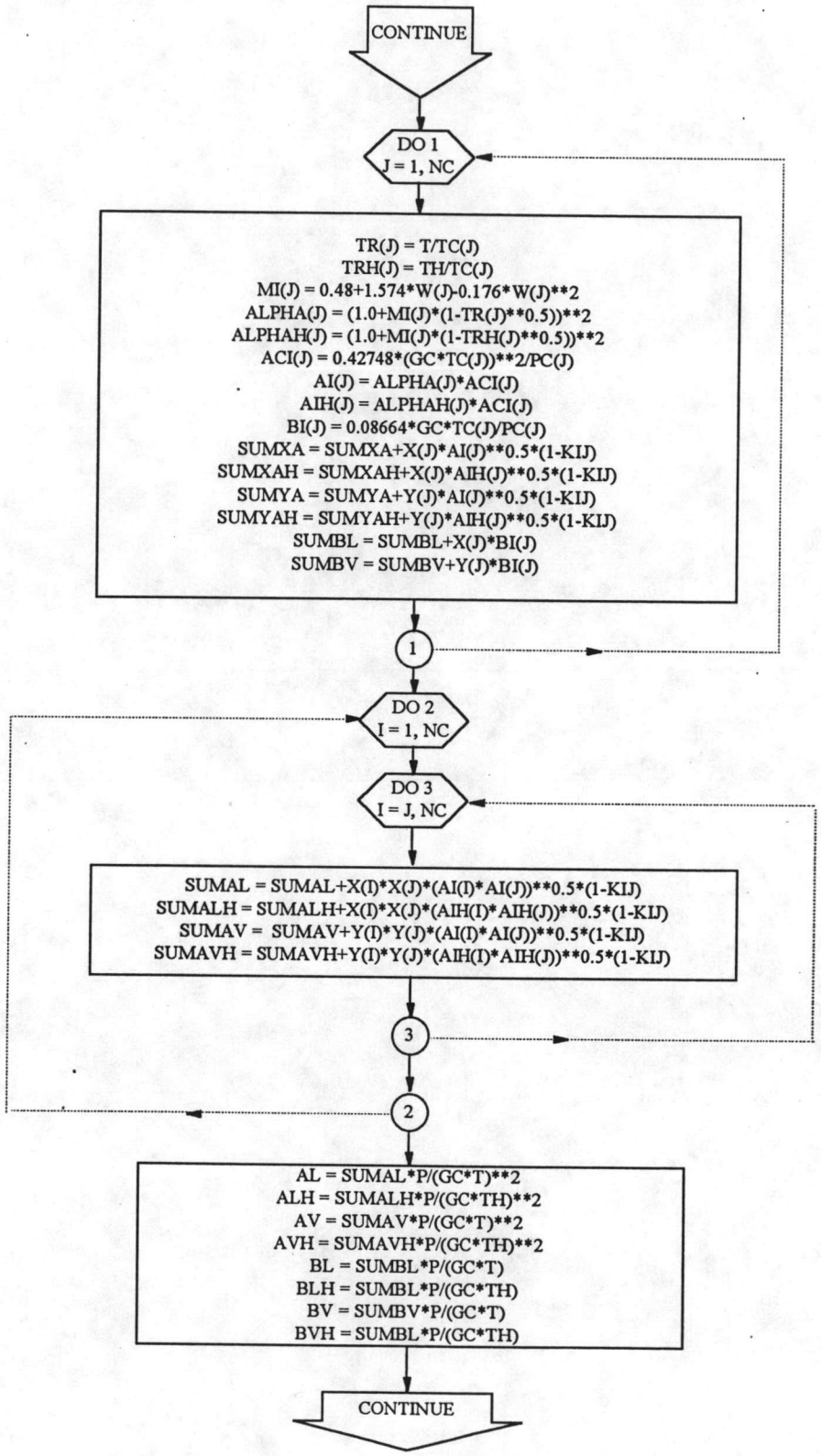
SUBROUTINE MWDENSS



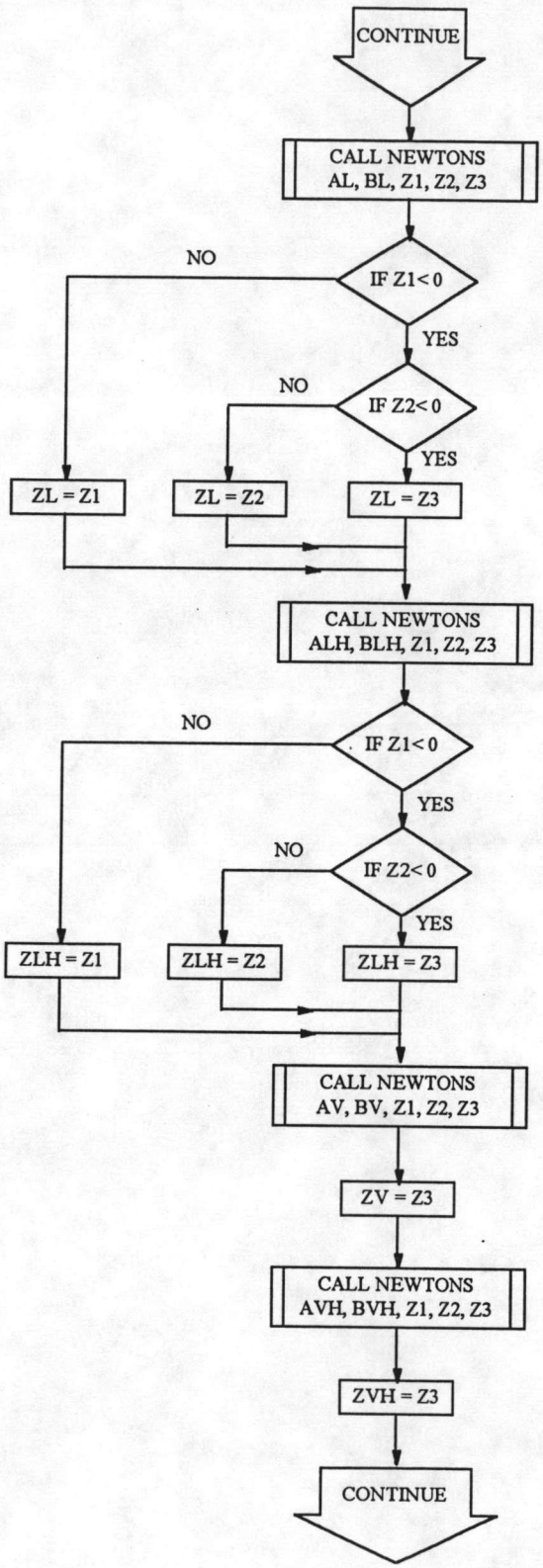
SUBROUTINE MWDENSS (Continue)



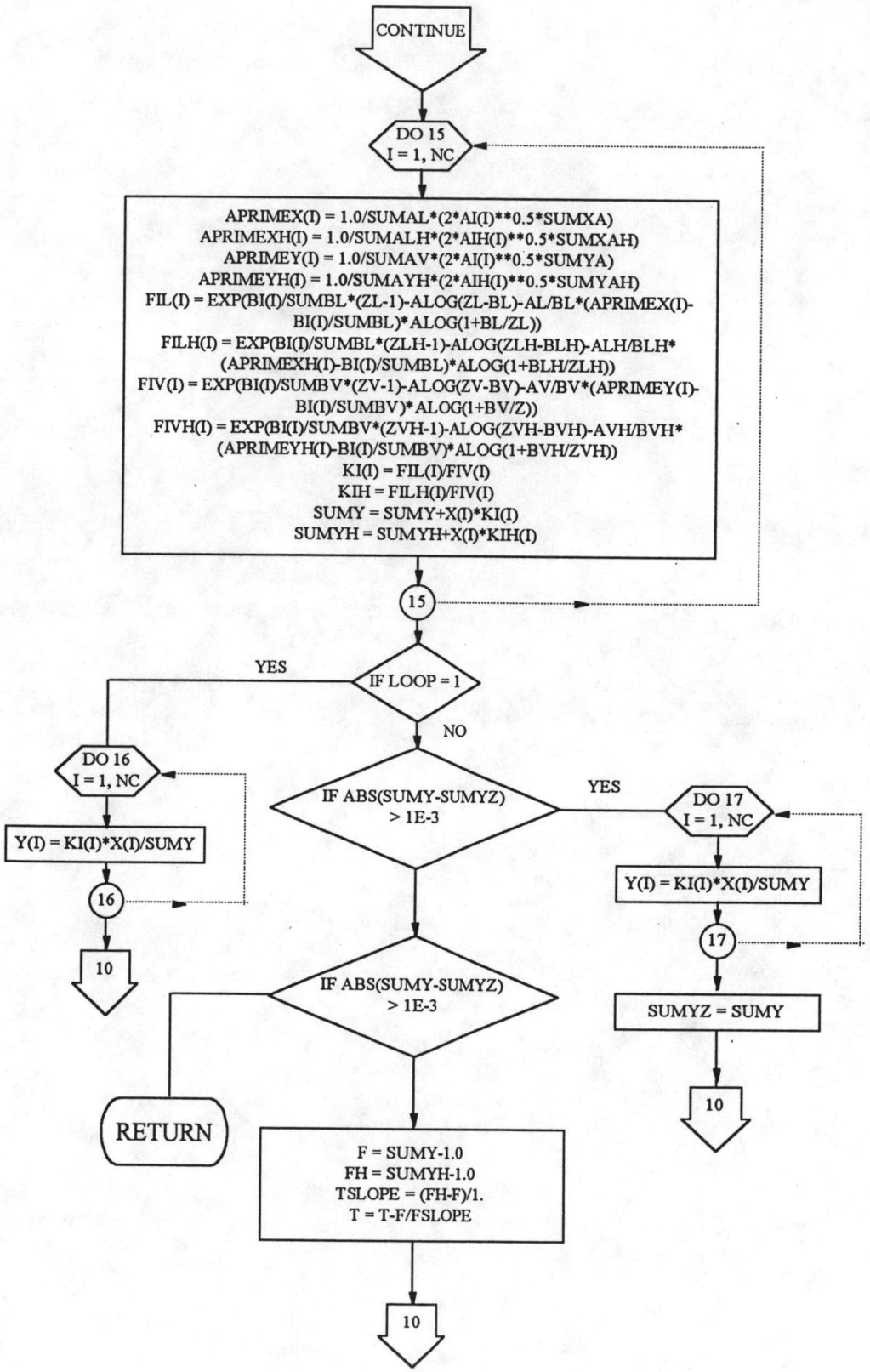
SUBROUTINE BUBPTS



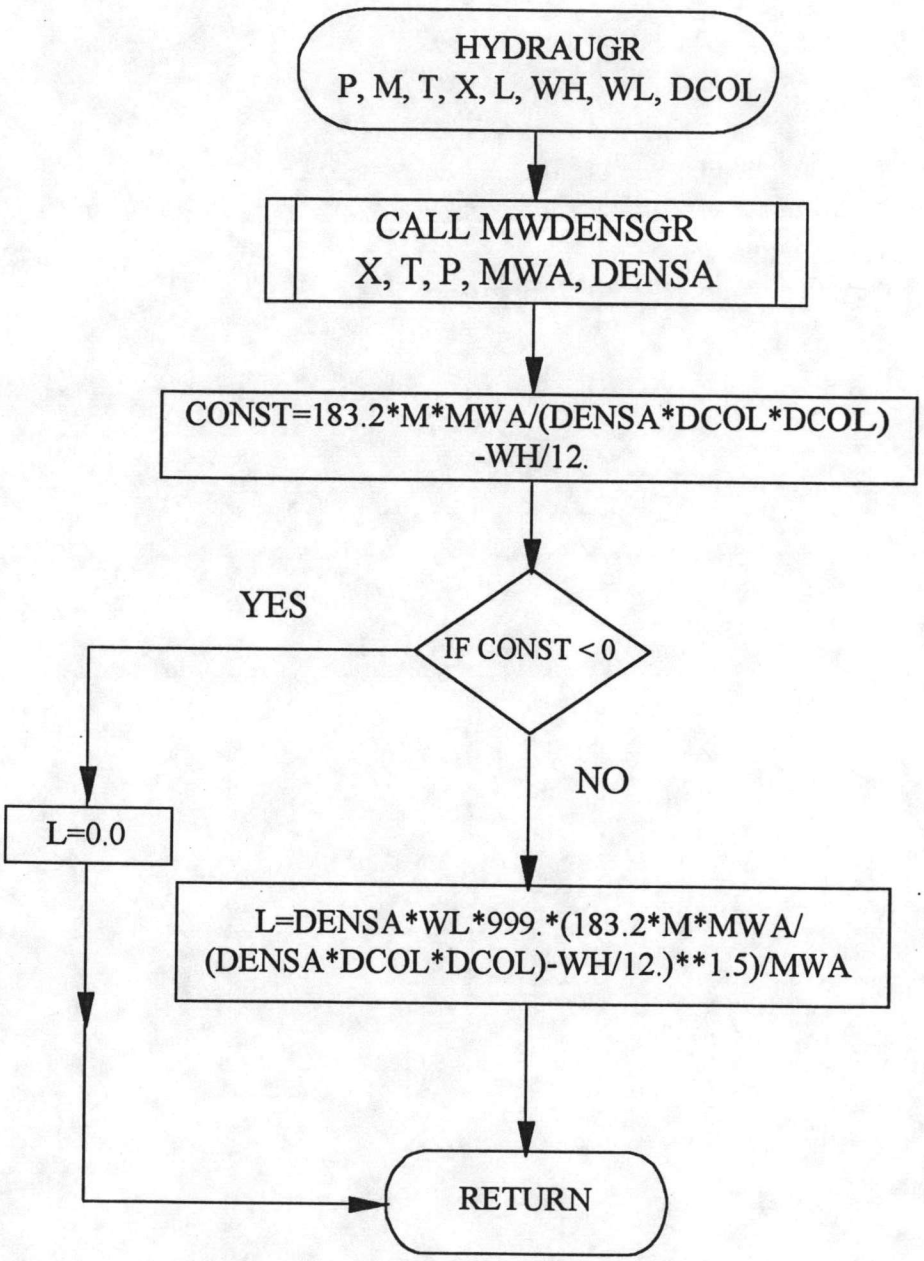
SUBROUTINE BUBPTS (Continue)



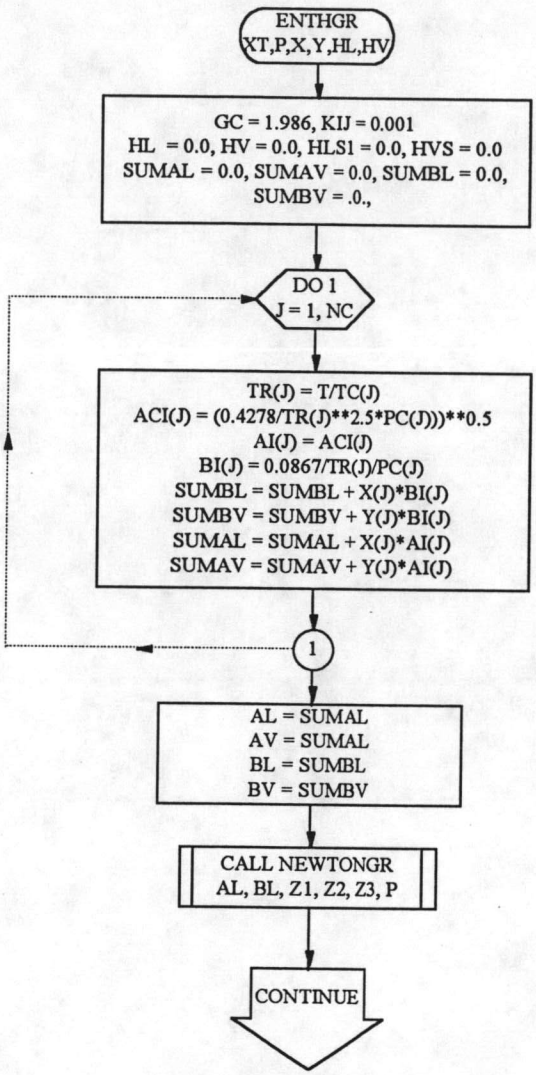
SUBROUTINE BUBPTS (Continue)



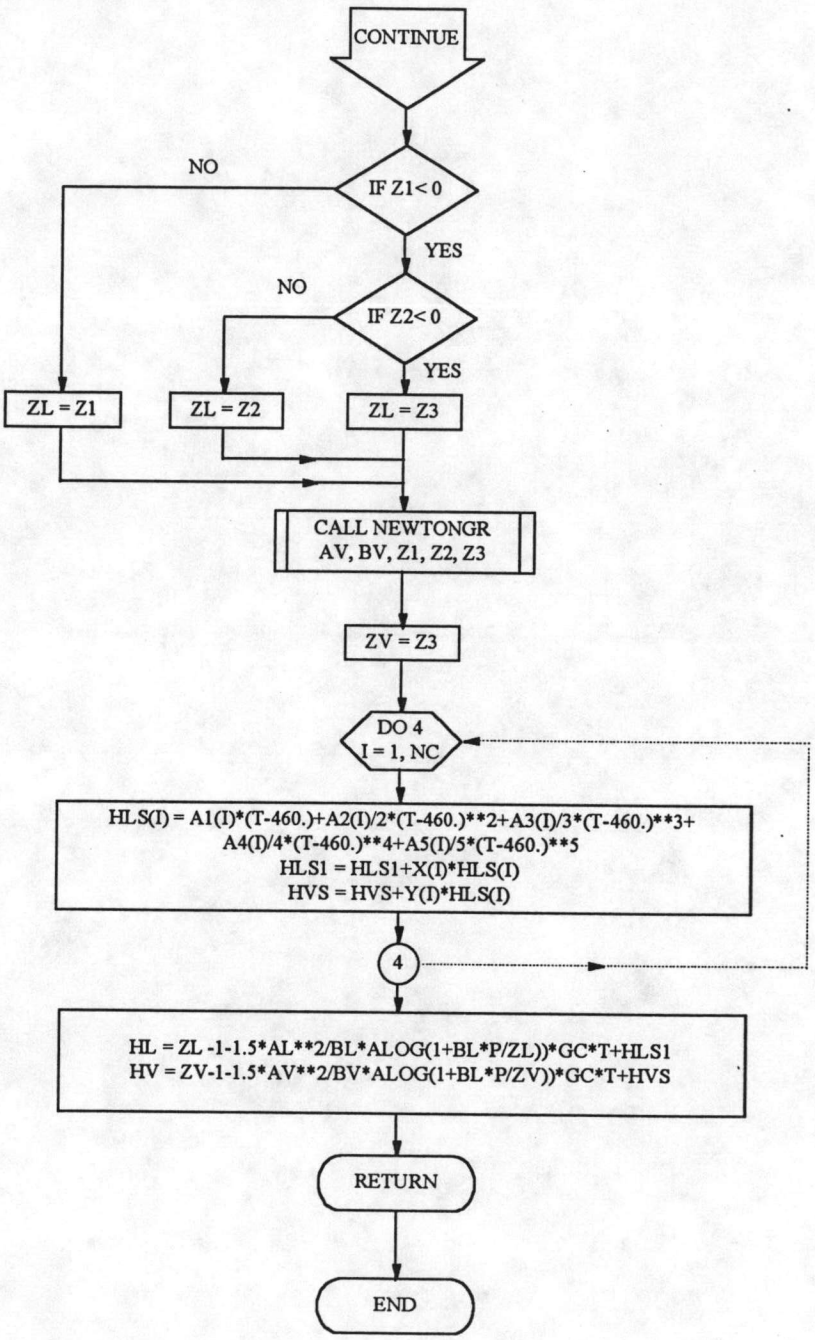
SUBROUTINE BUBPTS (Continue)



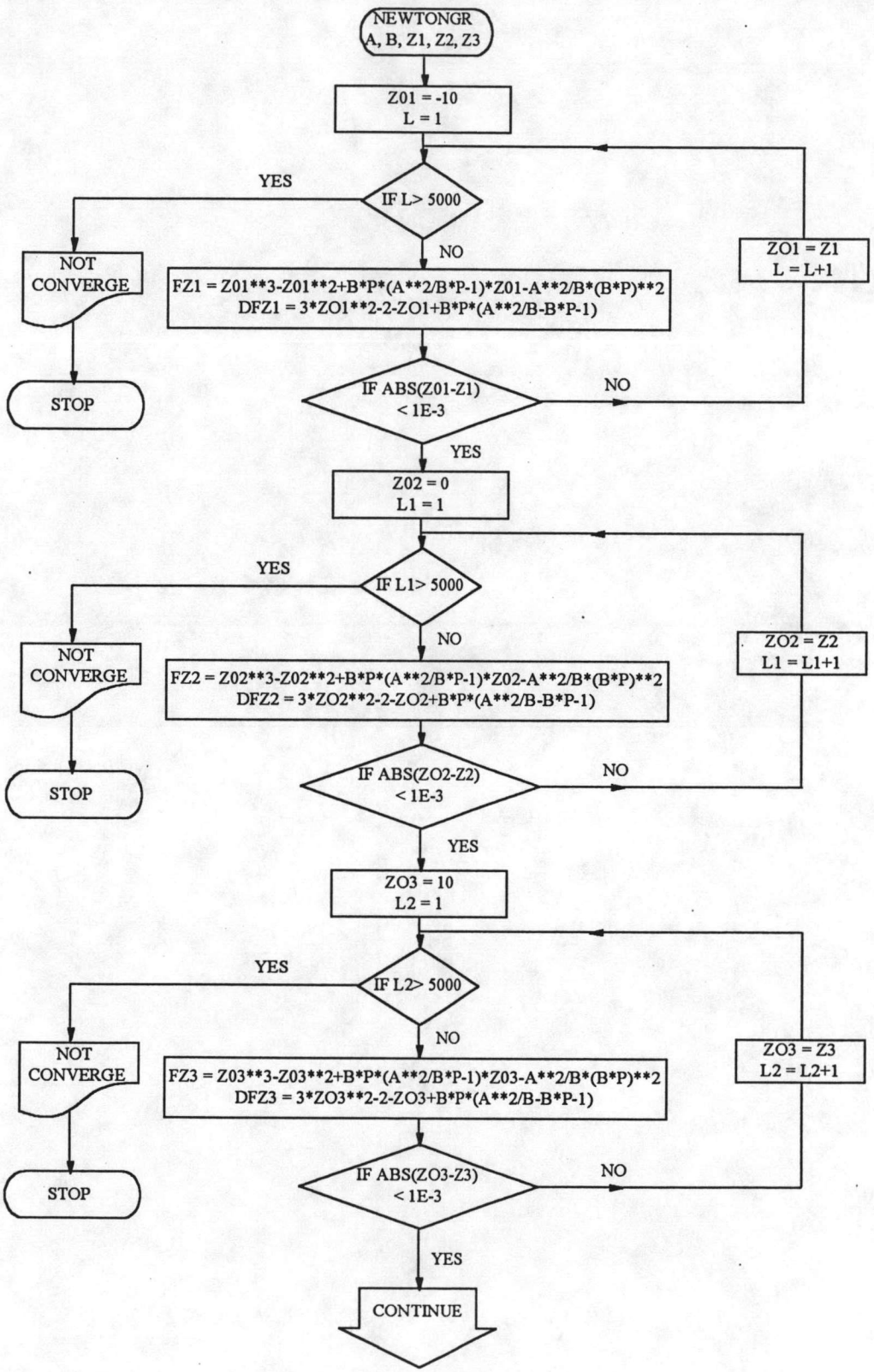
SUBPROGRAM HYDRAUGR



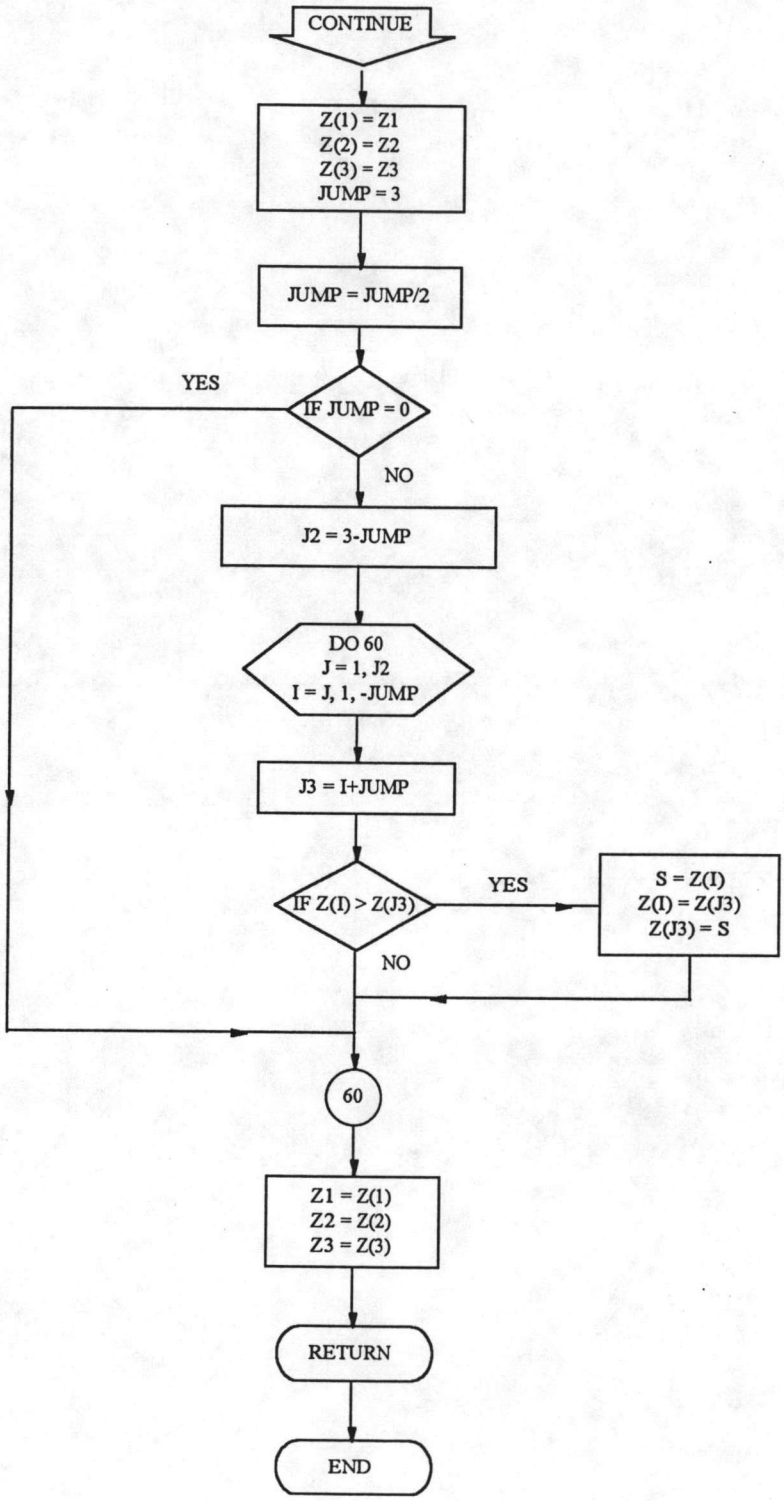
SUBROUTINE ENTHGR



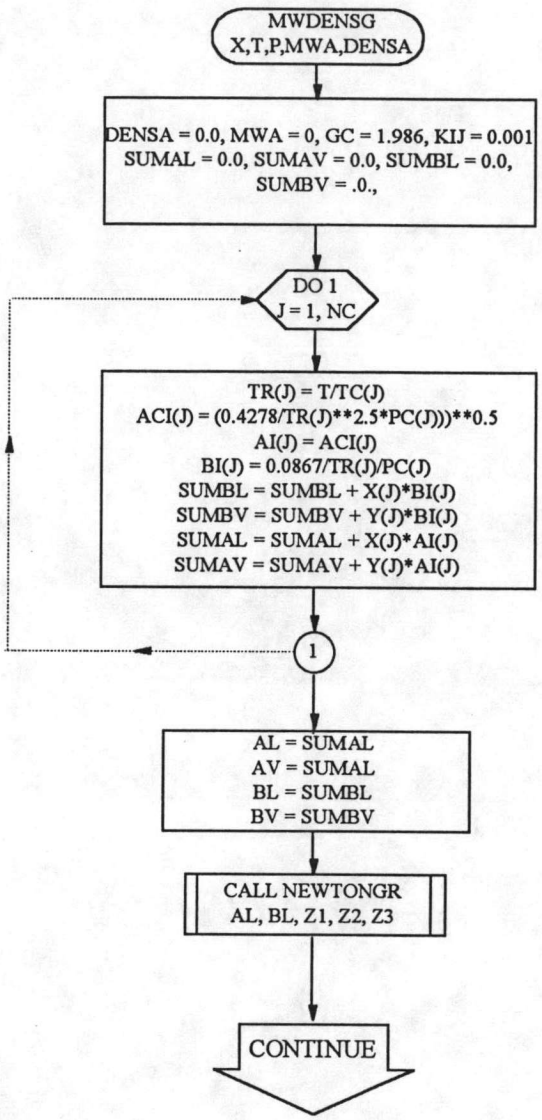
SUBROUTINE ENTHGR (Continue)



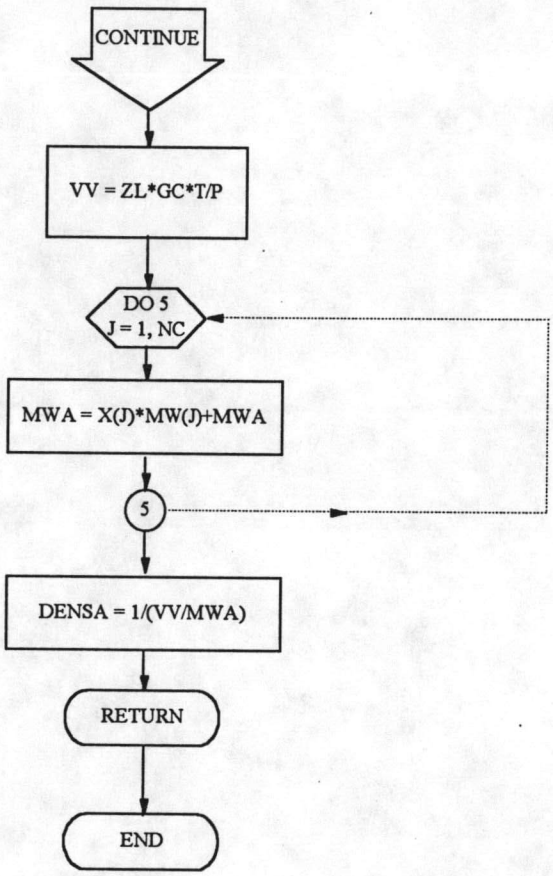
SUBROUTINE NEWTONGR



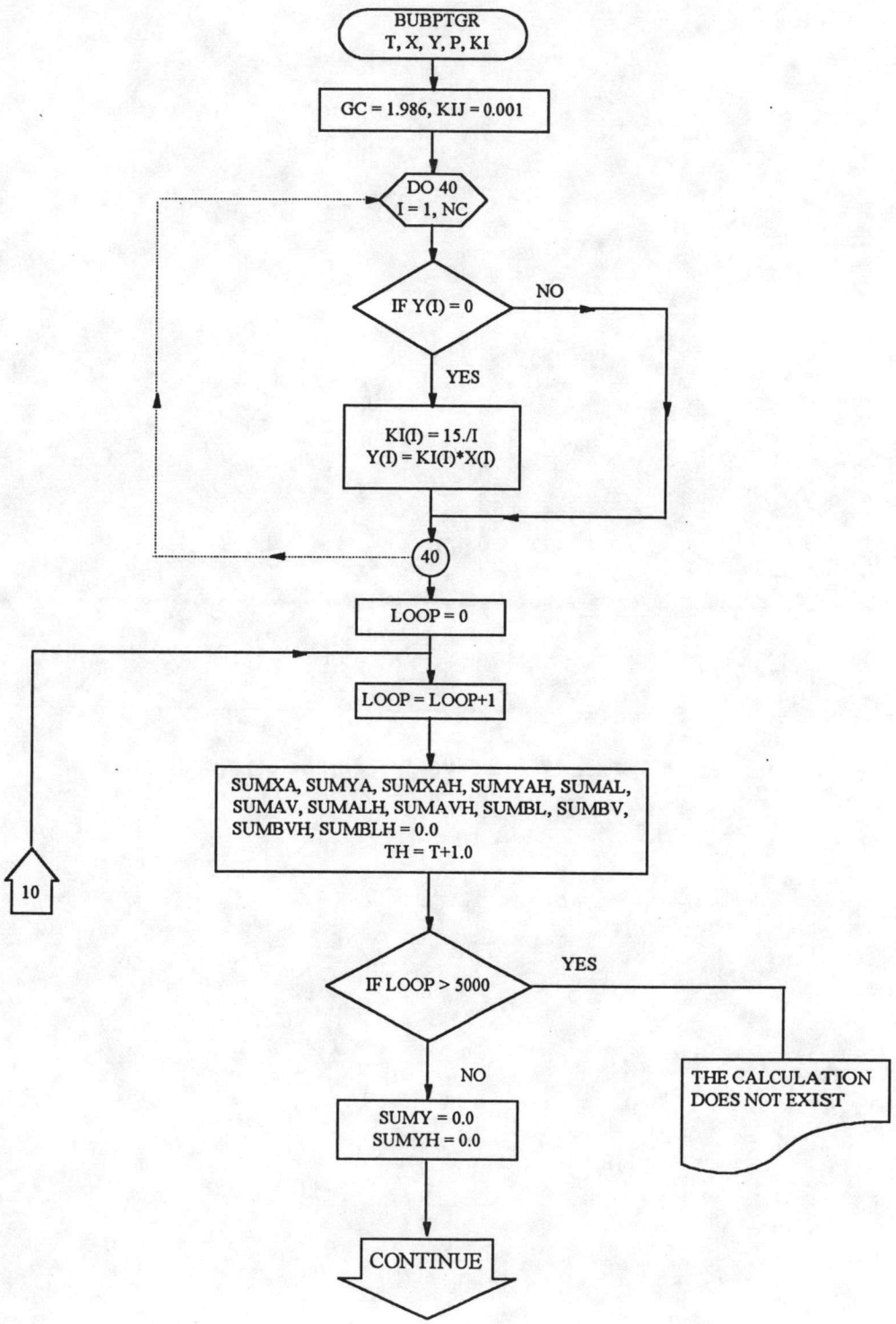
SUBROUTINE NEWTONGR (Continue)



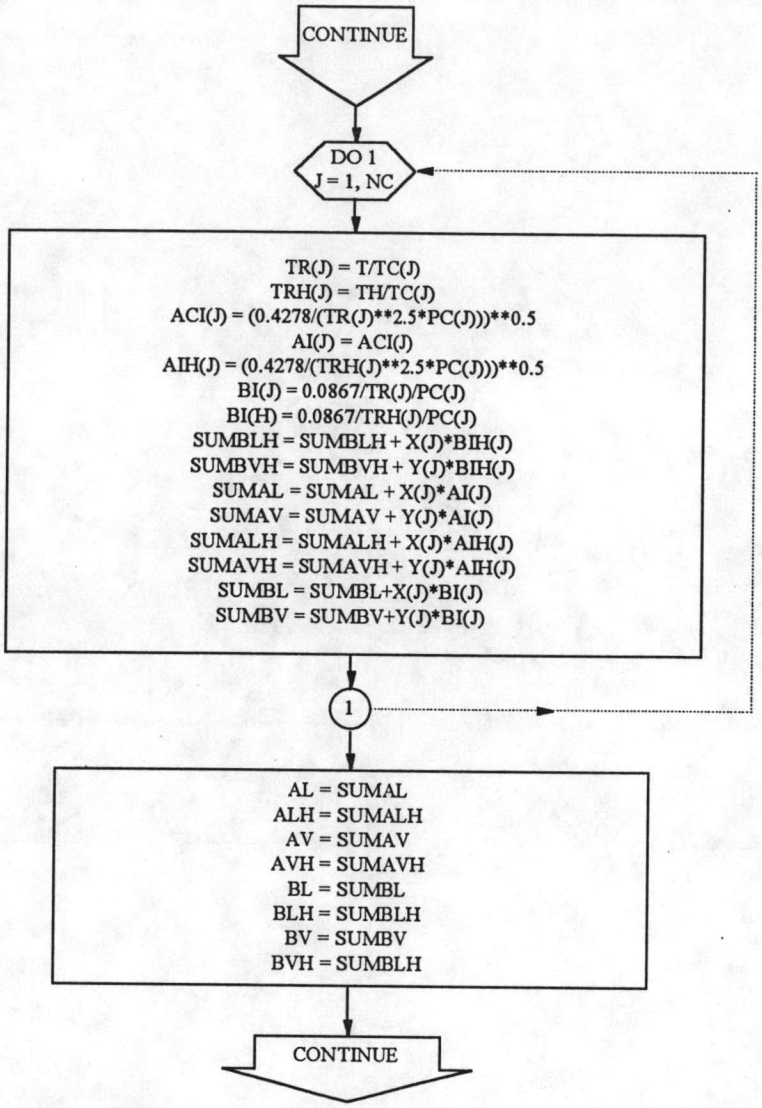
SUBROUTINE MWDENSG



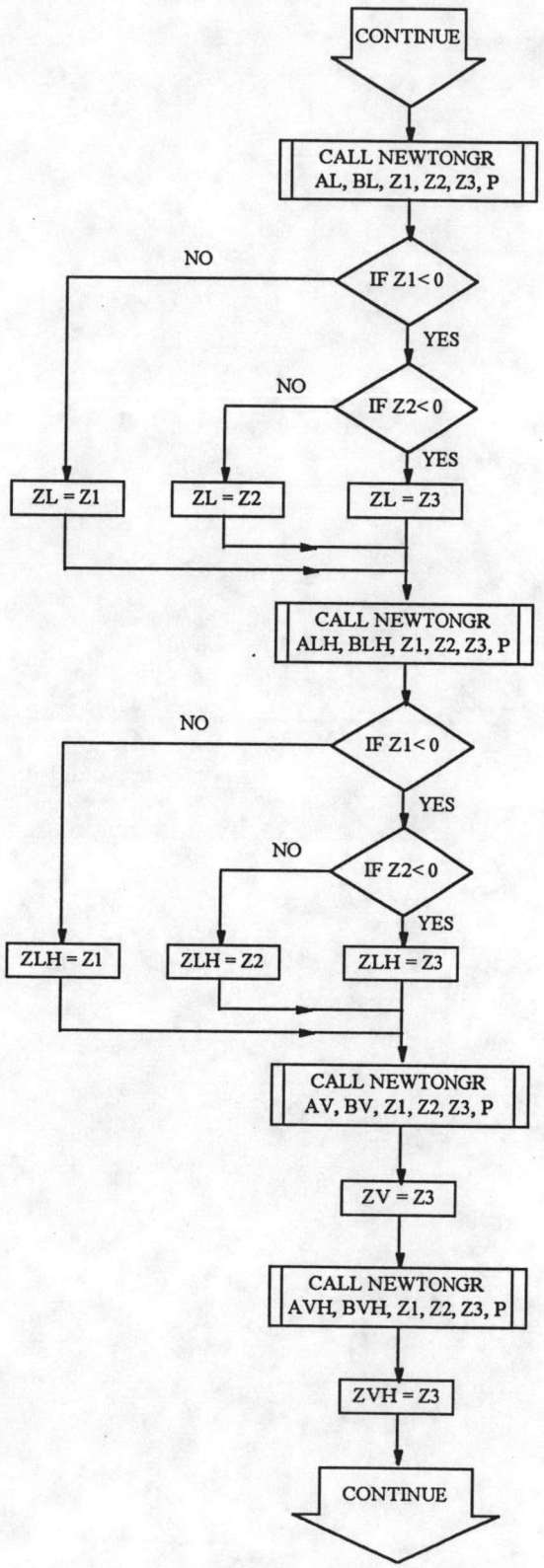
SUBROUTINE MWDENSG (Continue)



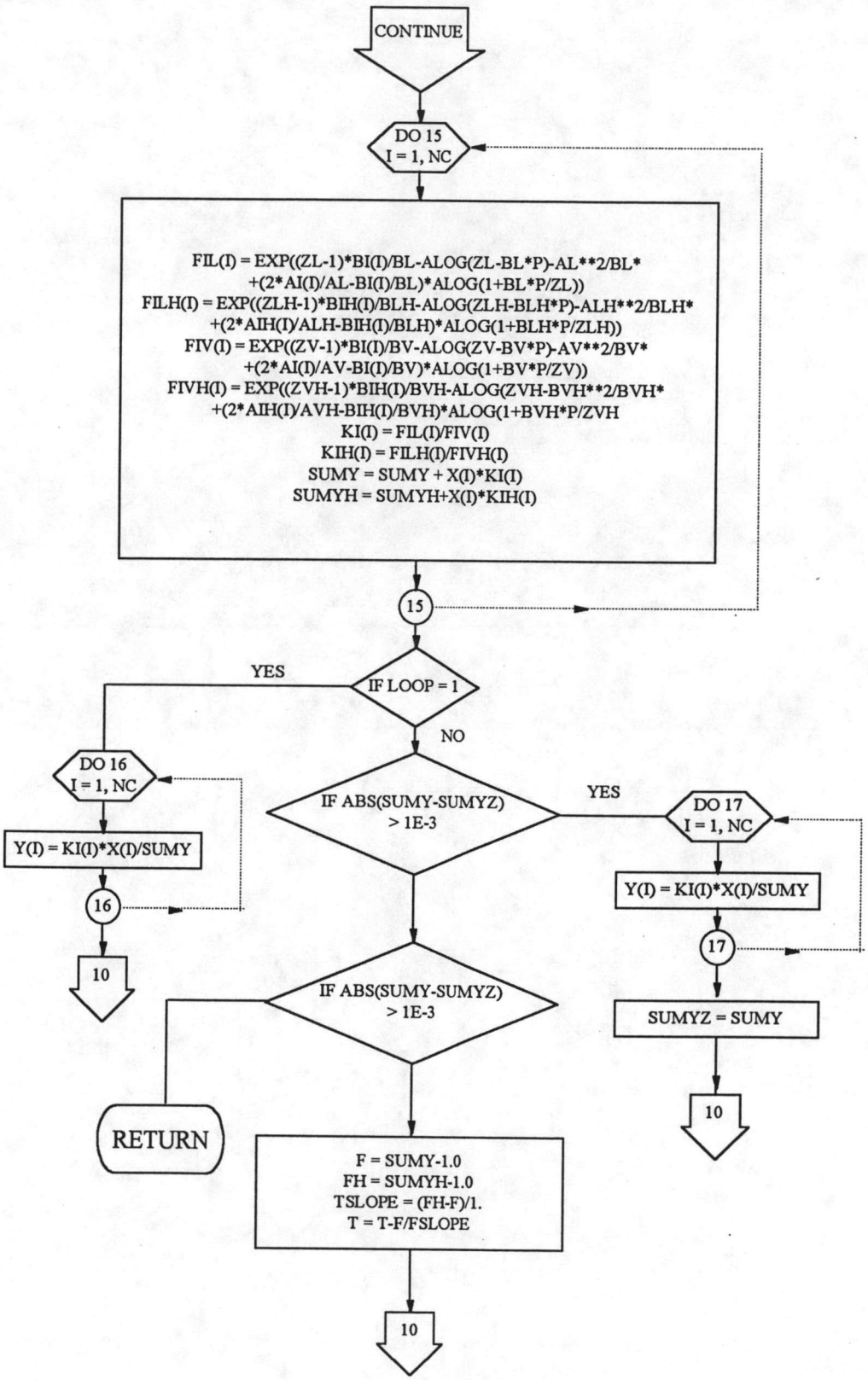
SUBROUTINE BUBPTGR



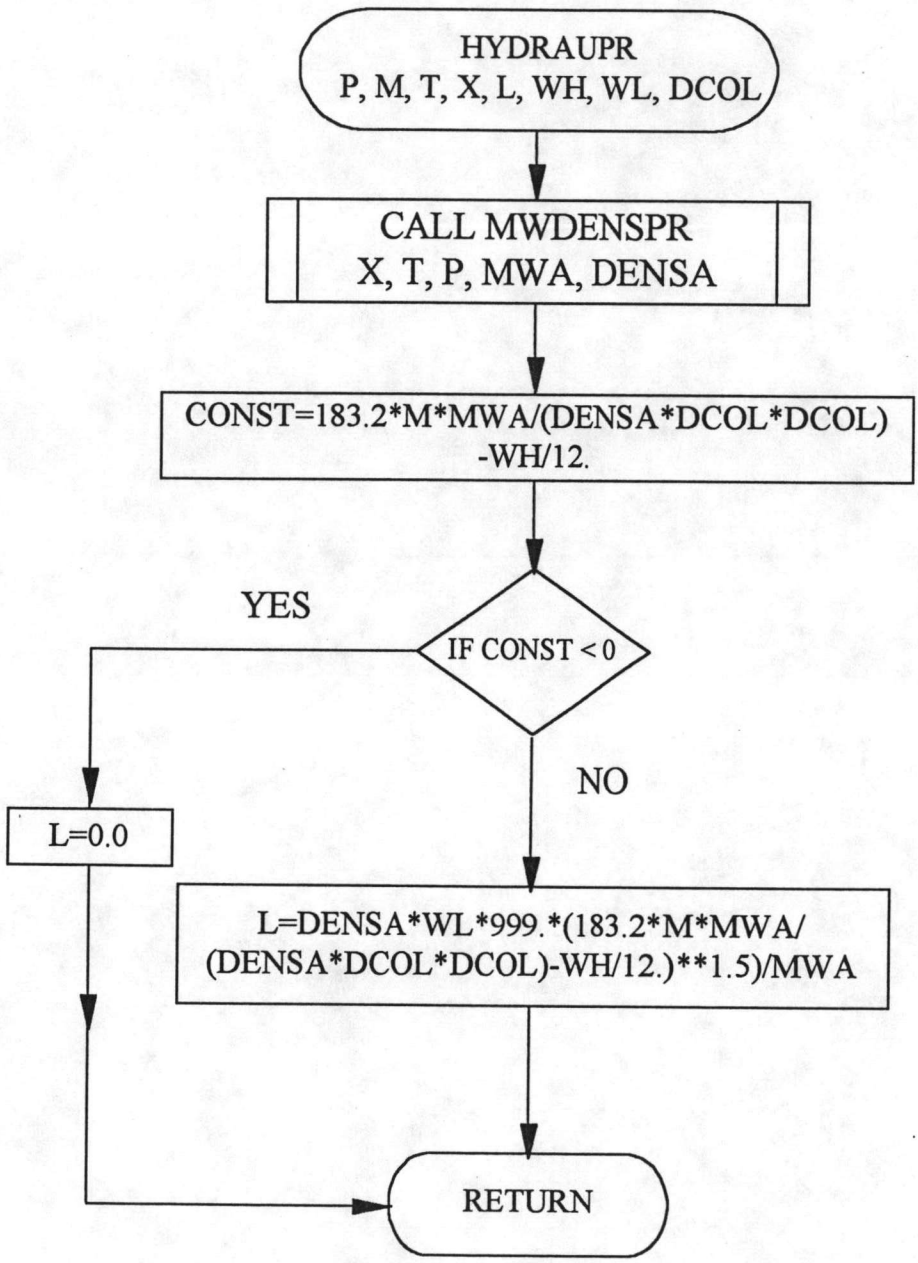
SUBROUTINE BUBPTGR (Continue)



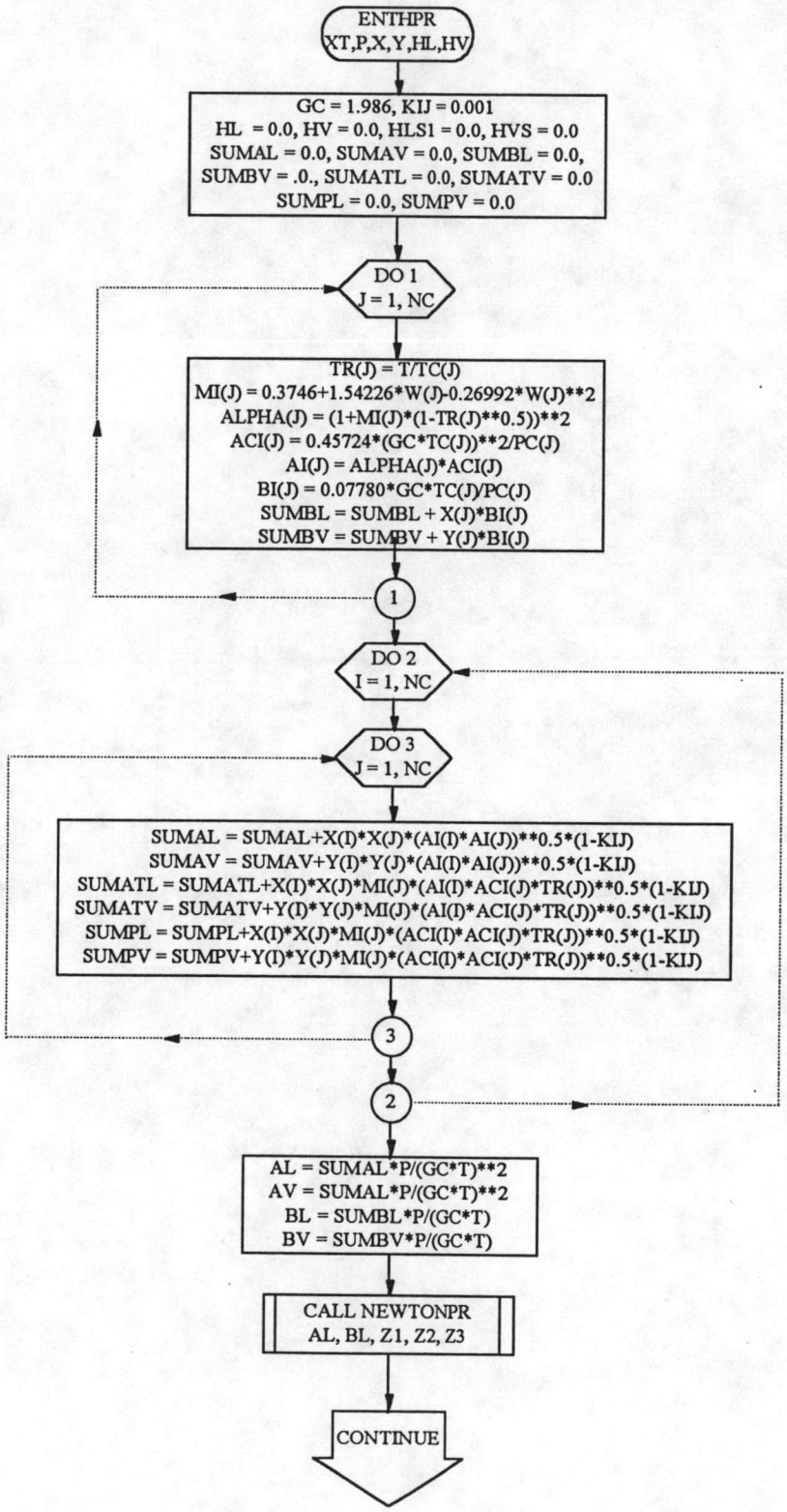
SUBROUTINE BUBPTGR (Continue)



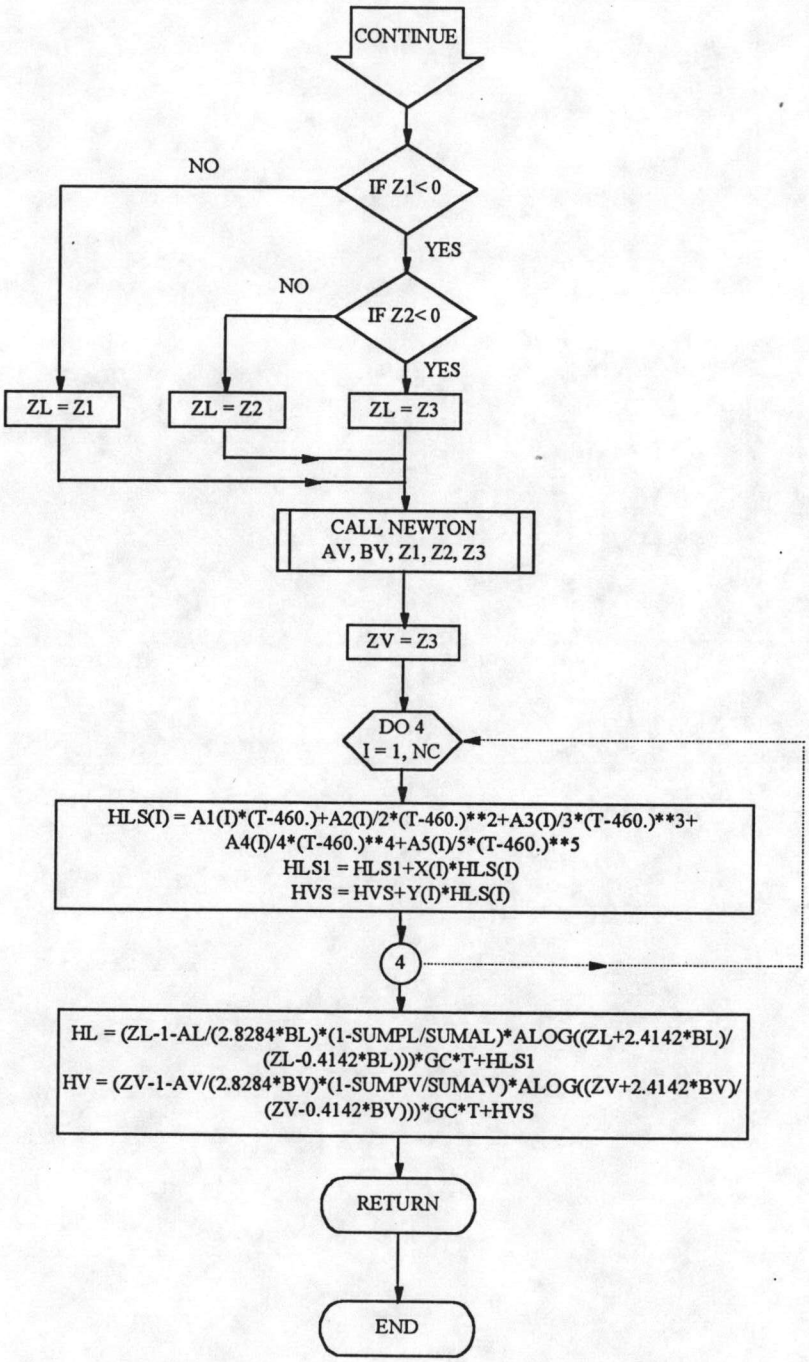
SUBROUTINE BUBPTGR (Continue)



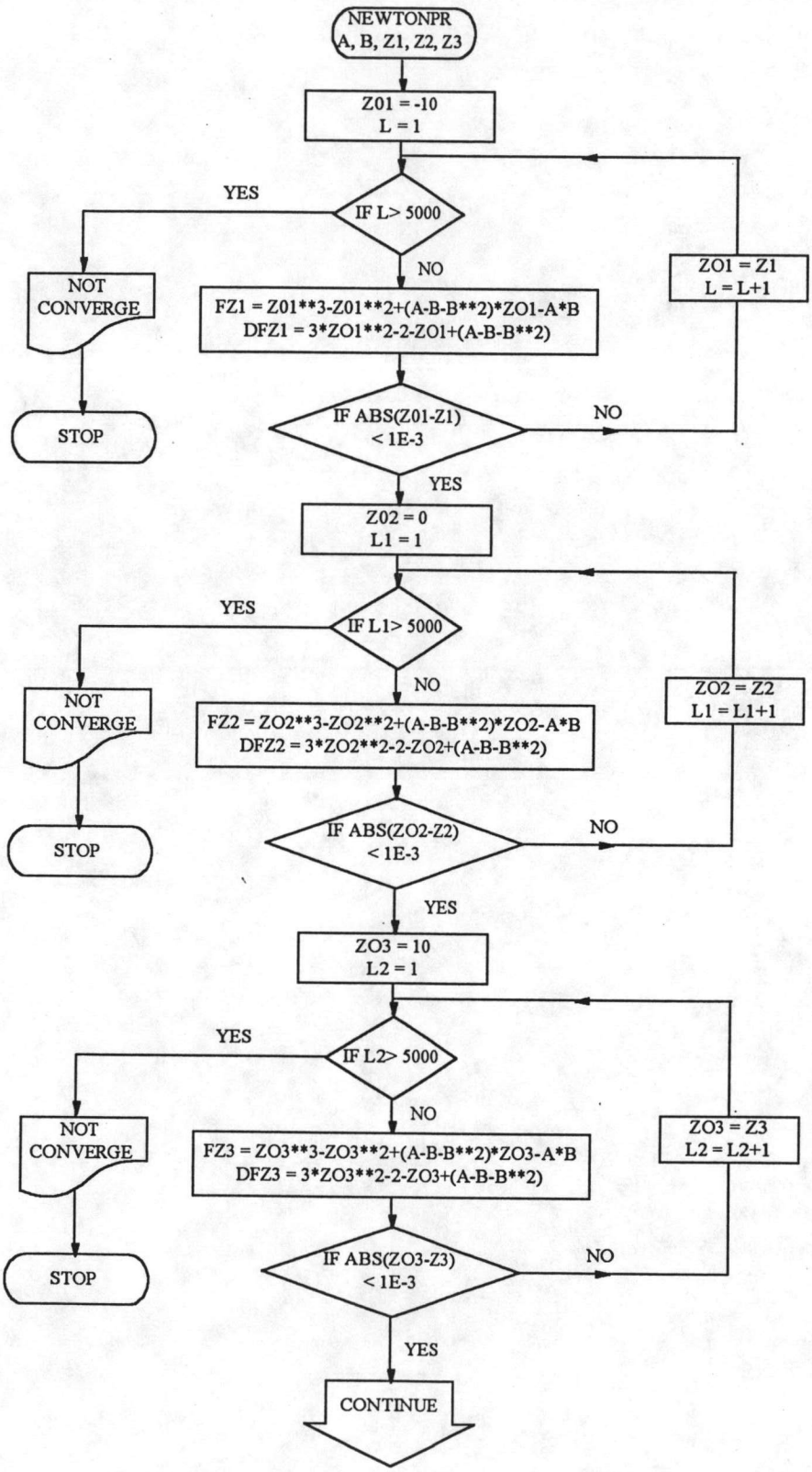
SUBPROGRAM HYDRAUPR



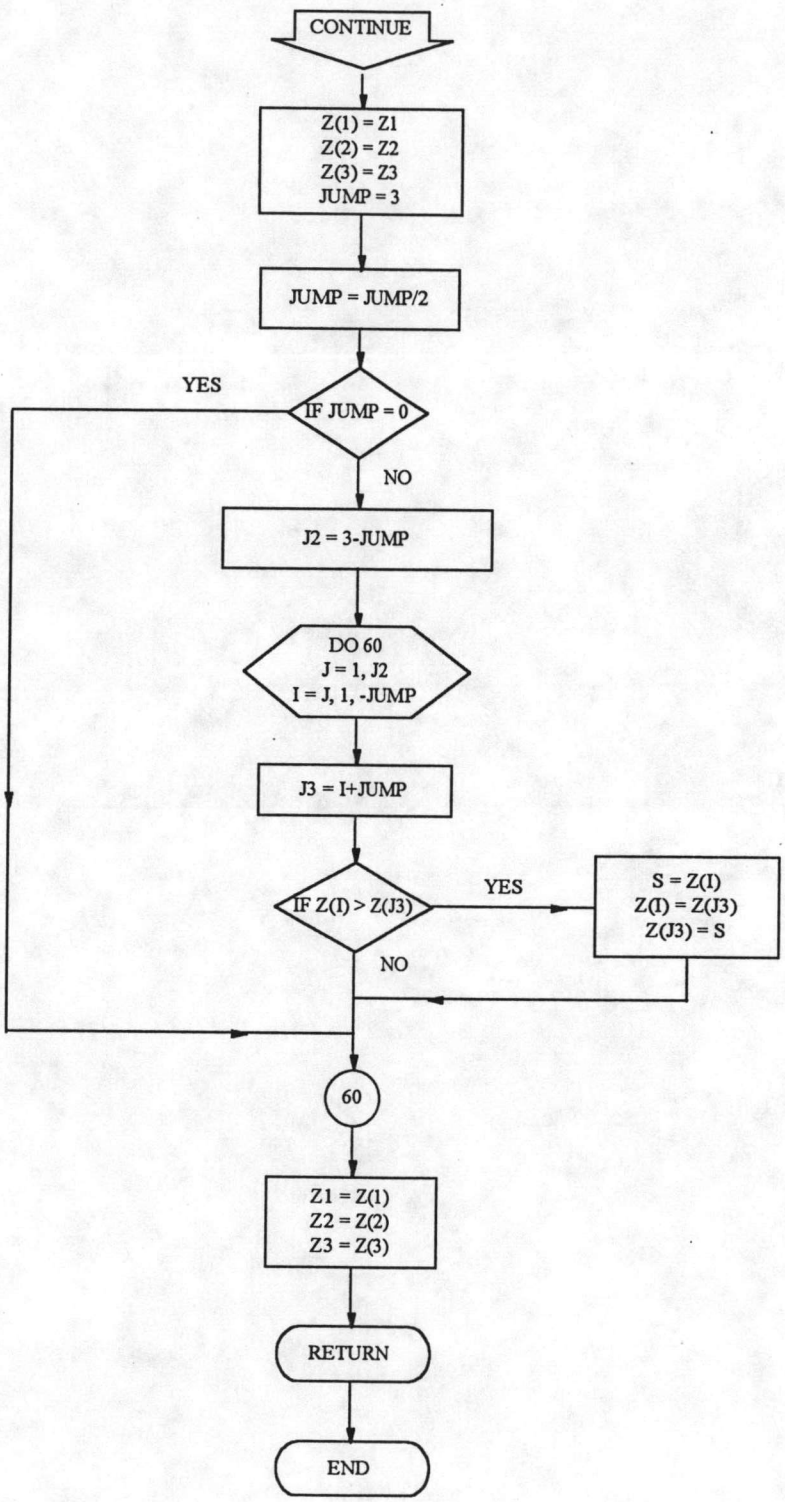
SUBROUTINE ENTHPR



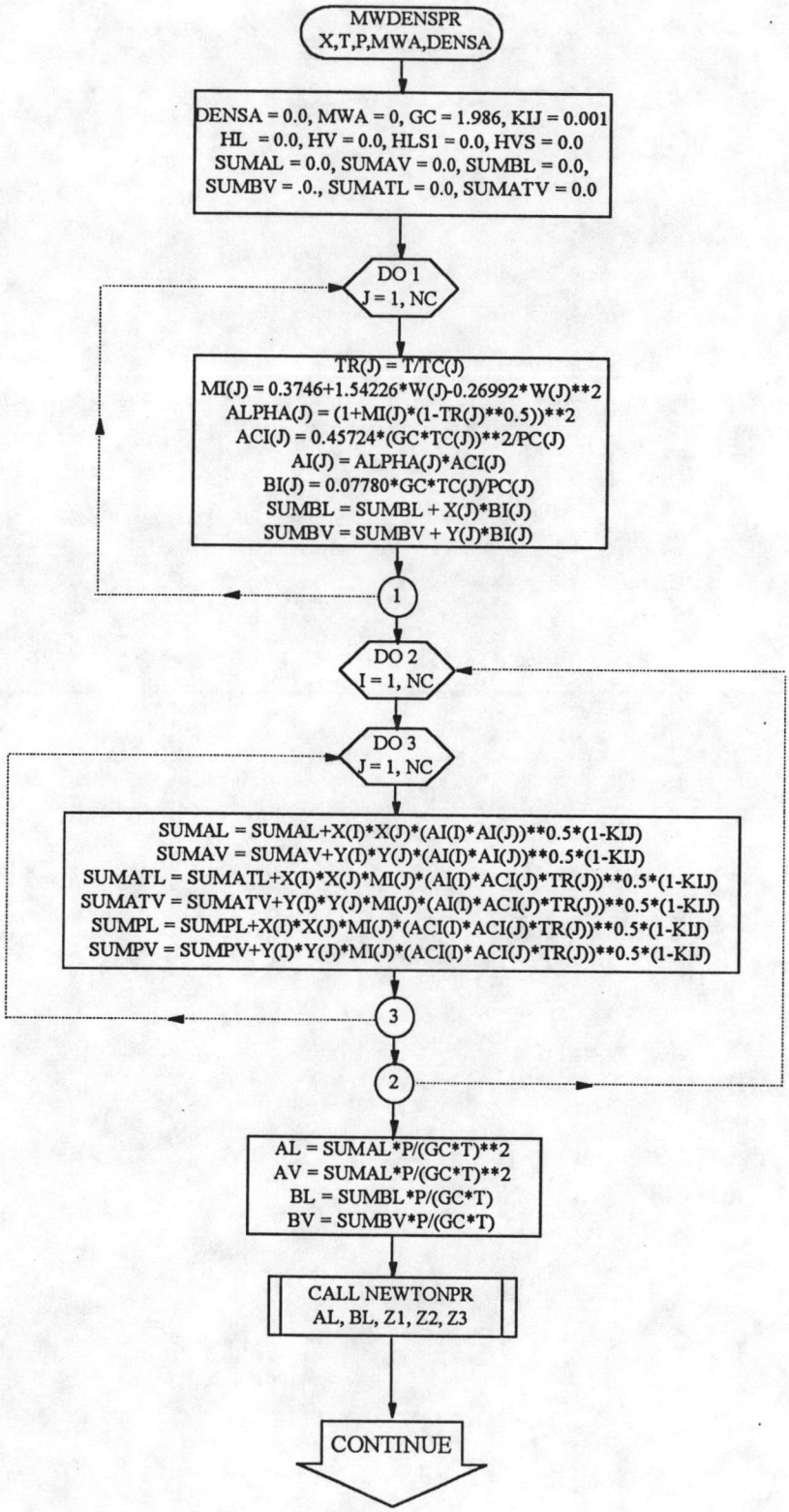
SUBROUTINE ENTHPR (Continue)



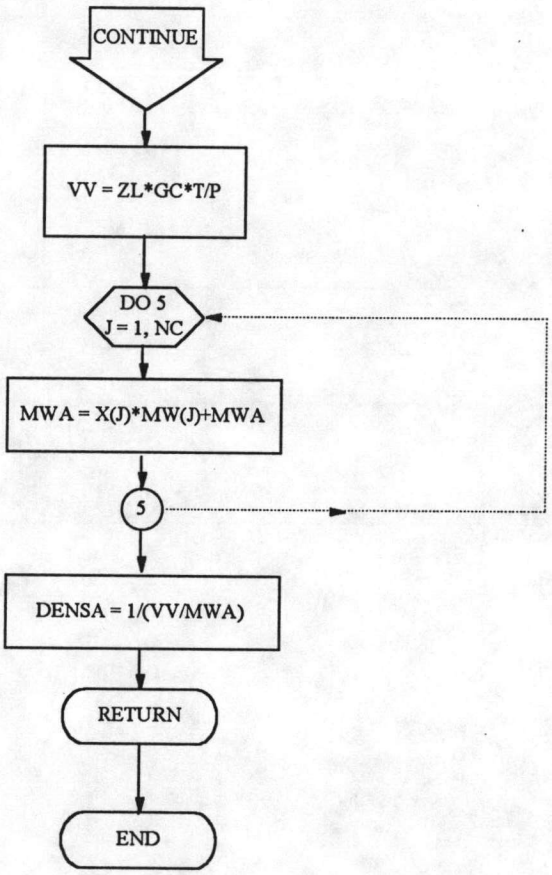
SUBROUTINE NEWTONPR



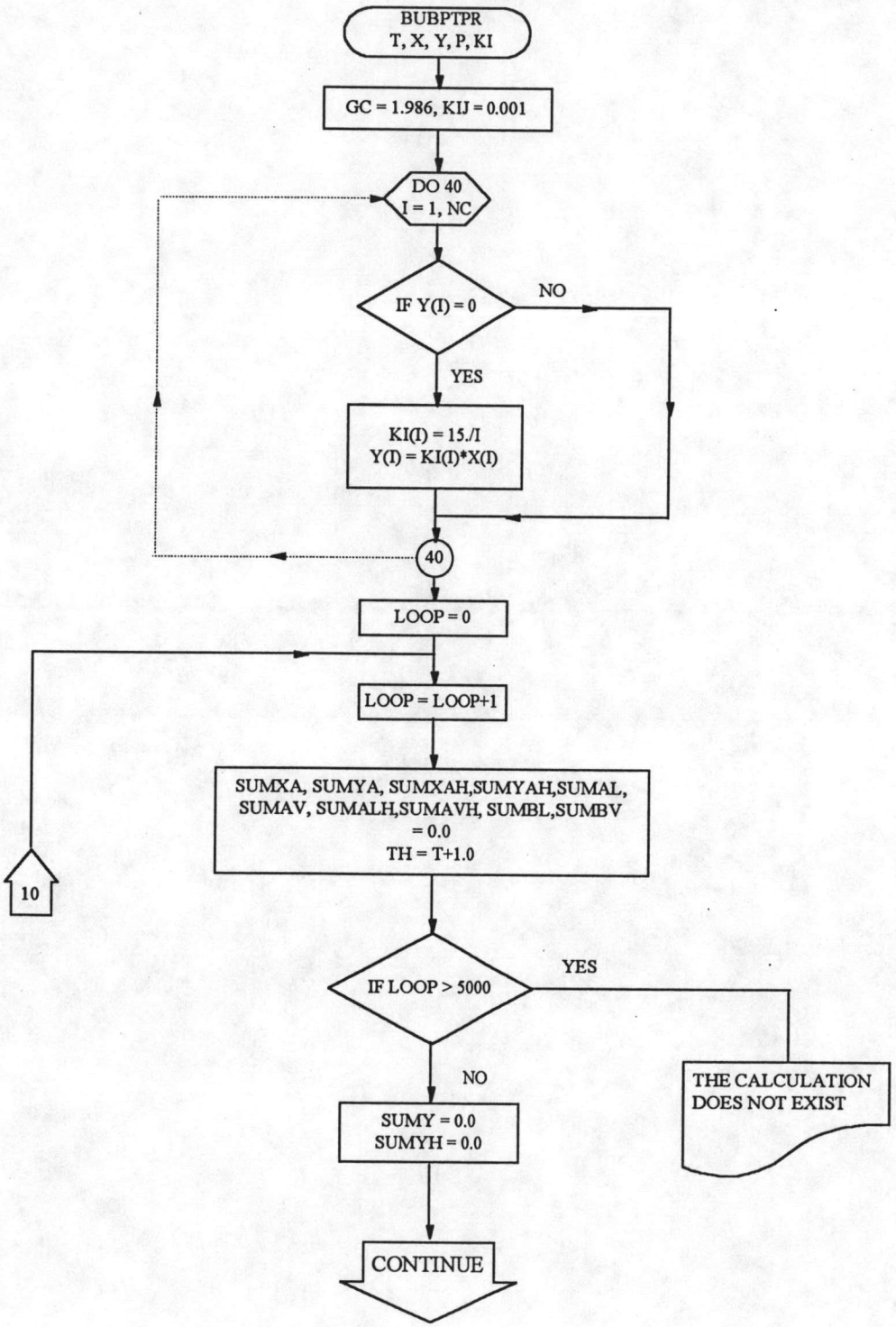
SUBROUTINE NEWTONPR (Continue)



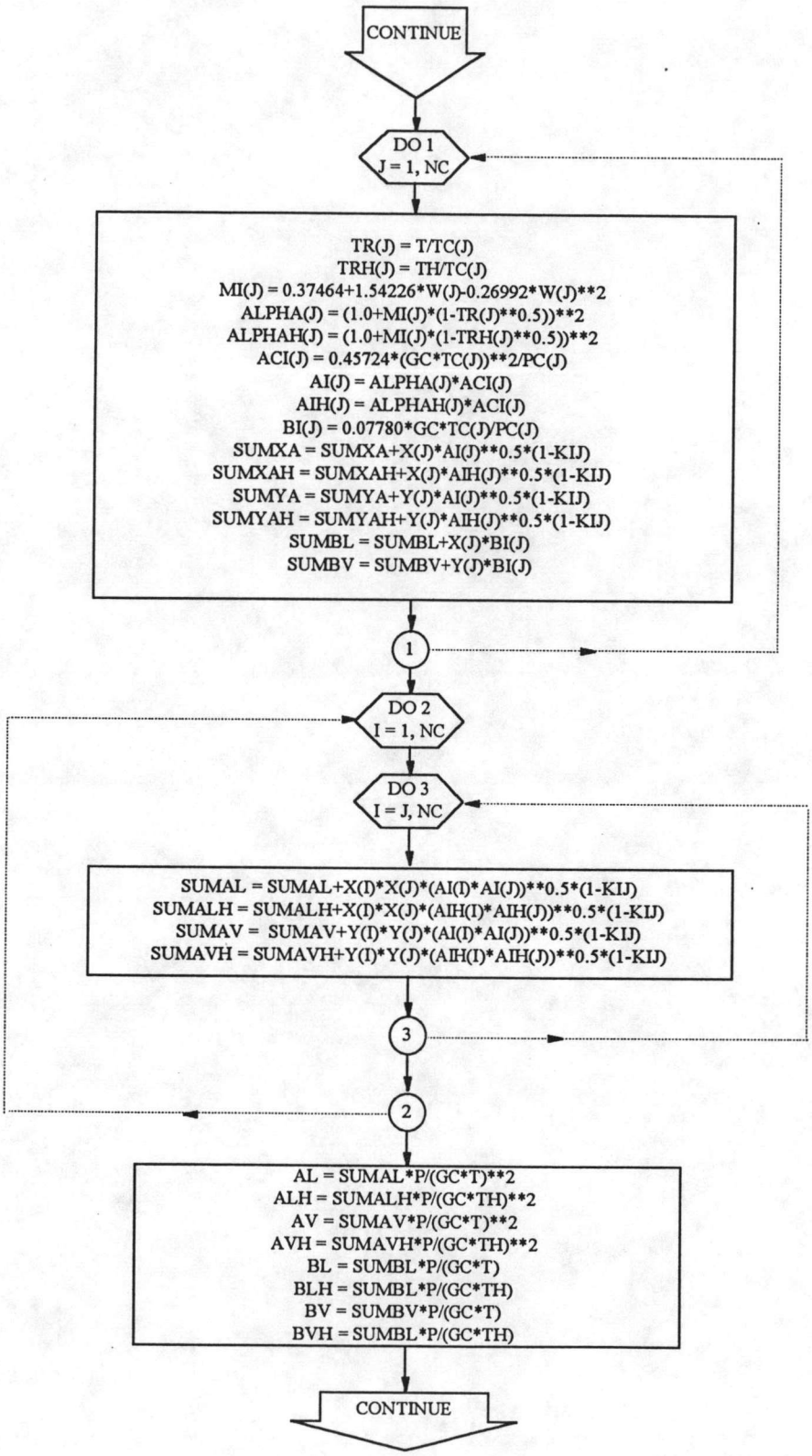
SUBROUTINE MWDENSPR



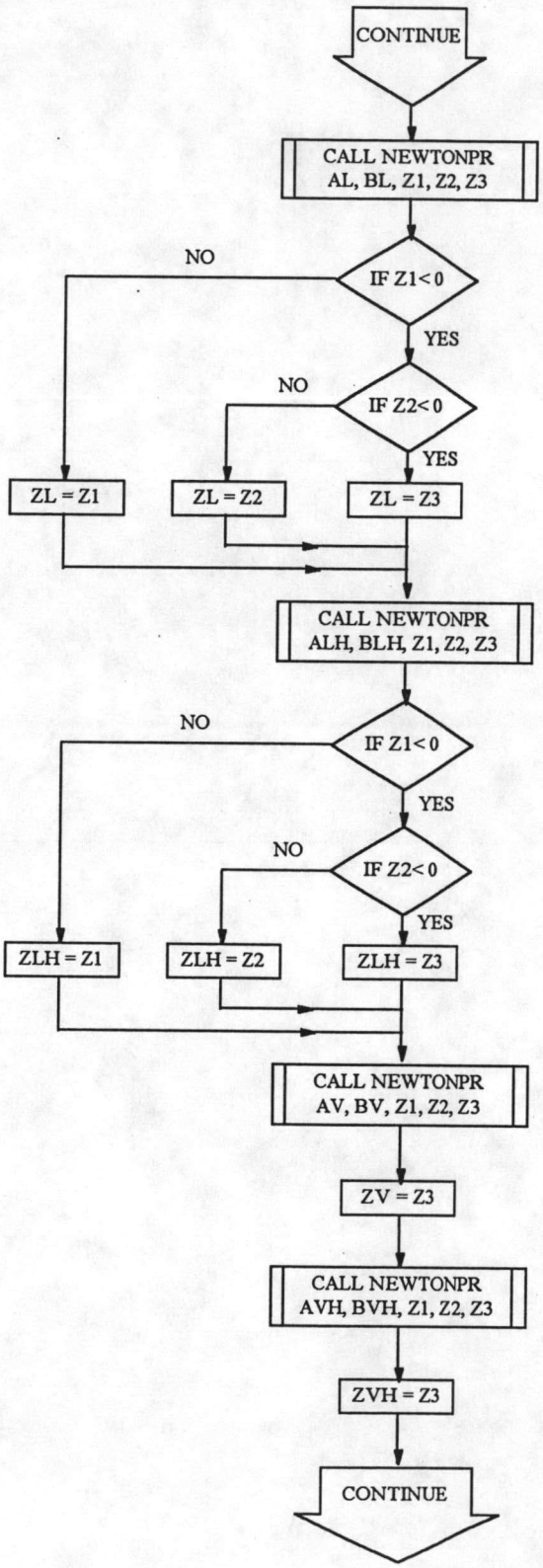
SUBROUTINE MWDENSPR (Continue)



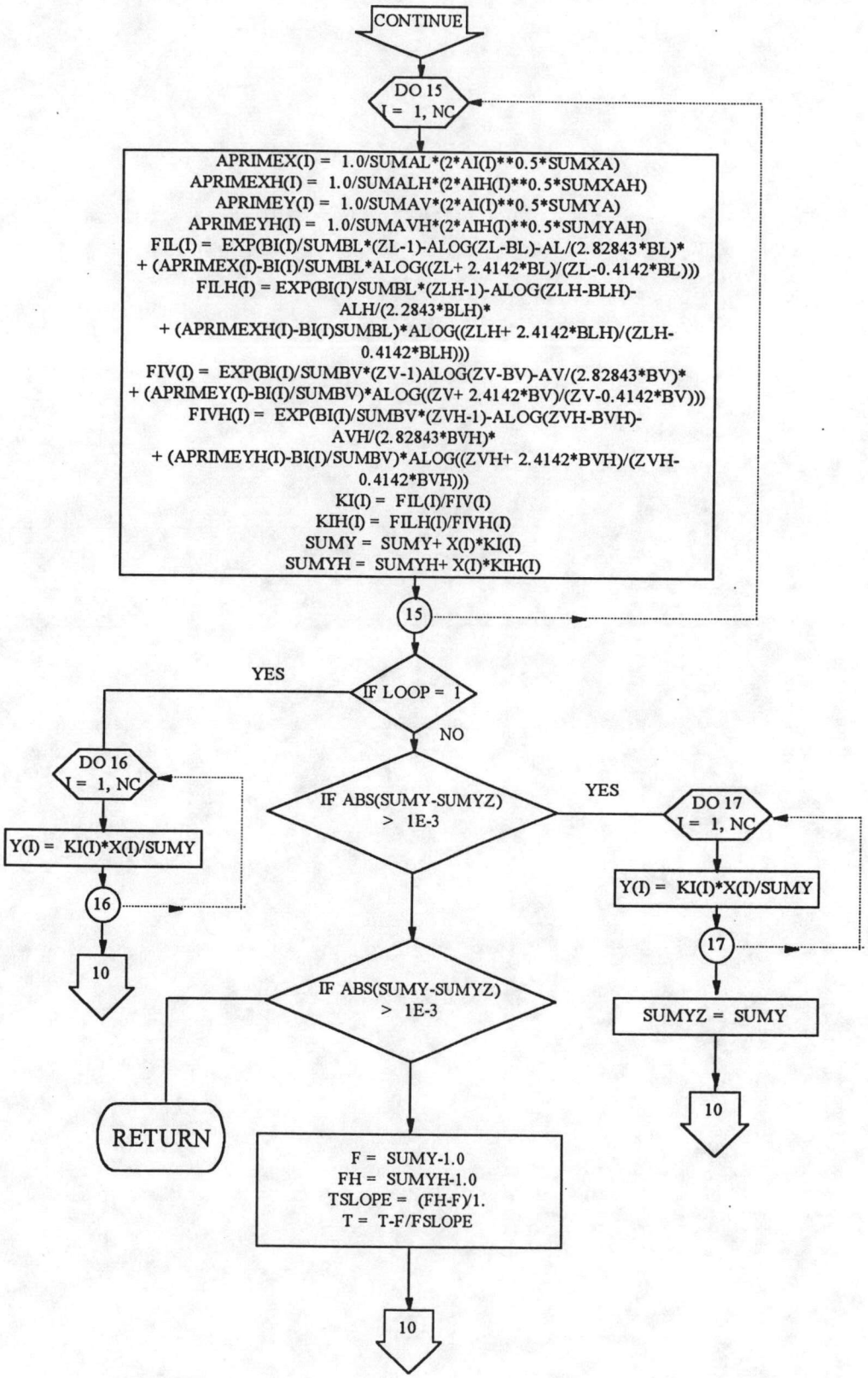
SUBROUTINE BUBPTPR



SUBROUTINE BUBPTPR (Continue)



SUBROUTINE BUBPTPR (Continue)



SUBROUTINE BUBPTPR (Continue)

Appendix B

List of computer program

```
CC    Dynamics Simulation Program for Multicomponent Distillation
CC    Created by Miss Kallaya        Klaithong  C 517152
CC    Chemical Engineering Department
CC    Chulalongkorn University @1994

      INCLUDE  'FGRAPH.FI'
      INCLUDE  'FGRAPH.FD'

CC
CC    Declare all variables
CC

CC
CC    INPUTS R, QR AND DV ARE FIXED
CC

      REAL MW,LO,MVB,MVD,MWA,MV,LV,M,L,MB,MD,MW1,TB,TC,PC,ZC,W,
     +A1,A2,A3,A4,A5

      CHARACTER*15 NAME,NAMEX,name1
      CHARACTER   str*2

      COMMON NC,MW(20),DENS(20),C1(20),C2(20),C3(20),BPT(20),AVP(20),
     +BVP(20),tc(20),pc(20),zc(20),w(20),a1(20),a2(20),a3(20),a4(20),
     +a5(20)

      DIMENSION LV(50),L(50),P(50),XF(20),YF(20),DXD(20),YAV(20),
     +YY(20),HL(50),HV(50),V(50),DM(50),DXM(50,20),XM(50,20),DXB(20)

      DIMENSION NAME(20),T(50),XB(20),X(50,20),Y(50,20),LO(50),XD(20),
     +YB(20),YD(20),XX(20),MV(50),M(50)
```

```
      INTEGER*2  dummy2, key
      EXTERNAL printmenu
      RECORD / rccoord / curpos

      OPEN( 1, FILE = 'physical.dat' )
      OPEN( 2, FILE = 'phytemp.dat' )
      OPEN( 5, FILE = 'result.dat')

CC
CC
CC

      CALL drawlines()
      CALL drawlines2()
      dummy2 = settextcolor(15)
      CALL settextposition( 3, 28, curpos )
      CALL outtext( 'Chulalongkorn  University' )
      CALL settextposition( 4, 25, curpos )
      CALL outtext( 'Chemical Engineering Department' )

      dummy2 = settextcolor( 14 )
      CALL settextposition( 11, 27, curpos )
      CALL outtext( 'Dynamics Simulation Program' )
      CALL settextposition(13, 38, curpos )
      CALL outtext( 'for' )
      CALL settextposition(15, 27, curpos )
      CALL outtext( 'Multicomponent Distillation' )

      dummy2 = settextcolor( 15 )
      CALL settextposition(21, 35, curpos )
      CALL outtext( 'Created by' )
      CALL settextposition( 22, 24, curpos )
      CALL outtext( 'Miss Kallaya  Klaithong   C 517152' )
      CALL settextposition( 23, 32, curpos )
      CALL outtext( 'M.Eng. Thesis @ 1994' )
```

```
      READ (*,*)

      CALL clearscreen( $VIEWPORT )

CC
CC    MENU
CC

 9000 key = -1
      DO WHILE( (key .LT. 0) .OR. (key .GT. 9) )
          CALL drawlines()
          CALL drawlines1()

          dummy2 = settextcolor( 14 )
          CALL settextposition( 4, 30, curpos )
          CALL outtext( '**** MAIN MENU ****' )
          dummy2 = settextcolor( 15 )
          CALL settextposition( 8, 25, curpos )
          CALL outtext( '0) Exit to Dos' )
          CALL settextposition( 9, 25, curpos )
          CALL outtext( '1) Components Database' )
          CALL settextposition( 10, 25, curpos )
          CALL outtext( '2) Characteristic of Column' )
          CALL settextposition( 11, 25, curpos )
          CALL outtext( '3) Operating Conditions' )
          CALL settextposition( 12, 25, curpos )
          CALL outtext( '4) Feed Data' )
          CALL settextposition( 13, 25, curpos )
          CALL outtext( '5) Initial Conditions')
          CALL settextposition( 14, 25, curpos )
          CALL outtext( '6) Edit Input Data' )
          CALL settextposition( 15, 25, curpos )
          CALL outtext( '7) Save Input Data' )
          CALL settextposition( 16, 25, curpos )
```

```
      CALL outtext( '8) Run' )
      CALL settextposition( 17, 25, curpos )
      CALL outtext( '9) Reporting & Reviewing the results' )
      dummy2 = settextcolor(14)
      CALL settextposition( 22, 25, curpos )
      CALL outtext ( 'Please ENTER your selection : ' )
      READ (*,*,ERR = 9000) key
   END DO
   dummy2 = setvideomode( $DEFAULTMODE )

   DO WHILE ( key .EQ. 0 )
        GOTO 400
   END DO

   DO WHILE ( key .EQ. 1 )
      GOTO 9100
   END DO

   DO WHILE ( key .EQ. 2 )
      GOTO 9200
   END DO

   DO WHILE ( key .EQ. 3 )
      GOTO 9300
   END DO

   DO WHILE ( key .EQ. 4 )
      GOTO 9400
   END DO

   DO WHILE ( key .EQ. 5 )
      GOTO 9500
   END DO

   DO WHILE ( key .EQ. 6 )
```

```
        GOTO 9600
      END DO

      DO WHILE ( key .EQ. 7 )
        GOTO 9700
      END DO

      DO WHILE ( key .EQ. 8 )
        GOTO 9800
      END DO

CC      DO WHILE ( key .EQ. 9 )
CC      GOTO 9900
CC      END DO

CC
CC      key = 1
CC

CC
CC      Component Database
CC

9100 keyc = -1
    DO WHILE( (keyc .LT. 0) .OR. (keyc .GT. 3) )
      CALL drawlines()
      CALL drawlines1()

      dummy2 = settextcolor( 14 )
      CALL settextposition( 4, 30, curpos )
      CALL outtext( '**** COMPONENT MENU ****' )
      dummy2 = settextcolor( 15 )
      CALL settextposition( 10, 25, curpos )
      CALL outtext( '0) Return to Main Menu' )
      CALL settextposition( 12, 25, curpos )
```

```
        CALL outtext( '1) List for All Components Database' )
        CALL settextposition( 14, 25, curpos )
        CALL outtext( '2) Select Component in your system ' )
        CALL settextposition( 16, 25, curpos )
        CALL outtext( '3) Append New Component Record' )
        dummy2 = settextcolor(14)
        CALL settextposition( 22, 25, curpos )
        CALL outtext ( 'Please ENTER your selection : ' )
        READ (*,*,ERR = 9100) keyc
      END DO
      dummy2 = setvideomode( $DEFAULTMODE )

      DO WHILE ( keyc .EQ. 0 )
          GOTO 9000
      END DO

      DO WHILE ( keyc .EQ. 1 )
          GOTO 9101
      END DO

      DO WHILE ( keyc .EQ. 2 )
          GOTO 9102
      END DO

      DO WHILE ( keyc .EQ. 3 )
          GOTO 9103
      END DO

CC
CC        keyc = 1
CC

CC
CC        1) List for All Component
CC
```

```
9101 CALL drawlines()
     CALL drawlines3()
     REWIND( UNIT = 1 )
9111 kk = 0
     dummy2 = settextcolor( 14 )
     CALL settextposition( 3, 27, curpos )
     CALL outtext( '*** Component Library ***' )
     dummy2 = settextcolor( 15 )
     kk1 = 0
9105 READ( 1, 9110, END = 9107 ) NAME1
     CALL settextposition( 6+kk, 10+kk1, curpos )
     CALL outtext( name1 )
     kk1 = kk1+16
     IF ( kk .GE. 16 ) THEN
          dummy2 = settextcolor( 14 )
          CALL settextposition( 23, 4, curpos )
          CALL outtext( ' Press any key to continue ...' )
          READ(*,*)
          CALL drawlines()
          GOTO 9111
     END IF
     IF ( kk1 .GE. 50 ) THEN
          kk  = kk + 1
          kk1 = 0
     END IF
     GOTO 9105
9110 FORMAT(3X,A15)
9120 FORMAT(A15)

9107 dummy2 = settextcolor( 14 )
     CALL settextposition( 23, 4, curpos )
     CALL outtext( ' Press any key to continue ...' )
     READ(*,*)
     GOTO 9100
```

```
CC
CC      keyc = 2
CC

CC
CC      2) Select Component
CC

 9102 CALL drawlines()
      CALL drawlines1()
      dummy2 = settextcolor( 14 )
      CALL settextposition( 4, 27, curpos )
      CALL outtext( '*** Component Selection ***' )
      dummy2 = settextcolor( 15 )
      CALL settextposition( 5, 20, curpos )
      CALL outtext( 'The name must be typed in CAPITAL LETTER' )
      dummy2 = settextcolor( 14 )
      CALL settextposition( 9, 18, curpos )
      CALL outtext( '  Total number of component ( NC ) = ' )
      READ(*,*) NC
 2220 DO 9130 j = 1, NC
      CALL settextposition( 10+j, 22, curpos )
      WRITE( str, '(I2)' ) j
      CALL outtext( 'Component Name no.'// str // ' =  ' )
      READ(*,9120) name(j)
      REWIND(UNIT = 1)
 2200 READ( 1, 2070, END = 2210 ) NAME1
      IF ( name1 .EQ. name(j) ) THEN
           BACKSPACE( UNIT = 1)
           READ( 1, 2080 ) mw1,tb1,tc1,pc1,zc1,w1,aa1,aa2,aa3,aa4,aa5
           mw(j)   = mw1
           bpt(j)  = tb1
           tc(j)   = tc1
           pc(j)   = pc1
```

```
            zc(j)    = zc1
            w(j)     = w1
            a1(j)    = aa1
            a2(j)    = aa2
            a3(j)    = aa3
            a4(j)    = aa4
            a5(j)    = aa5
            GOTO 9130
        END IF
        GOTO 2200
2210 CALL settextposition( 24, 5, curpos )
        CALL outtext( 'Component not exist !! ... Press ENTER' )
        READ(*,*)
        CALL drawlines()
        CALL drawlines1()
        dummy2 = settextcolor( 14 )
        CALL settextposition( 4, 27, curpos )
        CALL outtext( '*** Component Selection ***' )
        dummy2 = settextcolor( 15 )
        CALL settextposition( 5, 20, curpos )
        CALL outtext( 'The name must be typed in CAPITAL LETTER' )
        dummy2 = settextcolor( 14 )
        CALL settextposition( 9, 18, curpos )
        WRITE( str, '(I2)' ) NC
        CALL outtext( '  Total number of component ( NC ) = '// str )
        GOTO 2220
9130 CONTINUE
        CALL settextposition( 24, 5, curpos )
        CALL outtext( 'Press ENTER to continue ...' )
        READ(*,*)
2095 kk2 = 0
        CALL drawlines()
        CALL drawlines3()
        dummy2 = settextcolor( 14 )
        CALL settextposition( 3, 10, curpos )
```

```
      WRITE( str,'(I2)' ) NC
      CALL outtext( 'Your system has '//str//' components : ' )
      dummy2 = settextcolor( 15 )
      kk3 = 0
      DO 2090 j = 1,NC
      CALL settextposition( 6+kk2, 10+kk3, curpos )
      namex = name(j)
      CALL outtext( namex )
      kk3 = kk3+16
      IF ( kk2 .GE. 16 ) THEN
            dummy2 = settextcolor( 14 )
            CALL settextposition( 23, 4, curpos )
            CALL outtext( ' Press any key to continue ...' )
            READ(*,*)
            CALL drawlines()
            GOTO 2095
      END IF
      IF ( kk3 .GE. 50 ) THEN
            kk2  = kk2 + 1
            kk3 = 0
      END IF
 2090 CONTINUE
      dummy2 = settextcolor( 14 )
      CALL settextposition( 23, 4, curpos )
      CALL outtext( ' Press any key to continue ...' )
      READ(*,*)
      GOTO 9100
 2070 FORMAT(3X,A15)
 2080 FORMAT(18X,F8.3,3F7.1,F6.3,F8.4,F10.6,4E16.7)

CC
CC       keyc = 3
CC


CC
```

```
CC      3) Append New Component Record
CC


 9103 REWIND(UNIT = 1)
 3001 READ( 1, 3070, END = 3000 ) ID,NAME1,MW1,TB1,TC1,PC1,ZC1,W1,AA1,
     +AA2,AA3,AA4,AA5
      WRITE(2,3070) ID,NAME1,MW1,TB1,TC1,PC1,ZC1,W1,AA1,AA2,AA3,AA4,AA5
      GOTO 3001
 3000 CALL drawlines()
      dummy2 = settextcolor( 14 )
      CALL settextposition( 4, 22, curpos )
      CALL outtext( '**** APPEND NEW COMPONENT RECORD ****' )
      dummy2 = settextcolor( 15 )
      CALL settextposition( 6, 4, curpos )
      CALL outtext( 'Physical Property' )
      dummy2 = settextcolor( 14 )
      CALL settextposition( 10, 4, curpos )
      CALL outtext( 'ID no. ( 0 to terminate ) = ' )
      READ(*,3080) ID
      IF ( id .EQ. 0 ) THEN
          GOTO 3040
      END IF
      CALL settextposition( 12, 6, curpos )
      CALL outtext( 'Component Name       = ' )
      READ(*,3087) NAME1
      CALL settextposition( 13, 6, curpos )
      CALL outtext( 'Molecular Weight     = ' )
      READ(*,*) MW1
      CALL settextposition( 14, 6, curpos )
      CALL outtext( 'Boiling Point ( R )  = ' )
      READ(*,*) TB1
      CALL settextposition( 15, 6, curpos )
      CALL outtext( 'Critical Temp.( R )  = ' )
      READ(*,*) TC1
      CALL settextposition( 16, 6, curpos )
```


```
      CALL outtext( 'Critical Press.(psia) = ' )
      READ(*,*) PC1
      CALL settextposition( 17, 6, curpos )
      CALL outtext( 'Critical Z-factor     = ' )
      READ(*,*) ZC1
      CALL settextposition( 18, 6, curpos )
      CALL outtext( 'Acentric Factor       = ' )
      READ(*,*) W1

      dummy2 = settextcolor( 15 )
      CALL settextposition( 6, 45, curpos )
      CALL outtext( 'Heat Capacity Coefficients' )
      dummy2 = settextcolor( 14 )
      CALL settextposition( 12, 47, curpos )
      CALL outtext( ' a1   = ' )
      READ(*,*) AA1
      CALL settextposition( 13, 47, curpos )
      CALL outtext( ' a2   = ' )
      READ(*,*) AA2
      CALL settextposition( 14, 47, curpos )
      CALL outtext( ' a3   = ' )
      READ(*,*) AA3
      CALL settextposition( 15, 47, curpos )
      CALL outtext( ' a4   = ' )
      READ(*,*) AA4
      CALL settextposition( 16, 47, curpos )
      CALL outtext( ' a5   = ' )
      READ(*,*) AA5
      WRITE(2,3070) ID,NAME1,MW1,TB1,TC1,PC1,ZC1,W1,AA1,AA2,AA3,AA4,AA5
      GOTO 3000
3040 END FILE(UNIT = 2)
3070 FORMAT(I3,A15,F8.3,3F7.1,F6.3,F8.4,F10.6,4E16.7)
3080 FORMAT(I3)
3087 FORMAT(A15)
3090 FORMAT(F4.1)
```

```
3095 FORMAT(F5.3)
3097 FORMAT(F5.4)
     GOTO 9100

CC
CC       key = 2
CC

CC
CC     READ COLUMN DATA
CC

9200 CALL drawlines()
     CALL drawlines2()
     dummy2 = settextcolor( 15 )
     CALL settextposition( 4, 23, curpos )
     CALL outtext( '**** Characteristic of Column ****' )
2000 dummy2 = settextcolor( 14 )
     CALL settextposition(  8, 14, curpos )
     CALL outtext( 'Total number of plates  (NT)  =  ' )
     READ(*,*) NT
     CALL settextposition( 10, 14, curpos )
     CALL outtext( 'Number of trays in stripping section  (NF) =  ' )
     READ(*,*) NF
     CALL settextposition( 11, 14, curpos )
     CALL outtext( 'Weir height in stripping section     (WHS) =  ' )
     READ(*,*) WHS
     CALL settextposition( 12, 14, curpos )
     CALL outtext( 'Weir height in rectifying section    (WHR) =  ' )
     READ(*,*) WHR
     CALL settextposition( 13, 14, curpos )
     CALL outtext( 'Column diameter in stripping section  (DS) =  ' )
     READ(*,*) DS
     CALL settextposition( 14, 14, curpos )
     CALL outtext( 'Column diameter in rectifying section (DR) =  ' )
```

```
      READ(*,*) DR
      CALL settextposition( 15, 14, curpos )
      CALL outtext( 'Weir length in stripping section     (WLS) = ' )
      READ(*,*) WLS
      CALL settextposition( 16, 14, curpos )
      CALL outtext( 'Weir length in rectifying section    (WLR) = ' )
      READ(*,*) WLR
      CALL settextposition( 17, 14, curpos )
      CALL outtext( 'Volumetric holdup in column base     (MVB) = ' )
      READ(*,*) MVB
      CALL settextposition( 18, 14, curpos )
      CALL outtext( 'Volumetric holdup in reflux drum     (MVD) = ' )
      READ(*,*) MVD
  301 FORMAT(A\)
   13 FORMAT( 3I3 )
   14 FORMAT( 8F6.2 )
      CALL settextposition( 24, 5, curpos )
      CALL outtext( 'Press ENTER to continue ...' )
      READ(*,*)
      GOTO 9000

CC
CC       key = 3
CC

CC
CC     READ CONDITIONS
CC

 9300 CALL drawlines()
      CALL drawlines2()
      dummy2 = settextcolor( 15 )
      CALL settextposition( 4, 25, curpos )
      CALL outtext( '**** Operating Conditions ****' )
      dummy2 = settextcolor( 14 )
```

```
      CALL settextposition(  8, 14, curpos )
      CALL outtext( 'Pressure in top of column    PD, (psia)  =  ' )
      READ(*,*) PD
      CALL settextposition( 10, 14, curpos )
      CALL outtext( 'Pressure in bottom of column PB, (psia)  =  ' )
      READ(*,*) PB
      CALL settextposition( 12, 14, curpos )
      CALL outtext( 'Reboiler heat-duty  QR, (x 1,000 Btu/hr) =  ')
      READ(*,*) QR
      CALL settextposition( 14, 14, curpos )
      CALL outtext( 'Reflux rate  R, (moles/hr) =  ' )
      READ(*,*) R
      CALL settextposition( 16, 14, curpos )
      CALL outtext( 'Vapor product from reflux drum DV, (moles/hr) =  ')
      READ(*,*) DV
      CALL settextposition( 18, 14, curpos )
      CALL outtext( 'Murphree vapor-phase tray efficiency     EFF  =  ')
      READ(*,*) EFF
      CALL settextposition( 24, 5, curpos )
      CALL outtext( 'Press ENTER to continue ...' )
      READ(*,*)
      GOTO 9000

CC
CC       key = 4
CC


CC
CC    READ FEED
CC


 9400 CALL drawlines()
      CALL drawlines2()
      dummy2 = settextcolor( 15 )
      CALL settextposition( 4, 29, curpos )
```

```
      CALL outtext( '**** Feed Data ****' )
      dummy2 = settextcolor( 14 )
      CALL settextposition( 8, 14, curpos )
      CALL outtext( 'Liquid Feed flow rate (moles/hr)  = ' )
      READ(*,*) FL
      CALL settextposition( 9, 14, curpos )
      CALL outtext( 'Liquid Feed temperature     (F)   = ' )
      READ(*,*) TFL
      CALL settextposition( 11, 14, curpos )
      CALL outtext( 'Composition (mole fraction) : ' )
      DO 310 J = 1,NC
      CALL settextposition( 11+J, 17, curpos )
      namex = name(j)
      CALL outtext( namex // ' =  ' )
      READ(*,*) XF(J)
  310 CONTINUE
      CALL drawlines()
      CALL drawlines2()
      dummy2 = settextcolor( 15 )
      CALL settextposition( 4, 29, curpos )
      CALL outtext( '**** Feed Data ****' )
      dummy2 = settextcolor( 14 )
      CALL settextposition( 8, 14, curpos )
      CALL outtext( 'Vapor Feed flow rate (moles/hr)  = ' )
      READ(*,*) FV
      CALL settextposition( 9, 14, curpos )
      CALL outtext( 'Vapor Feed temperature     (F)   = ' )
      READ(*,*) TFV
      CALL settextposition( 11, 14, curpos )
      CALL outtext( 'Composition (mole fraction) : ' )
      DO 320 J = 1,NC
      CALL settextposition( 11+J, 17, curpos )
      namex = name(j)
      CALL outtext( namex // ' =  '    )
      READ(*,*) YF(J)
```

```
 320 CONTINUE
CC       CALL ENTHS(TFL,P(NF),XF,YF,HLF,HVF)
   8 FORMAT(1X,2F8.2,10E10.2)
     CALL settextposition( 24, 5, curpos )
     CALL outtext( 'Press ENTER to continue ...' )
     READ(*,*)
     GOTO 9000

CC
CC       key = 5
CC

CC
CC    READ INITIAL CONDITIONS
CC

 9500 CALL drawlines()
     CALL drawlines2()
     dummy2 = settextcolor( 15 )
     CALL settextposition( 4, 25, curpos )
     CALL outtext( '**** Initial Conditions ****' )
     dummy2 = settextcolor( 14 )
     CALL settextposition( 8, 14, curpos )
     CALL outtext( 'Temperature in column base TB, (F)  =  ' )
     READ(*,*) TB
     CALL settextposition( 10, 14, curpos )
     CALL outtext( 'Composition in Column base  :' )
     DO 510 J = 1,NC
     CALL settextposition( 11+J, 17, curpos )
     namex = name(j)
     CALL outtext( namex // ' =  ' )
     READ(*,*) XB(J)
 510 CONTINUE
     DO 15 N = 1,NT
     CALL drawlines()
```

```
      CALL drawlines2()
      dummy2 = settextcolor( 15 )
      CALL settextposition( 4, 25, curpos )
      CALL outtext( '**** Initial Conditions ****' )
      dummy2 = settextcolor( 14 )
      WRITE( str, '(I2)' ) N
      CALL settextposition( 8, 14, curpos )
      CALL outtext( 'Temperature in tray no.'//str//' (F) = ' )
      READ(*,*) T(N)
      CALL settextposition( 9, 14, curpos )
      CALL outtext( 'Liquid flow rate in tray no.'//str//' (moles/hr) =
     + ' )
      READ(*,*) LO(N)
      DO 520 J = 1,NC
      CALL settextposition( 10, 14, curpos )
      CALL outtext( 'Composition in tray no.'//str//' :' )
      CALL settextposition( 11+J, 17, curpos )
      namex = name(j)
      CALL outtext( namex // ' = ' )
      READ(*,*) X(N,J)
  520 CONTINUE
      BLANK=0.
      CALL settextposition( 24, 5, curpos )
      CALL outtext( 'Press ENTER to continue ...' )
      READ(*,*)
   15 CONTINUE
      CALL drawlines()
      CALL drawlines2()
      dummy2 = settextcolor( 15 )
      CALL settextposition( 4, 25, curpos )
      CALL outtext( '**** Initial Conditions ****' )
      dummy2 = settextcolor( 14 )
      CALL settextposition( 8, 14, curpos )
      CALL outtext( 'Temperature in reflux drum TD, (F) = ' )
      READ(*,*) TD
```

```
      CALL settextposition( 10, 14, curpos )
      CALL outtext( 'Composition in reflux drum  :' )
      DO 530 J = 1,NC
      CALL settextposition( 11+J, 17, curpos )
      namex = name(j)
      CALL outtext( namex // ' =  ' )
      READ(*,*) XD(J)
  530 CONTINUE
      CALL settextposition( 24, 5, curpos )
      CALL outtext( 'Press ENTER to continue ...' )
      READ(*,*)
      GOTO 9000

CC
CC       Read old data from floppy disk
CC

CC
CC       key = 6
CC

 9600 CALL old()
      REWIND(UNIT = 3)
      READ(3,9610) NC
      DO 9615 j = 1,NC
      READ(3,9625) name(j),mw(j),bpt(j),tc(j),pc(j),zc(j),w(j),a1(j),
     +a2(j),a3(j),a4(j),a5(j)
 9615 CONTINUE
      READ(3,9630) NT,NF,WHS,WHR,DS,DR,WLS,WLR,MVB,MVD
      READ(3,9640) PD,PB,QR,R,DV,EFF
      READ(3,9645) FL,TFL,(XF(J), J = 1,NC)
      READ(3,9645) FV,TFV,(YF(J), J = 1,NC)
      BLANK = 0.
      READ(3,9650) TB,BLANK,(XB(J), J = 1,NC)
      DO 9660 N = 1,NT
```

```
      NN = N
      READ(3,9655) NN,T(N),LO(N),(X(N,J),J=1,NC)
 9660 CONTINUE
      READ(3,9650) TB,R,(XD(J), J = 1,NC)
 9610 FORMAT(I3)
 9625 FORMAT(A15,F8.3,3F7.1,F6.3,F8.4,F10.6,4E16.7)
 9630 FORMAT(2I3,8F6.2)
 9640 FORMAT(6F8.2)
 9645 FORMAT(2F12.2,20E16.7)
 9650 FORMAT(5X,2F8.2,20E16.7)
 9655 FORMAT(1X,I3,1X,2F8.2,20E16.7)
      GOTO 9000

CC
CC       Save data into floppy disk
CC

CC
CC       key = 7
CC

 9700 CALL save()
      WRITE(4,9610) NC
      DO 9715 j = 1,NC
      WRITE(4,9625) name(j),mw(j),bpt(j),tc(j),pc(j),zc(j),w(j),a1(j),
     +a2(j),a3(j),a4(j),a5(j)
 9715 CONTINUE
      WRITE(4,9630) NT,NF,WHS,WHR,DS,DR,WLS,WLR,MVB,MVD
      WRITE(4,9640) PD,PB,QR,R,DV,EFF
      WRITE(4,9645) FL,TFL,(XF(J), J = 1,NC)
      WRITE(4,9645) FV,TFV,(YF(J), J = 1,NC)
      WRITE(4,9650) TB,BLANK,(XB(J), J = 1,NC)
      DO 9760 N = 1,NT
      NN = N
      WRITE(4,9655) NN,T(N),LO(N),(X(N,J),J=1,NC)
```

```
9760 CONTINUE
     WRITE(4,9650) TB,R,(XD(J), J = 1,NC)
     GOTO 9000

CC       RUN SRK MODEL

CC
CC       key = 8.1
CC

CC
CC       Assign temperature in Degree Rangin
CC

9800 TFL = TFL + 460.
     TFV = TFV + 460.
     TB  = TB     + 460.
     TD  = TD     + 460.+125.0
     DO 18 N = 1,NT
     T(N) = T(N) + 460.
  18 CONTINUE

CC
CC    CALCULATE PRESSURE PROFILE
CC

     CALL drawlines()
     DO 35 N=1,NT
  35 P(N)=(PB-(N*(PB-PD))/NT)
     DELTA = 0.0001
     WRITE(*,37) DELTA
  37 FORMAT(1X,' DELTA = ', F8.5)
     TIME = 0.
     TPRINT = 0.
     CALL ENTHS(TFL,P(NF),XF,YF,HLF,HVF)
```

```
CC
CC    CALL INTITIAL HOLDUPS
CC

      CALL MWDENSS(XB,TB,PB,MWA,DENSA)

      MB=MVB*DENSA/MWA

      DO 20 N=1,NF
      DO 21 J=1,NC
   21 XX(J) = X(N,J)
      CALL MWDENSS(XX,T(N),P(N),MWA,DENSA)
      LV(N) = LO(N)*MWA/DENSA
      L(N) = LO(N)
      HFOW = (LV(N)/(999.*WLS))**.66667
      MV(N) = (HFOW+WHS/12.)*3.1416*DS*DS/(4.*144.)
      M(N) = MV(N)*DENSA/MWA

   20 CONTINUE
      DO 25 N = NF+1,NT
      DO 26 J = 1,NC
   26 XX(J) = X(N,J)
      CALL MWDENSS(XX,T(N),P(N),MWA,DENSA)
      LV(N) = LO(N)*MWA/DENSA
      L(N) = LO(N)
      HFOW = (LV(N)/(999.*WLR))**.66667
      MV(N) = (HFOW+WHR/12.)*3.1416*DR*DR/(4.*144.)
      M(N) = MV(N)*DENSA/MWA

   25 CONTINUE
      DO 30 N=1,NT
      DO 31 J=1,NC
      XM(N,J)=M(N)*X(N,J)
   31 CONTINUE
```

```
 30 CONTINUE
    CALL MWDENSS(XD,TD,PD,MWA,DENSA)
    MD=MVD*DENSA/MWA

CC
CC     Initial Guess of V(5) for first efficiency calculation
CC

    V(NF) = 822.

CC
CC    Main Loop for each time step
CC
    xxx=0

100 CONTINUE

    write(*,*) xxx
    CALL BUBPTS (TB,XB,YB,PB)
    CALL ENTHS (TB,PB,XB,YB,HLB,HVB)
    DO 105 J=1,NC

105 XX(J)=X(1,J)
    CALL BUBPTS (T(1),XX,YY,P(1))
    DO 106 J=1,NC

    Y(1,J)=YB(J)+EFF*(YY(J)-YB(J))
106 YY(J)=Y(1,J)
    CALL ENTHS(T(1),P(1),XX,YY,HL(1),HV(1))

    DO 110 N=2,NF
    DO 111 J=1,NC
111 XX(J)=X(N,J)
    CALL BUBPTS(T(N),XX,YY,P(N))
    DO 112 J = 1,NC
```

```
      Y(N,J)=(YY(J)-Y(N-1,J))*EFF+Y(N-1,J)
  112 YY(J)=Y(N,J)
      CALL ENTHS(T(N),P(N),XX,YY,HL(N),HV(N))

  110 CONTINUE
      DO 113 J=1,NC
  113 XX(J)=X(NF+1,J)
      CALL BUBPTS(T(NF+1),XX,YY,P(NF+1))
      DO 114 J=1,NC
      YAV(J)=(YF(J)*FV+Y(NF,J)*V(NF))/(V(NF)+FV)
      Y(NF+1,J)=(YY(J)-YAV(J))*EFF+YAV(J)
  114 YY(J)=Y(NF+1,J)
      CALL ENTHS(T(NF+1),P(NF+1),XX,YY,HL(NF+1),HV(NF+1))
      DO 115 N=NF+2,NT
      DO 116 J=1,NC
  116 XX(J)=X(N,J)
      CALL BUBPTS(T(N),XX,YY,P(N))
      DO 117 J=1,NC
      Y(N,J)=(YY(J)-Y(N-1,J))*EFF+Y(N-1,J)
  117 YY(J)=Y(N,J)
      CALL ENTHS (T(N),P(N),XX,YY,HL(N),HV(N))
  115 CONTINUE
      CALL BUBPTS (TD,XD,YD,PD)
      CALL ENTHS(TD,PD,XD,YD,HLD,HVD)

CC
CC      CALCULATE VAPOR RATES
CC

      VB=(QR*1000.-L(1)*(HLB-HL(1)))/(HVB-HLB)
      B = L(1)-VB
CC      IF (B .LT. 0.) THEN
CC      WRITE(*,*) ' ******** B IS LESS THAN 0 ********'
CC      STOP
```

```
CC       END IF

      V(1)=(HL(2)*L(2)+HVB*VB-HL(1)*L(1))/HV(1)
      DO 120 N=2,NF-1
      V(N)=(HL(N+1)*L(N+1)+HV(N-1)*V(N-1)-HL(N)*L(N))/HV(N)
  120 CONTINUE
      V(NF)=(HL(NF+1)*L(NF+1)+HV(NF-1)*V(NF-1)-HL(NF)*L(NF)+HLF*FL)/HV
     +(NF)
      V(NF+1)=(HL(NF+2)*L(NF+2)+HV(NF)*V(NF)+HVF*FV-HL(NF+1)*L(NF+1))/
     +HV(NF+1)
      DO 130 N=NF+2,NT-1
  130 V(N)=(HL(N+1)*L(N+1)+HV(N-1)*V(N-1)-HL(N)*L(N))/HV(N)
      V(NT)=(HLD*R+HV(NT-1)*V(NT-1)-HL(NT)*L(NT))/HV(NT)
      DL=V(NT)-DV-R
CC       IF (DL .LT. 0) THEN
CC       WRITE(*,*) ' ******** DL IS LESS THAN 0 ********'
CC       STOP
CC       END IF

CC
CC       EVALUATE DERIVATIVES
CC

      DM(1)=L(2)+VB-V(1)-L(1)
      DO 140 N=2,NF-1
  140 DM(N)=L(N+1)+V(N-1)-L(N)-V(N)
      DM(NF)=L(NF+1)+FL+V(NF-1)-L(NF)-V(NF)
      DM(NF+1)=L(NF+2)+FV+V(NF)-L(NF+1)-V(NF+1)
      DO 150 N=NF+2,NT-1
  150 DM(N)=L(N+1)+V(N-1)-L(N)-V(N)
      DM(NT)=R+V(NT-1)-L(NT)-V(NT)
      DO 160 J=1,NC
      DXB(J)=(X(1,J)*L(1)-YB(J)*VB-XB(J)*B)/MB
      DXM(1,J)=X(2,J)*L(2)+YB(J)*VB-X(1,J)*L(1)-Y(1,J)*V(1)
      DO 165 N=2,NF-1
```

```
165 DXM(N,J)=X(N+1,J)*L(N+1)+Y(N-1,J)*V(N-1)-X(N,J)*L(N)-V(N)*
   +Y(N,J)
    DXM(NF,J)=X(NF+1,J)*L(NF+1)+Y(NF-1,J)*V(NF-1)-X(NF,J)*L(NF)-
   +V(NF)*Y(NF,J)+FL*XF(J)
    DXM(NF+1,J)=X(NF+2,J)*L(NF+2)+Y(NF,J)*V(NF)-X(NF+1,J)*L(NF+1)
   +-V(NF+1)*Y(NF+1,J)+FV*YF(J)
    DO 170 N=NF+2,NT-1
170 DXM(N,J)=X(N+1,J)*L(N+1)+Y(N-1,J)*V(N-1)-X(N,J)*L(N)-V(N)*Y(N,J)
    DXM(NT,J)=XD(J)*R+Y(NT-1,J)*V(NT-1)-X(NT,J)*L(NT)-Y(NT,J)*V(NT)
    DXD(J)=(V(NT)*Y(NT,J)-DV*YD(J)-(R+DL)*XD(J))/MD
160 CONTINUE
    IF (TIME .GT. 4e-2) GOTO 9000
    IF (TIME .LT. TPRINT) GOTO 210

    CALL drawlines()
    WRITE(5,201)
201 FORMAT (5X,'TIME     T        X1       X2       X3       X4
   +     X5         L')
    WRITE(5,202) TIME,TB,(XB(J),J=1,NC),B
202 FORMAT (1X,F5.4,3X,F7.2,8F14.6,F10.1)
    DO 203 N=1,NT
203 WRITE(5,204) N,T(N),(X(N,J),J=1,NC),L(N)
204 FORMAT (3X,I3,3X,F7.2,8F14.6,F10.1)
    WRITE(5,205) TD,(XD(J),J=1,NC),R
205 FORMAT (9X,F7.2,8F14.6,F10.1)
    WRITE(5,206) (YD(J),J=1,NC),DL
206 FORMAT(16X,8F14.6,F10.1)
    TPRINT = TPRINT + .001
cc  CALL settextposition( 24, 5, curpos )
cc  CALL outtext( 'Press ENTER to continue ...' )
cc  READ(*,*)

CC
CC      INTEGRATION ALA EULER
CC
```

```
 210 TIME = TIME + DELTA
cc    WRITE(*,*) TIME
cc    READ(*,*)

      DO 215 N=1,NT
 215 M(N)=M(N)+DM(N)*DELTA
      DO 220 J = 1,NC
      XB(J)=XB(J)+DXB(J)*DELTA
          IF (XB(J) .LT. 0.) XB(J) = 0.0
          IF (XB(J) .GT. 1.) XB(J) = 1.
      DO 225 N=1,NT
      XM(N,J)=XM(N,J)+DXM(N,J)*DELTA
      X(N,J)=XM(N,J)/M(N)
          IF (X(N,J) .GT. 1.) X(N,J) = 1.
          IF (X(N,J) .LT. 0.) X(N,J) = 0.0
 225 CONTINUE
      XD(J)=XD(J)+DXD(J)*DELTA
          IF (XD(J) .LT. 0.) XD(J)=0.
          IF (XD(J) .GT. 1.) XD(J)=1.
 220 CONTINUE

CC
CC    CALCULATE NEW LIQUID RATES
CC

      DO 270 N=1,NF
      DO 271 J=1,NC
      XX(J)=X(N,J)
 271 CONTINUE
      CALL HYDRAUS(P(N),M(N),T(N),XX,L(N),WHS,WLS,DS)
 270 CONTINUE
      DO 273 N=NF+1,NT
      DO 275 J=1,NC
 275 XX(J)=X(N,J)
```

```
      CALL HYDRAUS(P(N),M(N),T(N),XX,L(N),WHR,WLR,DR)
  273 CONTINUE
      xxx=xxx+1.0
      GOTO 100
  400 STOP
      END

CC
CC      Calculate Liquid tray holdup by Francis Wier equation
CC

      SUBROUTINE  HYDRAUS(P,M,T,X,L,WH,WL,DCOL)

      REAL M,L,MW,MWA

      COMMON NC,MW(20),DENS(20),C1(20),C2(20),C3(20),BPT(20),AVP(20),
     +BVP(20),tc(20),pc(20),zc(20),w(20),a1(20),a2(20),a3(20),a4(20),
     +a5(20)

      DIMENSION X(20)

      CALL MWDENSS(X,T,P,MWA,DENSA)
      CONST=183.2*M*MWA/(DENSA*DCOL*DCOL)-WH/12.
      IF (CONST .LE. 0.) GOTO 10
      L=DENSA*WL*999.*((183.2*M*MWA/(DENSA*DCOL*DCOL)-WH/12.)**1.5)/MWA
      RETURN
   10 L=0.
      RETURN
      END

CC
CC      ENTHSalpy calculation
CC

      SUBROUTINE ENTHS(T,P,X,Y,HL,HV)
```

```
      REAL GC,MI

      COMMON NC,MW(20),DENS(20),C1(20),C2(20),C3(20),BPT(20),AVP(20),
     +BVP(20),tc(20),pc(20),zc(20),w(20),a1(20),a2(20),a3(20),a4(20),
     +a5(20)

      DIMENSION X(20),Y(20),TR(20),MI(20),ALPHA(20),ACI(20),AI(20),
     +BI(20),HLS(20)

CC
CC       Define constant value
CC

      GC = 1.986
      KIJ = 0.001

      HL = 0.0
      HV = 0.0
      HLS1 = 0.0
      HVS  = 0.0
      SUMAL  = 0.0
      SUMAV  = 0.0
      SUMBL  = 0.0
      SUMBV  = 0.0
      SUMATL = 0.0
      SUMATV = 0.0

      DO 1 J=1,NC
      TR(J) = T/TC(J)
      MI(J) = 0.48+1.574*W(J)-0.176*W(J)**2
      ALPHA(J) = (1.0+MI(J)*(1-TR(J)**(0.5)))**2
      ACI(J) = 0.42748*(GC*TC(J))**2/PC(J)
      AI(J) = ALPHA(J)*ACI(J)
      BI(J) = 0.08664*GC*TC(J)/PC(J)
```

```
      SUMBL = SUMBL + X(J)*BI(J)
      SUMBV = SUMBV + Y(J)*BI(J)
    1 CONTINUE

      DO 2 I = 1,NC
      DO 3 J = 1,NC
      SUMAL = SUMAL+X(I)*X(J)*(AI(I)*AI(J))**0.5*(1-KIJ)
      SUMAV = SUMAV+Y(I)*Y(J)*(AI(I)*AI(J))**0.5*(1-KIJ)
      SUMATL = SUMATL+X(I)*X(J)*MI(J)*(AI(I)*ACI(J)*TR(J))**0.5*(1-KIJ)
      SUMATV = SUMATV+Y(I)*Y(J)*MI(J)*(AI(I)*ACI(J)*TR(J))**0.5*(1-KIJ)
    3 CONTINUE
    2 CONTINUE

      AL = SUMAL*P/(GC*T)**2
      AV = SUMAV*P/(GC*T)**2
      BL = SUMBL*P/(GC*T)
      BV = SUMBV*P/(GC*T)
      CALL NEWTONS(AL,BL,Z1,Z2,Z3)
      IF ( Z1 .LE. 0 ) THEN
            IF ( Z2 .LE. 0 ) THEN
              ZL = Z3
            ELSE
              ZL = Z2
            END IF
      ELSE
            ZL = Z1
      END IF
      CALL NEWTONS(AV,BV,Z1,Z2,Z3)
      ZV = Z3

      DO 4 I = 1,NC
      HLS(I) = A1(I)*(T-460.)+A2(I)/2*(T-460.)**2+A3(I)/3*(T-460.)**3+
     +A4(I)/4*(T-460.)**4+A5(I)/5*(T-460.)**5
      HLS1 =  HLS1+X(I)*HLS(I)
      HVS  =  HVS +Y(I)*HLS(I)
```

```
 4 CONTINUE

    HL = (ZL-1-AL/BL*(1-SUMATL/SUMAL)*ALOG(1+BL/ZL))*GC*T+HLS1
    HV = (ZV-1-AV/BV*(1-SUMATV/SUMAV)*ALOG(1+BV/ZV))*GC*T+HVS

    RETURN
    END

CC
CC      Calculate the root of Z by NEWTONS's technique
CC

    SUBROUTINE NEWTONS(A,B,Z1,Z2,Z3)

    DIMENSION Z(3)

    ZO1 = -10
    L = 1
 10 IF ( L .GT. 8000 ) THEN
        WRITE(*,*) '   ****** NOT CONVERGE !! ******'
        STOP
    END IF
    FZ1 = ZO1**3-ZO1**2+(A-B-B**2)*ZO1-A*B
    DFZ1 = 3*ZO1**2-2*ZO1+(A-B-B**2)
    Z1 = ZO1-FZ1/DFZ1
    IF (ABS(ZO1-Z1) .LT. 1E-3) THEN
    GOTO 20
    END IF
    ZO1 = Z1
    L = L+1
    GOTO 10

 20 ZO2 = 0
    L1 = 1
 25 IF ( L1 .GT. 8000 ) THEN
```

```
      WRITE(*,*) '   ****** NOT CONVERGE !! ******'
      STOP
    END IF
    FZ2 = ZO2**3-ZO2**2+(A-B-B**2)*ZO2-A*B
    DFZ2 = 3*ZO2**2-2*ZO2+(A-B-B**2)
    Z2 = ZO2-FZ2/DFZ2
    IF (ABS(ZO2-Z2) .LT. 1E-3) THEN
    GOTO 30
    END IF
    ZO2 = Z2
    L1 = L1+1
    GOTO 25

30  ZO3 = 10
    L2 = 1
35  IF ( L2 .GT. 8000 ) THEN
      WRITE(*,*) '   ****** NOT CONVERGE !! ******'
      STOP
    END IF
    FZ3 = ZO3**3-ZO3**2+(A-B-B**2)*ZO3-A*B
    DFZ3 = 3*ZO3**2-2*ZO3+(A-B-B**2)
    Z3 = ZO3-FZ3/DFZ3
    IF (ABS(ZO3-Z3) .LT. 1E-3) THEN
    GOTO 40
    END IF
    ZO3 = Z3
    L2 = L2+1
    GOTO 35

    Z(1) = Z1
    Z(2) = Z2
    Z(3) = Z3

    jump = 3
50  jump = jump/2
```

```
      IF ( jump .NE. 0 ) THEN
          j2 = 3-jump
          DO 60 j = 1,j2
            DO 60 i = j,1,-jump
            j3 = i+jump
            IF ( Z(i) .GT. Z(j3)) THEN
                  s = Z(i)
                  Z(i) = Z(j3)
                  Z(j3) = s
            END IF
   60     CONTINUE
        GOTO 50
        END IF

        Z1 = Z(1)
        Z2 = Z(2)
        Z3 = Z(3)

   40 RETURN
        END

CC
CC      Calculate average density
CC

        SUBROUTINE MWDENSS(X,T,P,MWA,DENSA)

        COMMON NC,MW(20),DENS(20),C1(20),C2(20),C3(20),BPT(20),AVP(20),
       +BVP(20),tc(20),pc(20),zc(20),w(20),a1(20),a2(20),a3(20),a4(20),
       +a5(20)

        DIMENSION X(20),TR(20),MI(20),ALPHA(20),ACI(20),AI(20),BI(20)

        REAL MW,MWA,MI,GC
        DENSA=0.0
```

```
      MWA=0.

CC
CC      Define constant value
CC

      GC = 10.731
      KIJ = 0.001

      SUMAL  = 0.0
      SUMBL  = 0.0
      SUMATL = 0.0

      DO 1 J=1,NC
      TR(J) = T/TC(J)
      MI(J) = 0.48+1.574*W(J)-0.176*W(J)**2
      ALPHA(J) = (1.0+MI(J)*(1-TR(J)**(0.5)))**2
      ACI(J) = 0.42748*(GC*TC(J))**2/PC(J)
      AI(J) = ALPHA(J)*ACI(J)
      BI(J) = 0.08664*GC*TC(J)/PC(J)
      SUMBL = SUMBL + X(J)*BI(J)
    1 CONTINUE

      DO 2 I = 1,NC
      DO 3 J = 1,NC
      SUMAL = SUMAL+X(I)*X(J)*(AI(I)*AI(J))**0.5*(1-KIJ)
      SUMATL = SUMATL+X(I)*X(J)*MI(J)*(AI(I)*ACI(J)*TR(J))**0.5*(1-KIJ)
    3 CONTINUE
    2 CONTINUE

      AL = SUMAL*P/(GC*T)**2
      BL = SUMBL*P/(GC*T)

      CALL NEWTONS(AL,BL,Z1,Z2,Z3)
      IF ( Z1 .LE. 0 ) THEN
```

```
        IF ( Z2 .LE. 0 ) THEN
          ZL = Z3
        ELSE
          ZL = Z2
        END IF
      ELSE
        ZL = Z1
      END IF

      VV = ZL*GC*T/P

      DO 5 J=1,NC
    5 MWA=X(J)*MW(J)+MWA
      DENSA=1/(VV/MWA)

      RETURN
      END

CC
CC      Buble point Calculation
CC

      SUBROUTINE BUBPTS(T,X,Y,P)

      REAL GC;MI,KI, APRIMEX,APRIMEY,APRIMEXH,APRIMEYH,KIH,
     +FSLOPE,F,y

      COMMON NC,MW(20),DENS(20),C1(20),C2(20),C3(20),BPT(20),AVP(20),
     +BVP(20),tc(20),pc(20),zc(20),w(20),a1(20),a2(20),a3(20),a4(20),
     +a5(20)

      DIMENSION X(20),Y(20),TR(20),MI(20),ALPHA(20),ACI(20),
     +AI(20),BI(20),FIL(20),FIV(20),KI(20),APRIMEX(20),APRIMEY(20)

      DIMENSION FILH(20),FIVH(20),APRIMEXH(20),APRIMEYH(20),KIH(20),
```

```
     +ALPHAH(20),AIH(20),TRH(20)

CC
CC        Define constant value
CC

      GC = 1.986
      KIJ = 0.001

      DO 40 I = 1,NC
      IF (Y(I) .EQ. 0.) THEN
      ki(i) = 1.
      Y(I) = KI(I)*X(I)
      END IF
   40 continue


      LOOP = 0
   10 LOOP=LOOP+1

      SUMXA  = 0.0
      SUMYA  = 0.0
      SUMXAH = 0.0
      SUMYAH = 0.0
      SUMAL  = 0.0
      SUMAV  = 0.0
      SUMALH = 0.0
      SUMAVH = 0.0
      SUMBL  = 0.0
      SUMBV  = 0.0

      TH = T+1.0

      IF(LOOP .GT. 8000) GOTO 30
      SUMY=0.0
```

```
      SUMYH = 0.0

      DO 1 J=1,NC
      TR(J) = T/TC(J)
      TRH(J) = TH/TC(J)
      MI(J) = 0.48+1.574*W(J)-0.176*W(J)**2
      ALPHA(J) = (1.0+MI(J)*(1-TR(J)**(0.5)))**2
      ALPHAH(J) = (1.0+MI(J)*(1-TRH(J)**(0.5)))**2
      ACI(J) = 0.42748*(GC*TC(J))**2/PC(J)
      AI(J) = ALPHA(J)*ACI(J)
      AIH(J) = ALPHAH(J)*ACI(J)
      BI(J) = 0.08664*GC*TC(J)/PC(J)
      SUMXA = SUMXA+X(J)*AI(J)**0.5*(1-KIJ)
      SUMXAH = SUMXAH+X(J)*AIH(J)**0.5*(1-KIJ)
      SUMYA = SUMYA+Y(J)*AI(J)**0.5*(1-KIJ)
      SUMYAH = SUMYAH+Y(J)*AIH(J)**0.5*(1-KIJ)
      SUMBL = SUMBL + X(J)*BI(J)
      SUMBV = SUMBV + Y(J)*BI(J)
    1 CONTINUE

      DO 2 I = 1,NC
      DO 3 J = 1,NC
      SUMAL = SUMAL+X(I)*X(J)*(AI(I)*AI(J))**0.5*(1-KIJ)
      SUMALH = SUMALH + X(I)*X(J)*(AIH(I)*AIH(J))**0.5*(1-KIJ)
      SUMAV = SUMAV+Y(I)*Y(J)*(AI(I)*AI(J))**0.5*(1-KIJ)
      SUMAVH = SUMAVH + Y(I)*Y(J)*(AIH(I)*AIH(J))**0.5*(1-KIJ)
    3 CONTINUE
    2 CONTINUE

      AL = SUMAL*P/(GC*T)**2
      ALH = SUMALH*P/(GC*TH)**2
      AV = SUMAV*P/(GC*T)**2
      AVH = SUMAVH*P/(GC*TH)**2
      BL = SUMBL*P/(GC*T)
      BLH = SUMBL*P/(GC*TH)
```

```
BV  = SUMBV*P/(GC*T)
BVH = SUMBL*P/(GC*TH)

CALL NEWTONS(AL,BL,Z1,Z2,Z3)
IF ( Z1 .LE. 0 ) THEN
      IF ( Z2 .LE. 0 ) THEN
        ZL = Z3
      ELSE
        ZL = Z2
      END IF
ELSE
      ZL = Z1
END IF
CALL NEWTONS(ALH,BLH,Z1,Z2,Z3)
IF ( Z1 .LE. 0 ) THEN
      IF ( Z2 .LE. 0 ) THEN
        ZLH = Z3
      ELSE
        ZLH = Z2
      END IF
ELSE
      ZLH = Z1
END IF
CALL NEWTONS(AV,BV,Z1,Z2,Z3)
ZV = Z3
CALL NEWTONS(AVH,BVH,Z1,Z2,Z3)
ZVH = Z3

DO 15 I=1,NC
APRIMEX(I) = 1.0/SUMAL*(2*AI(I)**0.5*SUMXA)
APRIMEXH(I) = 1.0/SUMALH*(2*AIH(I)**0.5*SUMXAH)
APRIMEY(I) = 1.0/SUMAV*(2*AI(I)**0.5*SUMYA)
APRIMEYH(I) = 1.0/SUMAVH*(2*AIH(I)**0.5*SUMYAH)
FIL(I) = EXP(BI(I)/SUMBL*(ZL-1)-ALOG(ZL-BL)-AL/BL*(APRIMEX(I)-
+BI(I)/SUMBL)*ALOG(1+BL/ZL))
```

```
      FILH(I) = EXP(BI(I)/SUMBL*(ZLH-1)-ALOG(ZLH-BLH)-ALH/BLH*
     +(APRIMEXH(I)-BI(I)/SUMBL)*ALOG(1+BLH/ZLH))
      FIV(I) = EXP(BI(I)/SUMBV*(ZV-1)-ALOG(ZV-BV)-AV/BV*(APRIMEY(I)-
     +BI(I)/SUMBV)*ALOG(1+BV/ZV))
      FIVH(I) = EXP(BI(I)/SUMBV*(ZVH-1)-ALOG(ZVH-BVH)-AVH/BVH*
     +(APRIMEYH(I)-BI(I)/SUMBV)*ALOG(1+BVH/ZVH))
      KI(I) = FIL(I)/FIV(I)
      KIH(I) = FILH(I)/FIVH(I)
      SUMY = SUMY +X(I)*KI(I)
      SUMYH = SUMYH +X(I)*KIH(I)
   15 CONTINUE

      IF(LOOP .EQ. 1) THEN
      DO 16 I = 1,NC
      Y(I) = KI(I)*X(I)/SUMY
   16 CONTINUE
      GOTO 10
      END IF

      IF (ABS(SUMY-SUMYZ) .GE. 1E-3) THEN

      DO 17 I = 1,NC
      Y(I) = KI(I)*X(I)/SUMY

   17 CONTINUE
      SUMYZ = SUMY
      GOTO 10
      END IF

      IF ( ABS(SUMY-1.) .LT. 1E-3) RETURN
      F=SUMY-1.0
      FH = SUMYH-1.0
      FSLOPE=(FH-F)/1.
      T=T-F/FSLOPE
      GOTO 10
```

```
1000 WRITE(*,*) '   ********** Converged !!'
     read(*,*)
     WRITE(*,*) '          TEMP          SUMY          LOOP '
     WRITE(*,*)
     WRITE(*,*) T, SUMY, LOOP
     READ(*,*)
     WRITE(*,*)
     WRITE(*,*) '      yi              ki'
     write(*,*)
     DO 1010 I = 1,NC
     WRITE(*,*) Y(I),KI(I)
     READ(*,*)
1010 CONTINUE
     stop

  30 WRITE(*,*) 'The calculation does not converge !!'
     STOP
     END

CC
CC       Draw Large Border lines
CC

     SUBROUTINE drawlines()

     INCLUDE  'FGRAPH.FD'
     RECORD / rccoord / curpos

     dummy = setvideomode( $ERESCOLOR )
     CALL clearscreen( $GCLEARSCREEN )
     dummy2 = setcolor( 3 )
     CALL     moveto( 5, 5, curpos )
     dummy2 = lineto( 5, 340 )
     dummy2 = lineto( 635,340 )
```

```
      dummy2 = lineto( 635,5 )
      dummy2 = lineto( 5, 5 )
      CALL          moveto( 7, 7, curpos )
      dummy2 = lineto( 7, 338)
      dummy2 = lineto( 633, 338)
      dummy2 = lineto( 633, 7 )
      dummy2 = lineto( 7, 7 )

      RETURN
      END

CC
CC      Draw Small Border lines
CC

      SUBROUTINE drawlines1()

      INCLUDE  'FGRAPH.FD'
      RECORD / rccoord / curpos

      dummy2 = setcolor( 4 )
      CALL          moveto( 120, 80, curpos )
      dummy2 = lineto( 120, 270 )
      dummy2 = lineto( 520, 270 )
      dummy2 = lineto( 520, 80 )
      dummy2 = lineto( 120,80 )
      CALL          moveto( 122, 82, curpos )
      dummy2 = lineto( 122, 268 )
      dummy2 = lineto( 518, 268 )
      dummy2 = lineto( 518, 82 )
      dummy2 = lineto( 122, 82 )

      RETURN
      END
```

```
CC
CC      Draw Middle Border lines
CC

      SUBROUTINE drawlines2()

      INCLUDE  'FGRAPH.FD'
      RECORD / rccoord / curpos

      dummy2 = setcolor( 4 )
      CALL          moveto( 80, 80, curpos )
      dummy2 = lineto( 80, 270 )
      dummy2 = lineto( 560, 270 )
      dummy2 = lineto( 560, 80 )
      dummy2 = lineto( 80,80 )
      CALL          moveto( 82, 82, curpos )
      dummy2 = lineto( 82, 268 )
      dummy2 = lineto( 558, 268 )
      dummy2 = lineto( 558, 82 )
      dummy2 = lineto( 82, 82 )

      RETURN
      END

CC
CC      Draw Component Border lines
CC

      SUBROUTINE drawlines3()

      INCLUDE  'FGRAPH.FD'
      RECORD / rccoord / curpos

      dummy2 = setcolor( 4 )
      CALL          moveto( 60, 60, curpos )
```

```
      dummy2 = lineto( 60, 290 )
      dummy2 = lineto( 580, 290 )
      dummy2 = lineto( 580, 60 )
      dummy2 = lineto( 60,60 )
      CALL          moveto( 62, 62, curpos )
      dummy2 = lineto( 62, 288 )
      dummy2 = lineto( 578, 288 )
      dummy2 = lineto( 578, 62 )
      dummy2 = lineto( 62, 62 )

      RETURN
      END

CC
CC        Data Menu
CC

      SUBROUTINE datamenu(dat)

      INCLUDE  'FGRAPH.FD'
      RECORD / rccoord / curpos

  100 key2 = -1
      DO WHILE ((key2 .LT. 0) .OR. (key2 .GT. 4))
      CALL drawlines()
      CALL drawlines1()
      dummy2 = settextcolor( 14 )
      CALL settextposition( 4, 30, curpos )
      CALL outtext( '**** Data Menu ****' )
      dummy2 = settextcolor( 15 )
      CALL settextposition( 10, 25, curpos )
      CALL outtext( '0) Return to Main Menu' )
      CALL settextposition( 12, 25, curpos )
      CALL outtext( '1) New Data' )
      CALL settextposition( 14, 25, curpos )
```

```
      CALL outtext( '2) Old Data' )
      CALL settextposition( 16, 25, curpos )
      CALL outtext( '3) Save Data' )
      dummy2 = settextcolor( 14 )
      CALL settextposition( 22, 25, curpos )
      CALL outtext( 'Please ENTER your selection : ' )
      READ(*,*,ERR = 100) key2
      dat = key2
      END DO
      dummy2 = setvideomode( $DEFAULTMODE )

      DO WHILE (key2 .EQ. 0)
          RETURN
      END DO

      DO WHILE (key2 .EQ. 1)
          RETURN
      END DO

      DO WHILE (key2 .EQ. 2)
          CALL old()
          RETURN
      END DO

      DO WHILE (key2 .EQ. 3)
          CALL save()
          RETURN
      END DO

      END

CC
CC        Edit old data
CC
```

```
      SUBROUTINE old()

      INCLUDE  'FGRAPH.FD'
      RECORD / rccoord / curpos
      CHARACTER*20 fname

  100 CALL drawlines()
      CALL drawlines1()
      dummy2 = settextcolor( 14 )
      CALL settextposition( 4, 28, curpos )
      CALL outtext( '*** Edit Old Data ***' )
      CALL settextposition( 13, 18, curpos )
      CALL outtext( 'Please type data filename :  ' )
      READ(*,120) fname
      OPEN( 3, FILE = fname, ERR = 100)
  120 FORMAT(A)

      RETURN
      END

CC
CC        Save data
CC

      SUBROUTINE save()

      INCLUDE  'FGRAPH.FD'
      RECORD / rccoord / curpos
      CHARACTER*20 filename

  100 CALL drawlines()
      CALL drawlines1()
      dummy2 = settextcolor( 14 )
      CALL settextposition( 4, 28, curpos )
      CALL outtext( '*** Save Data ***' )
```

```
      CALL settextposition( 13, 18, curpos )
      CALL outtext( 'Please type data filename : ' )
      READ(*,120) filename
      OPEN( 4, FILE = filename, ERR = 100)
  120 FORMAT(A)

      RETURN
      END
```

```
CC      RUN GRK MODEL

CC
CC      key = 8.2
CC


CC
CC      Assign temperature in Degree Rangin
CC


9800 TFL = TFL + 460.
     TFV = TFV + 460.
     TB  = TB     + 460.
     TD  = TD     + 460.+125.0
     DO 18 N = 1,NT
     T(N) = T(N) + 460.
  18 CONTINUE

CC
CC    CALCULATE PRESSURE PROFILE
CC


     CALL drawlines()
     DO 35 N=1,NT
  35 P(N)=(PB-(N*(PB-PD))/NT)
     DELTA = 0.0001
     WRITE(*,37) DELTA
  37 FORMAT(1X,' DELTA = ', F8.5)
     TIME = 0.
     TPRINT = 0.
     CALL ENTHGR(TFL,P(NF),XF,YF,HLF,HVF)


CC
CC    CALL INTITIAL HOLDUPS
CC
```

```
      CALL MWDENSG(XB,TB,PB,MWA,DENSA)

      MB=MVB*DENSA/MWA

      DO 20 N=1,NF
      DO 21 J=1,NC
   21 XX(J) = X(N,J)
      CALL MWDENSG(XX,T(N),P(N),MWA,DENSA)
      LV(N) = LO(N)*MWA/DENSA
      L(N) = LO(N)
      HFOW = (LV(N)/(999.*WLS))**.66667
      MV(N) = (HFOW+WHS/12.)*3.1416*DS*DS/(4.*144.)
      M(N) = MV(N)*DENSA/MWA

   20 CONTINUE
      DO 25 N = NF+1,NT
      DO 26 J = 1,NC
   26 XX(J) = X(N,J)
      CALL MWDENSG(XX,T(N),P(N),MWA,DENSA)
      LV(N) = LO(N)*MWA/DENSA
      L(N) = LO(N)
      HFOW = (LV(N)/(999.*WLR))**.66667
      MV(N) = (HFOW+WHR/12.)*3.1416*DR*DR/(4.*144.)
      M(N) = MV(N)*DENSA/MWA

   25 CONTINUE
      DO 30 N=1,NT
      DO 31 J=1,NC
      XM(N,J)=M(N)*X(N,J)
   31 CONTINUE
   30 CONTINUE
      CALL MWDENSG(XD,TD,PD,MWA,DENSA)
      MD=MVD*DENSA/MWA
```

```
CC
CC     Initial Guess of V(5) for first efficiency calculation
CC

      V(NF) = 822.

CC
CC     Main Loop for each time step
CC
      xxx=0

  100 CONTINUE

      write(*,*) xxx
      CALL BUBPTGR (TB,XB,YB,PB)
      CALL ENTHGR (TB,PB,XB,YB,HLB,HVB)
      DO 105 J=1,NC

  105 XX(J)=X(1,J)
      CALL BUBPTGR (T(1),XX,YY,P(1))
      DO 106 J=1,NC

      Y(1,J)=YB(J)+EFF*(YY(J)-YB(J))
  106 YY(J)=Y(1,J)
      CALL ENTHGR (T(1),P(1),XX,YY,HL(1),HV(1))

      DO 110 N=2,NF
      DO 111 J=1,NC
  111 XX(J)=X(N,J)
      CALL BUBPTGR (T(N),XX,YY,P(N))
      DO 112 J = 1,NC

      Y(N,J)=(YY(J)-Y(N-1,J))*EFF+Y(N-1,J)
  112 YY(J)=Y(N,J)
      CALL ENTHGR(T(N),P(N),XX,YY,HL(N),HV(N))
```

```
110 CONTINUE
    DO 113 J=1,NC
113 XX(J)=X(NF+1,J)
    CALL BUBPTGR(T(NF+1),XX,YY,P(NF+1))
    DO 114 J=1,NC
    YAV(J)=(YF(J)*FV+Y(NF,J)*V(NF))/(V(NF)+FV)
    Y(NF+1,J)=(YY(J)-YAV(J))*EFF+YAV(J)
114 YY(J)=Y(NF+1,J)
    CALL ENTHGR(T(NF+1),P(NF+1),XX,YY,HL(NF+1),HV(NF+1))
    DO 115 N=NF+2,NT
    DO 116 J=1,NC
116 XX(J)=X(N,J)
    CALL BUBPTGR(T(N),XX,YY,P(N))
    DO 117 J=1,NC
    Y(N,J)=(YY(J)-Y(N-1,J))*EFF+Y(N-1,J)
117 YY(J)=Y(N,J)
    CALL ENTHGR (T(N),P(N),XX,YY,HL(N),HV(N))
115 CONTINUE
    CALL BUBPTGR (TD,XD,YD,PD)
    CALL ENTHGR(TD,PD,XD,YD,HLD,HVD)

CC
CC      CALCULATE VAPOR RATES
CC

    VB=(QR*1000.-L(1)*(HLB-HL(1)))/(HVB-HLB)
    B = L(1)-VB
CC      IF (B .LT. 0.) THEN
CC      WRITE(*,*) ' ******** B IS LESS THAN 0 ********'
CC      STOP
CC      END IF

    V(1)=(HL(2)*L(2)+HVB*VB-HL(1)*L(1))/HV(1)
    DO 120 N=2,NF-1
```


```
      V(N)=(HL(N+1)*L(N+1)+HV(N-1)*V(N-1)-HL(N)*L(N))/HV(N)
  120 CONTINUE
      V(NF)=(HL(NF+1)*L(NF+1)+HV(NF-1)*V(NF-1)-HL(NF)*L(NF)+HLF*FL)/HV
     +(NF)
      V(NF+1)=(HL(NF+2)*L(NF+2)+HV(NF)*V(NF)+HVF*FV-HL(NF+1)*L(NF+1))/
     +HV(NF+1)
      DO 130 N=NF+2,NT-1
  130 V(N)=(HL(N+1)*L(N+1)+HV(N-1)*V(N-1)-HL(N)*L(N))/HV(N)
      V(NT)=(HLD*R+HV(NT-1)*V(NT-1)-HL(NT)*L(NT))/HV(NT)
      DL=V(NT)-DV-R
CC        IF (DL .LT. 0) THEN
CC        WRITE(*,*) ' ******** DL IS LESS THAN 0 ********'
CC        STOP
CC        END IF

CC
CC        EVALUATE DERIVATIVES
CC

      DM(1)=L(2)+VB-V(1)-L(1)
      DO 140 N=2,NF-1
  140 DM(N)=L(N+1)+V(N-1)-L(N)-V(N)
      DM(NF)=L(NF+1)+FL+V(NF-1)-L(NF)-V(NF)
      DM(NF+1)=L(NF+2)+FV+V(NF)-L(NF+1)-V(NF+1)
      DO 150 N=NF+2,NT-1
  150 DM(N)=L(N+1)+V(N-1)-L(N)-V(N)
      DM(NT)=R+V(NT-1)-L(NT)-V(NT)
      DO 160 J=1,NC
      DXB(J)=(X(1,J)*L(1)-YB(J)*VB-XB(J)*B)/MB
      DXM(1,J)=X(2,J)*L(2)+YB(J)*VB-X(1,J)*L(1)-Y(1,J)*V(1)
      DO 165 N=2,NF-1
  165 DXM(N,J)=X(N+1,J)*L(N+1)+Y(N-1,J)*V(N-1)-X(N,J)*L(N)-V(N)*
     +Y(N,J)
      DXM(NF,J)=X(NF+1,J)*L(NF+1)+Y(NF-1,J)*V(NF-1)-X(NF,J)*L(NF)-
     +V(NF)*Y(NF,J)+FL*XF(J)
```

```
      DXM(NF+1,J)=X(NF+2,J)*L(NF+2)+Y(NF,J)*V(NF)-X(NF+1,J)*L(NF+1)
     +-V(NF+1)*Y(NF+1,J)+FV*YF(J)
      DO 170 N=NF+2,NT-1
  170 DXM(N,J)=X(N+1,J)*L(N+1)+Y(N-1,J)*V(N-1)-X(N,J)*L(N)-V(N)*Y(N,J)
      DXM(NT,J)=XD(J)*R+Y(NT-1,J)*V(NT-1)-X(NT,J)*L(NT)-Y(NT,J)*V(NT)
      DXD(J)=(V(NT)*Y(NT,J)-DV*YD(J)-(R+DL)*XD(J))/MD
  160 CONTINUE
      IF (TIME .GT. 3e-2) GOTO 9000
      IF (TIME .LT. TPRINT) GOTO 210

      CALL drawlines()
      WRITE(5,201)
  201 FORMAT (5X,'TIME     T        X1       X2       X3       X4
     +     X5         L')
      WRITE(5,202) TIME,TB,(XB(J),J=1,NC),B
  202 FORMAT (1X,F5.4,3X,F7.2,8F14.6,F10.1)
      DO 203 N=1,NT
  203 WRITE(5,204) N,T(N),(X(N,J),J=1,NC),L(N)
  204 FORMAT (3X,I3,3X,F7.2,8F14.6,F10.1)
      WRITE(5,205) TD,(XD(J),J=1,NC),R
  205 FORMAT (9X,F7.2,8F14.6,F10.1)
      WRITE(5,206) (YD(J),J=1,NC),DL
  206 FORMAT(16X,8F14.6,F10.1)
      TPRINT = TPRINT + .001
cc    CALL settextposition( 24, 5, curpos )
cc    CALL outtext( 'Press ENTER to continue ...' )
cc    READ(*,*)

CC
CC       INTEGRATION ALA EULER
CC

  210 TIME = TIME + DELTA
cc    WRITE(*,*) TIME
cc    READ(*,*)
```

```
      DO 215 N=1,NT
  215 M(N)=M(N)+DM(N)*DELTA
      DO 220 J = 1,NC
      XB(J)=XB(J)+DXB(J)*DELTA
          IF (XB(J) .LT. 0.) XB(J) = 0.0
          IF (XB(J) .GT. 1.) XB(J) = 1.
      DO 225 N=1,NT
      XM(N,J)=XM(N,J)+DXM(N,J)*DELTA
      X(N,J)=XM(N,J)/M(N)
          IF (X(N,J) .GT. 1.) X(N,J) = 1.
          IF (X(N,J) .LT. 0.) X(N,J) = 0.0
  225 CONTINUE
      XD(J)=XD(J)+DXD(J)*DELTA
          IF (XD(J) .LT. 0.) XD(J)=0.
          IF (XD(J) .GT. 1.) XD(J)=1.
  220 CONTINUE

CC
CC    CALCULATE NEW LIQUID RATES
CC

      DO 270 N=1,NF
      DO 271 J=1,NC
      XX(J)=X(N,J)
  271 CONTINUE
      CALL HYDRAU(P(N),M(N),T(N),XX,L(N),WHS,WLS,DS)
  270 CONTINUE
      DO 273 N=NF+1,NT
      DO 275 J=1,NC
  275 XX(J)=X(N,J)
      CALL HYDRAU(P(N),M(N),T(N),XX,L(N),WHR,WLR,DR)
  273 CONTINUE
      xxx=xxx+1.0
      GOTO 100
```

```
400 STOP
    END

CC
CC      Calculate Liquid tray holdup by Francis Wier equation
CC

    SUBROUTINE  HYDRAU(P,M,T,X,L,WH,WL,DCOL)

    REAL M,L,MW,MWA

    COMMON NC,MW(20),DENS(20),C1(20),C2(20),C3(20),BPT(20),AVP(20),
   +BVP(20),tc(20),pc(20),zc(20),w(20),a1(20),a2(20),a3(20),a4(20),
   +a5(20)

    DIMENSION X(20)

    CALL MWDENSG(X,T,P,MWA,DENSA)
    CONST=183.2*M*MWA/(DENSA*DCOL*DCOL)-WH/12.
    IF (CONST .LE. 0.) GOTO 10
    L=DENSA*WL*999.*((183.2*M*MWA/(DENSA*DCOL*DCOL)-WH/12.)**1.5)/MWA
    RETURN
 10 L=0.
    RETURN
    END

CC
CC      Enthalpy calculation
CC

    SUBROUTINE ENTHGR(T,P,X,Y,HL,HV)

    REAL GC,MI

    COMMON NC,MW(20),DENS(20),C1(20),C2(20),C3(20),BPT(20),AVP(20),
```

```
     +BVP(20),tc(20),pc(20),zc(20),w(20),a1(20),a2(20),a3(20),a4(20),
     +a5(20)

      DIMENSION X(20),Y(20),TR(20),MI(20),ALPHA(20),ACI(20),AI(20),
     +BI(20),HLS(20)

CC
CC       Define constant value
CC

      GC = 1.986
      KIJ = 0.001

      HL = 0.0
      HV = 0.0
      HLS1 = 0.0
      HVS  = 0.0
      SUMAL  = 0.0
      SUMAV  = 0.0
      SUMBL  = 0.0
      SUMBV  = 0.0
      SUMATL = 0.0
      SUMATV = 0.0

      DO 1 J=1,NC
      TR(J) = T/TC(J)
      ACI(J) = (0.4278/(TR(J)**2.5*PC(J)))**0.5
      AI(J) = ACI(J)
      BI(J) = 0.0867/TR(J)/PC(J)
      SUMBL = SUMBL + X(J)*BI(J)
      SUMBV = SUMBV + Y(J)*BI(J)
      SUMAL = SUMAL + X(J)*AI(J)
      SUMAV = SUMAV + Y(J)*AI(J)
    1 CONTINUE
```

```
      AL = SUMAL
      AV = SUMAV
      BL = SUMBL
      BV = SUMBV
      CALL NEWTON(AL,BL,P,Z1,Z2,Z3)
      IF ( Z1 .LE. 0 ) THEN
           IF ( Z2 .LE. 0 ) THEN
             ZL = Z3
           ELSE
             ZL = Z2
           END IF
      ELSE
           ZL = Z1
      END IF
      CALL NEWTON(AV,BV,P,Z1,Z2,Z3)
      ZV = Z3

      DO 4 I = 1,NC
      HLS(I) = A1(I)*(T-460.)+A2(I)/2*(T-460.)**2+A3(I)/3*(T-460.)**3+
     +A4(I)/4*(T-460.)**4+A5(I)/5*(T-460.)**5
      HLS1 =  HLS1+X(I)*HLS(I)
      HVS  =  HVS +Y(I)*HLS(I)
    4 CONTINUE

      HL = (ZL-1-1.5*AL**2/BL*ALOG(1+BL*P/ZL))*GC*T+HLS1
      HV = (ZV-1-1.5*AV**2/BV*ALOG(1+BV*P/ZV))*GC*T+HVS
      RETURN
      END

CC
CC      Calculate the root of Z by newton's technique
CC

      SUBROUTINE NEWTON(A,B,P,Z1,Z2,Z3)
```

```
      DIMENSION Z(3)

      ZO1 = -10
      L = 1
   10 IF ( L .GT. 8000 ) THEN
            WRITE(*,*) '   ****** NOT CONVERGE !! ******'
            STOP
      END IF
      FZ1 = ZO1**3-ZO1**2+B*P*(A**2/B-B*P-1)*ZO1-A**2/B*(B*P)**2
      DFZ1 = 3*ZO1**2-2*ZO1+B*P*(A**2/B-B*P-1)
      Z1 = ZO1-FZ1/DFZ1
      IF (ABS(ZO1-Z1) .LT. 1E-3) THEN
      GOTO 20
      END IF
      ZO1 = Z1
      L = L+1
      GOTO 10

   20 ZO2 = 0
      L1 = 1
   25 IF ( L1 .GT. 8000 ) THEN
            WRITE(*,*) '   ****** NOT CONVERGE !! ******'
            STOP
      END IF
      FZ2 = ZO2**3-ZO2**2+B*P*(A**2/B-B*P-1)*ZO2-A**2/B*(B*P)**2
      DFZ2 = 3*ZO2**2-2*ZO2+B*P*(A**2/B-B*P-1)
      Z2 = ZO2-FZ2/DFZ2
      IF (ABS(ZO2-Z2) .LT. 1E-3) THEN
      GOTO 30
      END IF
      ZO2 = Z2
      L1 = L1+1
      GOTO 25
```

```
30 ZO3 = 10
   L2 = 1
35 IF ( L2 .GT. 8000 ) THEN
        WRITE(*,*) '   ****** NOT CONVERGE !! ******'
        STOP
   END IF
   FZ3 = ZO3**3-ZO3**2+B*P*(A**2/B-B*P-1)*ZO3-A**2/B*(B*P)**2
   DFZ3 = 3*ZO3**2-2*ZO3+B*P*(A**2/B-B*P-1)
   Z3 = ZO3-FZ3/DFZ3
   IF (ABS(ZO3-Z3) .LT. 1E-3) THEN
   GOTO 40
   END IF
   ZO3 = Z3
   L2 = L2+1
   GOTO 35

   Z(1) = Z1
   Z(2) = Z2
   Z(3) = Z3

   jump = 3
50 jump = jump/2
   IF ( jump .NE. 0 ) THEN
      j2 = 3-jump
      DO 60 j = 1,j2
        DO 60 i = j,1,-jump
        j3 = i+jump
        IF ( Z(i) .GT. Z(j3)) THEN
             s = Z(i)
             Z(i) = Z(j3)
             Z(j3) = s
        END IF
60    CONTINUE
   GOTO 50
   END IF
```

```
      Z1 = Z(1)
      Z2 = Z(2)
      Z3 = Z(3)

   40 RETURN
      END

CC
CC      Calculate average density
CC

      SUBROUTINE MWDENSG(X,T,P,MWA,DENSA)

      COMMON NC,MW(20),DENS(20),C1(20),C2(20),C3(20),BPT(20),AVP(20),
     +BVP(20),tc(20),pc(20),zc(20),w(20),a1(20),a2(20),a3(20),a4(20),
     +a5(20)

      DIMENSION X(20),TR(20),MI(20),ALPHA(20),ACI(20),AI(20),BI(20)

      REAL MW,MWA,MI,GC
      DENSA=0.0
      MWA=0.

CC
CC      Define constant value
CC

      GC = 10.731
      KIJ = 0.001

      SUMAL  = 0.0
      SUMBL  = 0.0
      SUMATL = 0.0
```

```
      DO 1 J=1,NC
      TR(J) = T/TC(J)
      ACI(J) = (0.42748/(TR(J)**2.5*PC(J)))**0.5
      AI(J) = ACI(J)
      BI(J) = 0.0867/TR(J)/PC(J)
      SUMBL = SUMBL + X(J)*BI(J)
      SUMAL = SUMAL + X(J)*AI(J)
    1 CONTINUE


      AL = SUMAL
      BL = SUMBL

      CALL NEWTON(AL,BL,P,Z1,Z2,Z3)
      IF ( Z1 .LE. 0 ) THEN
            IF ( Z2 .LE. 0 ) THEN
              ZL = Z3
            ELSE
              ZL = Z2
            END IF
      ELSE
            ZL = Z1
      END IF

      VV = ZL*GC*T/P

      DO 5 J=1,NC
    5 MWA=X(J)*MW(J)+MWA
      DENSA=1/(VV/MWA)

      RETURN
      END

CC
CC       Buble point Calculation
```

```
CC

      SUBROUTINE BUBPTGR(T,X,Y,P)

      REAL GC,MI,KI,KIH,FSLOPE,F,y

      COMMON NC,MW(20),DENS(20),C1(20),C2(20),C3(20),BPT(20),AVP(20),
     +BVP(20),tc(20),pc(20),zc(20),w(20),a1(20),a2(20),a3(20),a4(20),
     +a5(20)

      DIMENSION X(20),Y(20),TR(20),MI(20),ALPHA(20),ACI(20),
     +AI(20),BI(20),BIH(20),FIL(20),FIV(20),KI(20)

      DIMENSION FILH(20),FIVH(20),APRIMEXH(20),APRIMEYH(20),KIH(20),
     +ALPHAH(20),AIH(20),TRH(20)

CC
CC       Define constant value
CC

      GC = 1.986
      KIJ = 0.001

      DO 40 I = 1,NC
      IF (Y(I) .EQ. 0.) THEN
      ki(i) = 1.
      Y(I) = KI(I)*X(I)
      END IF
   40 continue


     LOOP = 0
   10 LOOP=LOOP+1

     SUMXA  = 0.0
```

```
      SUMYA  = 0.0
      SUMXAH = 0.0
      SUMYAH = 0.0
      SUMAL  = 0.0
      SUMAV  = 0.0
      SUMALH = 0.0
      SUMAVH = 0.0
      SUMBL  = 0.0
      SUMBV  = 0.0
      SUMBLH = 0.0
      SUMBVH = 0.0

      TH = T+1.0

      IF(LOOP .GT. 8000).GOTO 30
      SUMY=0.0
      SUMYH = 0.0

      DO 1 J=1,NC
      TR(J) = T/TC(J)
      TRH(J) = TH/TC(J)
      ACI(J) = (0.4278/(TR(J)**2.5*PC(J)))**0.5
      AI(J) = ACI(J)
      AIH(J) = (0.4278/(TRH(J)**2.5*PC(J)))**0.5
      BI(J) = 0.0867/TR(J)/PC(J)
      BIH(J) = 0.0867/TRH(J)/PC(J)
      SUMBL = SUMBL + X(J)*BI(J)
      SUMBV = SUMBV + Y(J)*BI(J)
      SUMBLH = SUMBLH + X(J)*BIH(J)
      SUMBVH = SUMBVH + Y(J)*BIH(J)
      SUMAL = SUMAL + X(J)*AI(J)
      SUMAV = SUMAV + Y(J)*AI(J)
      SUMALH = SUMALH + X(J)*AIH(J)
      SUMAVH = SUMAVH + Y(J)*AIH(J)
    1 CONTINUE
```

```
AL = SUMAL
ALH = SUMALH
AV = SUMAV
AVH = SUMAVH
BL = SUMBL
BLH = SUMBLH
BV  = SUMBV
BVH = SUMBLH

CALL NEWTON(AL,BL,P,Z1,Z2,Z3)
IF ( Z1 .LE. 0 ) THEN
      IF ( Z2 .LE. 0 ) THEN
        ZL = Z3
      ELSE
        ZL = Z2
      END IF
ELSE
      ZL = Z1
END IF
CALL NEWTON(ALH,BLH,P,Z1,Z2,Z3)
IF ( Z1 .LE. 0 ) THEN
      IF ( Z2 .LE. 0 ) THEN
        ZLH = Z3
      ELSE
        ZLH = Z2
      END IF
ELSE
      ZLH = Z1
END IF
CALL NEWTON(AV,BV,P,Z1,Z2,Z3)
ZV = Z3
CALL NEWTON(AVH,BVH,P,Z1,Z2,Z3)
ZVH = Z3
```

```
      DO 15 I=1,NC

      FIL(I) = EXP((ZL-1)*BI(I)/BL-ALOG(ZL-BL*P)-AL**2/BL*
     +(2*AI(I)/AL-BI(I)/BL)*ALOG(1+BL*P/ZL))
      FILH(I) = EXP((ZLH-1)*BIH(I)/BLH-ALOG(ZLH-BLH*P)-ALH**2/BLH*
     +(2*AIH(I)/ALH-BIH(I)/BLH)*ALOG(1+BLH*P/ZLH))
      FIV(I) = EXP((ZV-1)*BI(I)/BV-ALOG(ZV-BV*P)-AV**2/BV*
     +(2*AI(I)/AV-BI(I)/BV)*ALOG(1+BV*P/ZV))
      FIVH(I) = EXP((ZVH-1)*BIH(I)/BVH-ALOG(ZVH-BVH*P)-AVH**2/BVH*
     +(2*AIH(I)/AVH-BIH(I)/BVH)*ALOG(1+BVH*P/ZVH))

      KI(I) = FIL(I)/FIV(I)
      KIH(I) = FILH(I)/FIVH(I)
      SUMY = SUMY +X(I)*KI(I)
      SUMYH = SUMYH +X(I)*KIH(I)
   15 CONTINUE

      IF(LOOP .EQ. 1) THEN
      DO 16 I = 1,NC
      Y(I) = KI(I)*X(I)/SUMY
   16 CONTINUE
      GOTO 10
      END IF

      IF (ABS(SUMY-SUMYZ) .GE. 1E-3) THEN

      DO 17 I = 1,NC
      Y(I) = KI(I)*X(I)/SUMY

   17 CONTINUE
      SUMYZ = SUMY
      GOTO 10
      END IF
```

```
      IF ( ABS(SUMY-1.) .LT. 1E-3) RETURN
      F=SUMY-1.0
      FH = SUMYH-1.0
      FSLOPE=(FH-F)/1.
      T=T-F/FSLOPE
      GOTO 10

 1000 WRITE(*,*) '   ********** Converged !!'
      read(*,*)
      WRITE(*,*) '          TEMP          SUMY          LOOP '
      WRITE(*,*)
      WRITE(*,*) T, SUMY, LOOP
      READ(*,*)
      WRITE(*,*)
      WRITE(*,*) '      yi              ki'
      write(*,*)
      DO 1010 I = 1,NC
      WRITE(*,*) Y(I),KI(I)
      READ(*,*)
 1010 CONTINUE
      stop

   30 WRITE(*,*) 'The calculation does not converge !!'
      STOP
      END
```

```
CC      RUN PR MODEL

CC
CC      key = 8.3
CC


CC
CC      Assign temperature in Degree Rangin
CC


 9800 TFL = TFL + 460.
      TFV = TFV + 460.
      TB  = TB    + 460.
      TD  = TD    + 460.+115.0
      DO 18 N = 1,NT
      T(N) = T(N) + 460.
   18 CONTINUE

CC
CC    CALCULATE PRESSURE PROFILE
CC


      CALL drawlines()
      DO 35 N=1,NT
   35 P(N)=(PB-(N*(PB-PD))/NT)
      DELTA = 0.0001
      WRITE(*,37) DELTA
   37 FORMAT(1X,' DELTA = ', F8.5)
      TIME = 0.
      TPRINT = 0.
      CALL ENTHPR(TFL,P(NF),XF,YF,HLF,HVF)


CC
CC    CALL INTITIAL HOLDUPS
CC
```

```
      CALL MWDENSPR(XB,TB,PB,MWA,DENSA)

      MB=MVB*DENSA/MWA

      DO 20 N=1,NF
      DO 21 J=1,NC
   21 XX(J) = X(N,J)
      CALL MWDENSPR(XX,T(N),P(N),MWA,DENSA)
      LV(N) = LO(N)*MWA/DENSA
      L(N) = LO(N)
      HFOW = (LV(N)/(999.*WLS))**.66667
      MV(N) = (HFOW+WHS/12.)*3.1416*DS*DS/(4.*144.)
      M(N) = MV(N)*DENSA/MWA

   20 CONTINUE
      DO 25 N = NF+1,NT
      DO 26 J = 1,NC
   26 XX(J) = X(N,J)
      CALL MWDENSPR(XX,T(N),P(N),MWA,DENSA)
      LV(N) = LO(N)*MWA/DENSA
      L(N) = LO(N)
      HFOW = (LV(N)/(999.*WLR))**.66667
      MV(N) = (HFOW+WHR/12.)*3.1416*DR*DR/(4.*144.)
      M(N) = MV(N)*DENSA/MWA

   25 CONTINUE
      DO 30 N=1,NT
      DO 31 J=1,NC
      XM(N,J)=M(N)*X(N,J)
   31 CONTINUE
   30 CONTINUE
      CALL MWDENSPR(XD,TD,PD,MWA,DENSA)
      MD=MVD*DENSA/MWA
```

```
CC
CC     Initial Guess of V(5) for first efficiency calculation
CC

      V(NF) = 822.

CC
CC     Main Loop for each time step
CC
      xxx=0

  100 CONTINUE

      write(*,*) xxx
      CALL BUBPTPR (TB,XB,YB,PB)
      CALL ENTHPR (TB,PB,XB,YB,HLB,HVB)
      DO 105 J=1,NC

  105 XX(J)=X(1,J)
      CALL BUBPTPR (T(1),XX,YY,P(1))
      DO 106 J=1,NC

      Y(1,J)=YB(J)+EFF*(YY(J)-YB(J))
  106 YY(J)=Y(1,J)
      CALL ENTHPR(T(1),P(1),XX,YY,HL(1),HV(1))

      DO 110 N=2,NF
      DO 111 J=1,NC
  111 XX(J)=X(N,J)
      CALL BUBPTPR(T(N),XX,YY,P(N))
      DO 112 J = 1,NC

      Y(N,J)=(YY(J)-Y(N-1,J))*EFF+Y(N-1,J)
  112 YY(J)=Y(N,J)
      CALL ENTHPR(T(N),P(N),XX,YY,HL(N),HV(N))
```

```
 110 CONTINUE
     DO 113 J=1,NC
 113 XX(J)=X(NF+1,J)
     CALL BUBPTPR(T(NF+1),XX,YY,P(NF+1))
     DO 114 J=1,NC
     YAV(J)=(YF(J)*FV+Y(NF,J)*V(NF))/(V(NF)+FV)
     Y(NF+1,J)=(YY(J)-YAV(J))*EFF+YAV(J)
 114 YY(J)=Y(NF+1,J)
     CALL ENTHPR(T(NF+1),P(NF+1),XX,YY,HL(NF+1),HV(NF+1))
     DO 115 N=NF+2,NT
     DO 116 J=1,NC
 116 XX(J)=X(N,J)
     CALL BUBPTPR(T(N),XX,YY,P(N))
     DO 117 J=1,NC
     Y(N,J)=(YY(J)-Y(N-1,J))*EFF+Y(N-1,J)
 117 YY(J)=Y(N,J)
     CALL ENTHPR (T(N),P(N),XX,YY,HL(N),HV(N))
 115 CONTINUE
     CALL BUBPTPR (TD,XD,YD,PD)
     CALL ENTHPR(TD,PD,XD,YD,HLD,HVD)

CC
CC       CALCULATE VAPOR RATES
CC

     VB=(QR*1000.-L(1)*(HLB-HL(1)))/(HVB-HLB)
     B = L(1)-VB
CC       IF (B .LT. 0.) THEN
CC       WRITE(*,*) ' ******** B IS LESS THAN 0 ********'
CC       STOP
CC       END IF

     V(1)=(HL(2)*L(2)+HVB*VB-HL(1)*L(1))/HV(1)
     DO 120 N=2,NF-1
```

```
      V(N)=(HL(N+1)*L(N+1)+HV(N-1)*V(N-1)-HL(N)*L(N))/HV(N)
  120 CONTINUE
      V(NF)=(HL(NF+1)*L(NF+1)+HV(NF-1)*V(NF-1)-HL(NF)*L(NF)+HLF*FL)/HV
     +(NF)
      V(NF+1)=(HL(NF+2)*L(NF+2)+HV(NF)*V(NF)+HVF*FV-HL(NF+1)*L(NF+1))/
     +HV(NF+1)
      DO 130 N=NF+2,NT-1
  130 V(N)=(HL(N+1)*L(N+1)+HV(N-1)*V(N-1)-HL(N)*L(N))/HV(N)
      V(NT)=(HLD*R+HV(NT-1)*V(NT-1)-HL(NT)*L(NT))/HV(NT)
      DL=V(NT)-DV-R
CC        IF (DL .LT. 0) THEN
CC        WRITE(*,*) ' ******** DL IS LESS THAN 0 ********'
CC        STOP
CC        END IF

CC
CC        EVALUATE DERIVATIVES
CC

      DM(1)=L(2)+VB-V(1)-L(1)
      DO 140 N=2,NF-1
  140 DM(N)=L(N+1)+V(N-1)-L(N)-V(N)
      DM(NF)=L(NF+1)+FL+V(NF-1)-L(NF)-V(NF)
      DM(NF+1)=L(NF+2)+FV+V(NF)-L(NF+1)-V(NF+1)
      DO 150 N=NF+2,NT-1
  150 DM(N)=L(N+1)+V(N-1)-L(N)-V(N)
      DM(NT)=R+V(NT-1)-L(NT)-V(NT)
      DO 160 J=1,NC
      DXB(J)=(X(1,J)*L(1)-YB(J)*VB-XB(J)*B)/MB
      DXM(1,J)=X(2,J)*L(2)+YB(J)*VB-X(1,J)*L(1)-Y(1,J)*V(1)
      DO 165 N=2,NF-1
  165 DXM(N,J)=X(N+1,J)*L(N+1)+Y(N-1,J)*V(N-1)-X(N,J)*L(N)-V(N)*
     +Y(N,J)
      DXM(NF,J)=X(NF+1,J)*L(NF+1)+Y(NF-1,J)*V(NF-1)-X(NF,J)*L(NF)-
     +V(NF)*Y(NF,J)+FL*XF(J)
```

```
      DXM(NF+1,J)=X(NF+2,J)*L(NF+2)+Y(NF,J)*V(NF)-X(NF+1,J)*L(NF+1)
     +-V(NF+1)*Y(NF+1,J)+FV*YF(J)
      DO 170 N=NF+2,NT-1
  170 DXM(N,J)=X(N+1,J)*L(N+1)+Y(N-1,J)*V(N-1)-X(N,J)*L(N)-V(N)*Y(N,J)
      DXM(NT,J)=XD(J)*R+Y(NT-1,J)*V(NT-1)-X(NT,J)*L(NT)-Y(NT,J)*V(NT)
      DXD(J)=(V(NT)*Y(NT,J)-DV*YD(J)-(R+DL)*XD(J))/MD
  160 CONTINUE
      IF (TIME .GT. 4e-2) GOTO 9000
      IF (TIME .LT. TPRINT) GOTO 210

      CALL drawlines()
      WRITE(5,201)
  201 FORMAT (5X,'TIME     T        X1       X2       X3       X4
     +     X5         L')
      WRITE(5,202) TIME,TB,(XB(J),J=1,NC),B
  202 FORMAT (1X,F5.4,3X,F7.2,8F14.6,F10.1)
      DO 203 N=1,NT
  203 WRITE(5,204) N,T(N),(X(N,J),J=1,NC),L(N)
  204 FORMAT (3X,I3,3X,F7.2,8F14.6,F10.1)
      WRITE(5,205) TD,(XD(J),J=1,NC),R
  205 FORMAT (9X,F7.2,8F14.6,F10.1)
      WRITE(5,206) (YD(J),J=1,NC),DL
  206 FORMAT(16X,8F14.6,F10.1)
      TPRINT = TPRINT + .001
cc    CALL settextposition( 24, 5, curpos )
cc    CALL outtext( 'Press ENTER to continue ...' )
cc    READ(*,*)

CC
CC        INTEGRATION ALA EULER
CC

  210 TIME = TIME + DELTA
cc    WRITE(*,*) TIME
cc    READ(*,*)
```

```
      DO 215 N=1,NT
  215 M(N)=M(N)+DM(N)*DELTA
      DO 220 J = 1,NC
      XB(J)=XB(J)+DXB(J)*DELTA
          IF (XB(J) .LT. 0.) XB(J) = 0.0
          IF (XB(J) .GT. 1.) XB(J) = 1.
      DO 225 N=1,NT
      XM(N,J)=XM(N,J)+DXM(N,J)*DELTA
      X(N,J)=XM(N,J)/M(N)
          IF (X(N,J) .GT. 1.) X(N,J) = 1.
          IF (X(N,J) .LT. 0.) X(N,J) = 0.0
  225 CONTINUE
      XD(J)=XD(J)+DXD(J)*DELTA
          IF (XD(J) .LT. 0.) XD(J)=0.
          IF (XD(J) .GT. 1.) XD(J)=1.
  220 CONTINUE

CC
CC    CALCULATE NEW LIQUID RATES
CC

      DO 270 N=1,NF
      DO 271 J=1,NC
      XX(J)=X(N,J)
  271 CONTINUE
      CALL HYDRAU(P(N),M(N),T(N),XX,L(N),WHS,WLS,DS)
  270 CONTINUE
      DO 273 N=NF+1,NT
      DO 275 J=1,NC
  275 XX(J)=X(N,J)
      CALL HYDRAU(P(N),M(N),T(N),XX,L(N),WHR,WLR,DR)
  273 CONTINUE
      xxx=xxx+1.0
      GOTO 100
```

```
400 STOP
    END

CC
CC      Calculate Liquid tray holdup by Francis Wier equation
CC

    SUBROUTINE  HYDRAU(P,M,T,X,L,WH,WL,DCOL)

    REAL M,L,MW,MWA

    COMMON NC,MW(20),DENS(20),C1(20),C2(20),C3(20),BPT(20),AVP(20),
   +BVP(20),tc(20),pc(20),zc(20),w(20),a1(20),a2(20),a3(20),a4(20),
   +a5(20)

    DIMENSION X(20)

    CALL MWDENSPR(X,T,P,MWA,DENSA)
    CONST=183.2*M*MWA/(DENSA*DCOL*DCOL)-WH/12.
    IF (CONST .LE. 0.) GOTO 10
    L=DENSA*WL*999.*((183.2*M*MWA/(DENSA*DCOL*DCOL)-WH/12.)**1.5)/MWA
    RETURN
 10 L=0.
    RETURN
    END

CC
CC      Enthalpy calculation
CC

    SUBROUTINE ENTHPR(T,P,X,Y,HL,HV)

    REAL GC,MI

    COMMON NC,MW(20),DENS(20),C1(20),C2(20),C3(20),BPT(20),AVP(20),
```

```
     +BVP(20),tc(20),pc(20),zc(20),w(20),a1(20),a2(20),a3(20),a4(20),
     +a5(20)

      DIMENSION X(20),Y(20),TR(20),MI(20),ALPHA(20),ACI(20),AI(20),
     +BI(20),HLS(20)

CC
CC       Define constant value
CC

      GC = 1.986
      KIJ = 0.001

      HL = 0.0
      HV = 0.0
      HLS1 = 0.0
      HVS  = 0.0
      SUMAL  = 0.0
      SUMAV  = 0.0
      SUMBL  = 0.0
      SUMBV  = 0.0
      SUMATL = 0.0
      SUMATV = 0.0
      SUMPL  = 0.0
      SUMPV  = 0.0

      DO 1 J=1,NC
      TR(J) = T/TC(J)
      MI(J) = 0.37464+1.54226*W(J)-0.26992*W(J)**2
      ALPHA(J) = (1.0+MI(J)*(1-TR(J)**(0.5)))**2
      ACI(J) = 0.45724*(GC*TC(J))**2/PC(J)
      AI(J) = ALPHA(J)*ACI(J)
      BI(J) = 0.07780*GC*TC(J)/PC(J)
      SUMBL = SUMBL + X(J)*BI(J)
      SUMBV = SUMBV + Y(J)*BI(J)
```

```
1 CONTINUE

 DO 2 I = 1,NC
 DO 3 J = 1,NC
 SUMAL = SUMAL+X(I)*X(J)*(AI(I)*AI(J))**0.5*(1-KIJ)
 SUMAV = SUMAV+Y(I)*Y(J)*(AI(I)*AI(J))**0.5*(1-KIJ)
 SUMATL = SUMATL+X(I)*X(J)*MI(J)*(AI(I)*ACI(J)*TR(J))**0.5*(1-KIJ)
 SUMATV = SUMATV+Y(I)*Y(J)*MI(J)*(AI(I)*ACI(J)*TR(J))**0.5*(1-KIJ)
 SUMPL = SUMPL+X(I)*X(J)*MI(J)*(ACI(I)*ACI(J)*TR(J))**0.5*(1-KIJ)
 SUMPV = SUMPV+Y(I)*Y(J)*MI(J)*(ACI(I)*ACI(J)*TR(J))**0.5*(1-KIJ)
3 CONTINUE
2 CONTINUE

 AL = SUMAL*P/(GC*T)**2
 AV = SUMAV*P/(GC*T)**2
 BL = SUMBL*P/(GC*T)
 BV = SUMBV*P/(GC*T)
 CALL NEWTON(AL,BL,Z1,Z2,Z3)
 IF ( Z1 .LE. 0 ) THEN
       IF ( Z2 .LE. 0 ) THEN
         ZL = Z3
       ELSE
         ZL = Z2
       END IF
 ELSE
       ZL = Z1
 END IF
 CALL NEWTON(AV,BV,Z1,Z2,Z3)
 ZV = Z3

 DO 4 I = 1,NC
 HLS(I) = A1(I)*(T-460.)+A2(I)/2*(T-460.)**2+A3(I)/3*(T-460.)**3+
+A4(I)/4*(T-460.)**4+A5(I)/5*(T-460.)**5
 HLS1 =  HLS1+X(I)*HLS(I)
 HVS  =  HVS +Y(I)*HLS(I)
```

```
 4 CONTINUE

  HL = (ZL-1-AL/(2.8284*BL)*(1-SUMPL/SUMAL)*ALOG((ZL+2.4142*BL)/
 +(ZL-0.4142*BL)))*GC*T+HLS1
  HV = (ZV-1-AV/(2.8284*BV)*(1-SUMPV/SUMAV)*ALOG((ZV+2.4142*BV)/
 +(ZV-0.4142*BV)))*GC*T+HVS

  RETURN
  END

CC
CC      Calculate the root of Z by newton's technique
CC

  SUBROUTINE NEWTON(A,B,Z1,Z2,Z3)

  DIMENSION Z(3)

  ZO1 = -10
  L = 1
10 IF ( L .GT. 8000 ) THEN
      WRITE(*,*) '   ****** NOT CONVERGE !! ******'
      STOP
  END IF
  FZ1 = ZO1**3-ZO1**2+(A-B-B**2)*ZO1-A*B
  DFZ1 = 3*ZO1**2-2*ZO1+(A-B-B**2)
  Z1 = ZO1-FZ1/DFZ1
  IF (ABS(ZO1-Z1) .LT. 1E-3) THEN
  GOTO 20
  END IF
  ZO1 = Z1
  L = L+1
  GOTO 10

20 ZO2 = 0
```

```
      L1 = 1
   25 IF ( L1 .GT. 8000 ) THEN
            WRITE(*,*) '   ****** NOT CONVERGE !! ******'
            STOP
      END IF
      FZ2 = ZO2**3-ZO2**2+(A-B-B**2)*ZO2-A*B
      DFZ2 = 3*ZO2**2-2*ZO2+(A-B-B**2)
      Z2 = ZO2-FZ2/DFZ2
      IF (ABS(ZO2-Z2) .LT. 1E-3) THEN
      GOTO 30
      END IF
      ZO2 = Z2
      L1 = L1+1
      GOTO 25

   30 ZO3 = 10
      L2 = 1
   35 IF ( L2 .GT. 8000 ) THEN
            WRITE(*,*) '   ****** NOT CONVERGE !! ******'
            STOP
      END IF
      FZ3 = ZO3**3-ZO3**2+(A-B-B**2)*ZO3-A*B
      DFZ3 = 3*ZO3**2-2*ZO3+(A-B-B**2)
      Z3 = ZO3-FZ3/DFZ3
      IF (ABS(ZO3-Z3) .LT. 1E-3) THEN
      GOTO 40
      END IF
      ZO3 = Z3
      L2 = L2+1
      GOTO 35

      Z(1) = Z1
      Z(2) = Z2
      Z(3) = Z3
```

```
      jump = 3
   50 jump = jump/2
      IF ( jump .NE. 0 ) THEN
          j2 = 3-jump
          DO 60 j = 1,j2
            DO 60 i = j,1,-jump
            j3 = i+jump
            IF ( Z(i) .GT. Z(j3)) THEN
                  s = Z(i)
                  Z(i) = Z(j3)
                  Z(j3) = s
            END IF
   60     CONTINUE
      GOTO 50
      END IF

      Z1 = Z(1)
      Z2 = Z(2)
      Z3 = Z(3)

   40 RETURN
      END

CC
CC       Calculate average density
CC

      SUBROUTINE MWDENSPR(X,T,P,MWA,DENSA)

      COMMON NC,MW(20),DENS(20),C1(20),C2(20),C3(20),BPT(20),AVP(20),
     +BVP(20),tc(20),pc(20),zc(20),w(20),a1(20),a2(20),a3(20),a4(20),
     +a5(20)

      DIMENSION X(20),TR(20),MI(20),ALPHA(20),ACI(20),AI(20),BI(20)
```

```
      REAL MW,MWA,MI,GC
      DENSA=0.0
      MWA=0.

CC
CC      Define constant value
CC

      GC = 10.731
      KIJ = 0.001

      SUMAL  = 0.0
      SUMBL  = 0.0
      SUMATL = 0.0

      DO 1 J=1,NC
      TR(J) = T/TC(J)
      MI(J) = 0.37464+1.54226*W(J)-0.26992*W(J)**2
      ALPHA(J) = (1.0+MI(J)*(1-TR(J)**(0.5)))**2
      ACI(J) = 0.45724*(GC*TC(J))**2/PC(J)
      AI(J) = ALPHA(J)*ACI(J)
      BI(J) = 0.07780*GC*TC(J)/PC(J)
      SUMBL = SUMBL + X(J)*BI(J)
    1 CONTINUE

      DO 2 I = 1,NC
      DO 3 J = 1,NC
      SUMAL = SUMAL+X(I)*X(J)*(AI(I)*AI(J))**0.5*(1-KIJ)
      SUMATL = SUMATL+X(I)*X(J)*MI(J)*(AI(I)*ACI(J)*TR(J))**0.5*(1-KIJ)
    3 CONTINUE
    2 CONTINUE

      AL = SUMAL*P/(GC*T)**2
      BL = SUMBL*P/(GC*T)
```

```
      CALL NEWTON(AL,BL,Z1,Z2,Z3)
      IF ( Z1 .LE. 0 ) THEN
            IF ( Z2 .LE. 0 ) THEN
              ZL = Z3
            ELSE
              ZL = Z2
            END IF
      ELSE
            ZL = Z1
      END IF

      VV = ZL*GC*T/P

      DO 5 J=1,NC
    5 MWA=X(J)*MW(J)+MWA
      DENSA=1/(VV/MWA)

      RETURN
      END

CC
CC       Buble point Calculation
CC

      SUBROUTINE BUBPTPR(T,X,Y,P)

      REAL GC,MI,KI, APRIMEX,APRIMEY,APRIMEXH,APRIMEYH,KIH,
     +FSLOPE,F,y

      COMMON NC,MW(20),DENS(20),C1(20),C2(20),C3(20),BPT(20),AVP(20),
     +BVP(20),tc(20),pc(20),zc(20),w(20),a1(20),a2(20),a3(20),a4(20),
     +a5(20)

      DIMENSION X(20),Y(20),TR(20),MI(20),ALPHA(20),ACI(20),
     +AI(20),BI(20),FIL(20),FIV(20),KI(20),APRIMEX(20),APRIMEY(20)
```

```
      DIMENSION FILH(20),FIVH(20),APRIMEXH(20),APRIMEYH(20),KIH(20),
     +ALPHAH(20),AIH(20),TRH(20)

CC
CC       Define constant value
CC

      GC = 1.986
      KIJ = 0.001

      DO 40 I = 1,NC
      IF (Y(I) .EQ. 0.) THEN
      ki(i) = 1.
      Y(I) = KI(I)*X(I)
      END IF
   40 continue

      LOOP = 0
   10 LOOP=LOOP+1

      SUMXA  = 0.0
      SUMYA  = 0.0
      SUMXAH = 0.0
      SUMYAH = 0.0
      SUMAL  = 0.0
      SUMAV  = 0.0
      SUMALH = 0.0
      SUMAVH = 0.0
      SUMBL  = 0.0
      SUMBV  = 0.0

      TH = T+1.0
```

```
      IF(LOOP .GT. 8000) GOTO 30
      SUMY=0.0
      SUMYH = 0.0

      DO 1 J=1,NC
      TR(J) = T/TC(J)
      TRH(J) = TH/TC(J)
      MI(J) = 0.37464+1.54226*W(J)-0.26992*W(J)**2
      ALPHA(J) = (1.0+MI(J)*(1-TR(J)**(0.5)))**2
      ALPHAH(J) = (1.0+MI(J)*(1-TRH(J)**(0.5)))**2
      ACI(J) = 0.45724*(GC*TC(J))**2/PC(J)
      AI(J) = ALPHA(J)*ACI(J)
      AIH(J) = ALPHAH(J)*ACI(J)
      BI(J) = 0.07780*GC*TC(J)/PC(J)
      SUMXA = SUMXA+X(J)*AI(J)**0.5*(1-KIJ)
      SUMXAH = SUMXAH+X(J)*AIH(J)**0.5*(1-KIJ)
      SUMYA = SUMYA+Y(J)*AI(J)**0.5*(1-KIJ)
      SUMYAH = SUMYAH+Y(J)*AIH(J)**0.5*(1-KIJ)
      SUMBL = SUMBL + X(J)*BI(J)
      SUMBV = SUMBV + Y(J)*BI(J)
    1 CONTINUE

      DO 2 I = 1,NC
      DO 3 J = 1,NC
      SUMAL = SUMAL+X(I)*X(J)*(AI(I)*AI(J))**0.5*(1-KIJ)
      SUMALH = SUMALH + X(I)*X(J)*(AIH(I)*AIH(J))**0.5*(1-KIJ)
      SUMAV = SUMAV+Y(I)*Y(J)*(AI(I)*AI(J))**0.5*(1-KIJ)
      SUMAVH = SUMAVH + Y(I)*Y(J)*(AIH(I)*AIH(J))**0.5*(1-KIJ)
    3 CONTINUE
    2 CONTINUE

      AL = SUMAL*P/(GC*T)**2
      ALH = SUMALH*P/(GC*TH)**2
      AV = SUMAV*P/(GC*T)**2
      AVH = SUMAVH*P/(GC*TH)**2
```

```
BL = SUMBL*P/(GC*T)
BLH = SUMBL*P/(GC*TH)
BV  = SUMBV*P/(GC*T)
BVH = SUMBL*P/(GC*TH)

CALL NEWTON(AL,BL,Z1,Z2,Z3)
IF ( Z1 .LE. 0 ) THEN
      IF ( Z2 .LE. 0 ) THEN
        ZL = Z3
      ELSE
        ZL = Z2
      END IF
ELSE
      ZL = Z1
END IF
CALL NEWTON(ALH,BLH,Z1,Z2,Z3)
IF ( Z1 .LE. 0 ) THEN
      IF ( Z2 .LE. 0 ) THEN
        ZLH = Z3
      ELSE
        ZLH = Z2
      END IF
ELSE
      ZLH = Z1
END IF
CALL NEWTON(AV,BV,Z1,Z2,Z3)
ZV = Z3
CALL NEWTON(AVH,BVH,Z1,Z2,Z3)
ZVH = Z3

DO 15 I=1,NC
APRIMEX(I) = 1.0/SUMAL*(2*AI(I)**0.5*SUMXA)
APRIMEXH(I) = 1.0/SUMALH*(2*AIH(I)**0.5*SUMXAH)
APRIMEY(I) = 1.0/SUMAV*(2*AI(I)**0.5*SUMYA)
APRIMEYH(I) = 1.0/SUMAVH*(2*AIH(I)**0.5*SUMYAH)
```

```
      FIL(I) = EXP(BI(I)/SUMBL*(ZL-1)-ALOG(ZL-BL)-AL/(2.82843*BL)*
     +(APRIMEX(I)-BI(I)/SUMBL)*ALOG((ZL+2.4142*BL)/(ZL-0.4142*BL)))
      FILH(I) = EXP(BI(I)/SUMBL*(ZLH-1)-ALOG(ZLH-BLH)-ALH/(2.82843*BLH)*
     +(APRIMEXH(I)-BI(I)/SUMBL)*ALOG((ZLH+2.4142*BLH)/(ZLH-0.4142*BLH)))
      FIV(I) = EXP(BI(I)/SUMBV*(ZV-1)-ALOG(ZV-BV)-AV/(2.82843*BV)*
     +(APRIMEY(I)-BI(I)/SUMBV)*ALOG((ZV+2.4142*BV)/(ZV-0.4142*BV)))
      FIVH(I) = EXP(BI(I)/SUMBV*(ZVH-1)-ALOG(ZVH-BVH)-AVH/(2.82843*BVH)*
     +(APRIMEYH(I)-BI(I)/SUMBV)*ALOG((ZVH+2.4142*BVH)/(ZVH-0.4142*BVH)))
      KI(I) = FIL(I)/FIV(I)
      KIH(I) = FILH(I)/FIVH(I)
      SUMY = SUMY +X(I)*KI(I)
      SUMYH = SUMYH +X(I)*KIH(I)
   15 CONTINUE

      IF(LOOP .EQ. 1) THEN
      DO 16 I = 1,NC
      Y(I) = KI(I)*X(I)/SUMY
   16 CONTINUE
      GOTO 10
      END IF

      IF (ABS(SUMY-SUMYZ) .GE. 1E-3) THEN

      DO 17 I = 1,NC
      Y(I) = KI(I)*X(I)/SUMY

   17 CONTINUE
      SUMYZ = SUMY
      GOTO 10
      END IF

      IF ( ABS(SUMY-1.) .LT. 1E-3) RETURN
      F=SUMY-1.0
      FH = SUMYH-1.0
      FSLOPE=(FH-F)/1.
```

```
      T=T-F/FSLOPE
      GOTO 10

1000 WRITE(*,*) '   ********** Converged !!'
      read(*,*)
      WRITE(*,*) '          TEMP          SUMY          LOOP '
      WRITE(*,*)
      WRITE(*,*) T, SUMY, LOOP
      READ(*,*)
      WRITE(*,*)
      WRITE(*,*) '      yi              ki'
      write(*,*)
      DO 1010 I = 1,NC
      WRITE(*,*) Y(I),KI(I)
      READ(*,*)
1010 CONTINUE
      stop

 30 WRITE(*,*) 'The calculation does not converge !!'
      STOP
      END
```

VITA

Miss Kallaya Klaithong graduated high school from Benchama Ratrungsarit school in 1986 and recieved Bachelor Degree in Chemical Engineering from the Department of Chemical Technology, Faculty of Science, Chulalongkorn Universityin in 1990. After then she subsequently studied for a requirement of the Master's Degree in Chemical Engineering at the Department of Chemical Engineering, Faculty of Engineering, Chulalongkorn University from 1992 till 1994.

She also has an experience in working as process engineer at The Bangchak Petroleum Public Company Limited in 1990 uptill the present.