9. Computer Programmes

Appendix 1.

```
C
C-----------------MIGR1-- .MAIN----------------------------------------
C
C   THIS PROGRAM CALCULATED THE REDISTRIBUTION OF A SINGLE MIGRANT
C   DUE TO CONVECTIVE TRANSFER, HYDRODYNAMICAL DISPERSION AND KINETICS
C   OF SORPTION WITH LINEAR ISOTHERM. VALUES THAT ARE TO BE INPUT ARE:
C   A   : INITIAL SOLUTE CONCENTRATION IN COLUMN;
C   C0  : CONCENTRATION OF THE SOLUTION INTRODUCED AT THE TOP;
C   ALT : COLUMN LENGTH;
C   N   : NUMBER OF INTERVALS IN SIMPSON QUADRATURE;
C   T0  : INITIAL MOMENT OF TIME;
C   DT  : TIME PERIOD OF CALCULATED DATA;
C   X   : A DEPTH FOR WHICH CALCULATIONS ARE TO BE MADE;
C   BET : "INTERNAL" FLOW RATE;
C   GAM : A RATIO OF CAPACITIES OF THE TWO EXCHANGING PHASES;
C   LAM : THE RATE OF DESORPTION;
C   MY  : THE COEFFICIENT IN FREUNDLICH ISOTHERM;
C   B.C (MB) - BOUNDARY CONDITION AT THE TOP OF THE COLUMN :
C         MB=1 - CONCENTRATION AT THE TOP IS HELD CONSTANT,
C        ,MB=3 - THE SAME FOR FLOW INPUT;
C   D   : THE DISPERSION COEFFICIENT.
C   ALL THESE VARIABLES ARE TO BE INPUT IN FREE FORMAT. RESULTING VALUES
C   (CONCENTRATION OF FILTRATE IN SEQUENTIAL MOMENTS OF TIME; IF
C   DESIRED, SORBED CONCENTRATION VALUES CAN BE APPENDED AS WELL)
C   ARE DIRECTED TO DISK FILE, NAMED "MY.LC", IN PRINTABLE FORMAT.
C   ANY OTHER PERIPHERAL MAY BE USED INSTEAD.
C
C   SUPPORTING ROUTINES : SIMPSON, PARAB, SUBINT, BESREG, WERF, ERFC.
C
        PARAMETER M=40
        COMMON  /ALPHA/ BET,MB,AKAP,AKAP1,AKAP2
     *          /DELTA/ GAM1,ALAM,X,T,ALT
        DIMENSION Y(0:M),T1(0:M)
10      FORMAT (1X//T10,'D=',E12.6,T25,'BET=',E10.4,T42,
     *    'GAM=',F7.4,T55,'LAM=',F6.5/T10,'MY=',F8.5,
     *    T25,'A=',F7.4,T42,'ALT=',F8.2,T55,'X=',F8.4//
     *    T8,'TIME',T24,'FILTRAT'//)
20      FORMAT (1X,T6,F8.5,T20,F8.5)
        OPEN 1,'MY.LC',ATT='A'
1       ACCEPT 'A=',A,'C0=',C0,'ALT=',ALT,
     *      'N=',N,'T0=',T0,'DT=',DT,'X=',X,
     *      'BET=',BET,'GAM=',GAM,'MY=',AMY,
     *      'LAM=',ALAM,'B.C. (1 OR 3) ? ',MB,'D=',D
        DO 4 I=0,M
4       T1(I)=T0+I*DT
        AMY1=A*AMY
        AKAP=BET/D
        AKAP1=.5/SQRT(D)
        AKAP2=8.*BET*AKAP1
        GAM1=GAM*ALAM*AMY
        DO 6 I=0,M
        T=T1(I)
        ALAM2=2.*GAM1*ALAM*T
        CALL SIMPSON (N,U1,U2,U3)
        Y(I)=C0+(A-C0)*(EXP(-GAM1*T)*(PARAB(X,T,0.)-
     *        PARAB(X,T,ALT))+T*(ALAM2*U3+GAM1*U1))
6       CONTINUE
        WRITE (1,10) D,BET,GAM,ALAM,AMY,A,ALT,X
        WRITE (1,20) (T1(I),Y(I),I=0,M)
        CLOSE (1)
        STOP
        END
```

```
C
C------------------SUBROUTINE--SIMPSON-------------------------------
C
C    THIS SUBROUTINE REALIZES THE NUMERICAL INTEGRATION WITH
C    SIMPSON FORMULA. NUMBER OF SUBINTERVALS : N
C    INTERVAL : [0,1]
C    INTEGRATED FUNCTIONS Z1,Z2,Z3 ARE SUPPLIED WITH SUBROUTINE
C    SUBINT, WHILE RESULTING INTEGRALS U1,U2,U3 ARE
C    OUTPUT ARGUMENTS
C    CALLING SEQUENCE : CALL SIMPSON (N,U1,U2,U3)
C
C--------------------------------------------------------------------
      SUBROUTINE SIMPSON (N,U1,U2,U3)
      C=6*N
      Y1=0.
      Y2=0.
      Y3=0.
      W1=0.
      W2=0.
      W3=0.
      DO 1 I=1,N
      CALL SUBINT ((I-0.5)/N,Z1,Z2,Z3)
      Y1=Y1+Z1
      Y2=Y2+Z2
1     Y3=Y3+Z3
      DO 2 I=1,N-1
      CALL SUBINT (I/FLOAT(N),Z1,Z2,Z3)
      W1=W1+Z1
      W2=W2+Z2
2     W3=W3+Z3
      CALL SUBINT (0.,U1,U2,U3)
      CALL SUBINT (1.,Z1,Z2,Z3)
      U1=(U1+Z1+2.*W1+4.*Y1)/C
      U2=(U2+Z2+2.*W2+4.*Y2)/C
      U3=(U3+Z3+2.*W3+4.*Y3)/C
      RETURN
      END
C
C------------------SUBROUTINE--SUBINT--------------------------------
C
C    THIS SUBROUTINE PROVIDES THREE FUNCTIONS Y1,Y2 AND Y3 THAT
C    FURTHER ARE TO BE INTEGRATED WITH SUBROUTINE SIMPSON. Z IS
C    AN ARGUMENT, SOME CONSTANTS AND VARIABLES ARE SUPPLIED THROUGH
C    COMMON FIELD /DELTA/. FURTHER DEFINITIONS AND DISCUSSIONS
C    SEE IN TEXT.
C    SUPPORTING ROUTINES BESIDES STANDARD FORTRAN LIBRARY :
C      PARAB
C      BESREG
C    CALLING SEQUENCE : CALL SUBINT (Z,Y1,Y2,Y3)
C
C--------------------------------------------------------------------
      SUBROUTINE SUBINT (Z,Y1,Y2,Y3)
      COMMON /DELTA/ GAM1,ALAM,X,T,ALT
      IF (T.LE.0.) GO TO 1
      Y1=SQRT(GAM1*Z)
      Y2=SQRT(ALAM*(1.-Z))
      Y3=Y1-Y2
      Y3=T*Y3*Y3
      IF (Y3.LT.20.) GO TO 2
1     Y1=0.
      Y2=0.
      Y3=0.
```

```
      RETURN
2     Y3=EXP(-Y3)*(PARAB(X,T*Z,0.)-PARAB(X,T*Z,ALT))
      Y=2.*T*Y1*Y2
      CALL BESREG (Y,Y1,Y2)
      Y1=Y1*Y3
      Y2=Y2*Y3
      Y3=Y2*Z
      RETURN
      END
C
C-----------------------FUNCTION-----PARAB----------------------------
C
C   THIS SUBROUTINE COMPUTES VALUES OF FUNCTION OF TWO ARGUMENTS,
C   X AND T. THIS FUNCTION PRESENTS A SOLUTION OF A PARABOLIC EQUA-
C   TION ON A SEMIINFINITE INTERVAL; BOUNDARY CONDITION AT X=0
C   MAY BE OF FIRST (MB=1), SECOND (MB=2) AND THIRD KIND (MB=3).
C   INITIAL CONDITION IS A JUMP FUNCTION EQUAL TO ZERO FOR X.LT.S
C   AND 1.0 FOR X.GE.S . T MUST BE NON-NEGATIVE.
C   THE SUBROUTINE ASSUMES THAT SOME CONSTANTS ARE BEING SUPPLIED
C   THROUGH COMMON FIELD /ALPHA/ FROM MAIN PROGRAM. MORE DETAILED DIS-
C   CUSSION SEE IN TEXT. .
C   SUPPORTING ROUTINES : ERFC AND WERF
C   CALLING SEQUENCE : Y = PARAB (X,T,S)
C
C--------------------------------------------------------------------
      FUNCTION PARAB (X,T,S)
      COMMON /ALPHA/ BET,MB,AKAP,AKAP1,AKAP2
      IF (T.GT.0) GO TO 1
      PARAB=1.
      IF (S.GT.X) PARAB=0.
      RETURN
1     A=SQRT(T)
      W=AKAP1/A
      W1=AKAP2*A
      A=BET*T+S
      XRET=(A-X)*W
      XADV=(A+X)*W
      Y=ERFC(XRET)
      A=AKAP*X-XADV*XADV
      IF (A.LT.-174.67) GO TO 5
      Y1=WERF(XADV)
      IF (MB-2) 2,4,3
2     Y1=-Y1
      GO TO 4
3     Y1=Y1+W1*(Y1*XADV-0.5641896)
4     Y=Y+Y1*EXP(A)
5     PARAB=.5*Y
      RETURN
      END
C
C---------------SUBROUTINE--BESREG-----------------------------------
C
C   THIS SUBROUTINE COMPUTES VALUES OF TWO FUNCTIONS Y1 AND Y2
C   FOR NON-NEGATIVE ARGUMENT X.
C   Y1 = I0(X) * EXP (-X),
C   Y2 = I1(X) * EXP (-X)/X,
C   I0 AND I1 BEING THE MODIFIED BESSEL FUNCTIONS OF FIRST KIND
C   WITH ORDERS 0 AND 1 CORRESPONDENTLY. RELATIVE ERROR IS
C   BELIEVED NOT TO BE GREATER THAN 2.0E-7
C   CALLING SEQUENCE : CALL BESREG (X,Y1,Y2)
C
C--------------------------------------------------------------------
```

```
      SUBROUTINE BESREG (X,Y1,Y2)
      DIMENSION CS(7),SN2(7),A(6),B(6)
      DATA CS,SN2,A,B
     * /.9945219,.5510565,.8660254,.7431448,
     *   .5877853,.4067366,.2079117,.0109262,
     *   .0954915,.250,.4477358,.6545085,.8345653,
     *   .9567727,.125,.0703125,.7324219E-1,.1121521,
     *   .2271080,.5725014,.375,.1171875,
     *   .1025391,.1441956,.2775764,.6765926/
      Y1=1.
      Y2=1.
      IF (X.GT.11.) GO TO 2
      DO 1 J=1,7
      R=2.*COSH(X*CS(J))
      Y1=Y1+R
1     Y2=Y2+SN2(J)*R
      R=EXP(-X)/15.
      GO TO 3
2     R=1./X
      Y1=1.727720
      Y2=1.993532
      DO 25 I=6,1,-1
      Y1=Y1*R+A(I)
25    Y2=Y2*R+B(I)
      Y1=1.+Y1*R
      Y2=R*(1.-Y2*R)
      R=0.3989423/SQRT(X)
3     Y1=Y1*R
      Y2=Y2*R
      RETURN
      END
C
C---------------------------FUNCTION--WERF-----------------------------
C
C   THIS SUBROUTINE COMPUTES THE VALUE OF THE FUNCTION
C               WERF (X) = ERFC (X) * EXP (X**2),
C   ERFC(X) BEING THE COMPLEMENTARY ERROR INTEGRAL
C
C   CALLING SEQUENCE : Y=WERF (X)
C   TO PREVENT EXPONENT OVERFLOW X MUST BE GREATER THAN -13.216
C
C-----------------------------------------------------------------------
      FUNCTION WERF (X)
      DIMENSION P1(0:2),Q1(0:2),P2(0:4),Q2(0:4),P3(0:2),Q3(0:2)
      DATA P1,Q1/21.38533, 1.722276, 0.3166529,
     *           18.95226, 7.843746, 1.000000/
      DATA P2,Q2/7.373888, 6.865018, 3.031799, 0.5631696, 4.318779 E-5,
     *           7.373961, 15.18491, 12.79553, 5.354217, 1.0/
      DATA P3,Q3/-4.257996 E-2, -1.960698 E-1, -5.160823 E-2,
     *           0.1509421, 0.9214524, 1.0/
      IF (X.NE.0.) GO TO 30
20    WERF=1
      RETURN
30    Y=ABS(X)
      Z=Y*Y
      IF (Y.GT.0.46875) GO TO 130
      U=P1(0)+Z*(P1(1)+Z*P1(2))
      V=Q1(0)+Z*(Q1(1)+Z*Q1(2))
      W=X*U/V
      W=(1.-W)*EXP(Z)
115   WERF=W
      RETURN
```

```
130     IF (Y.GT.4.) GO TO 150
        U=P2(4)
        V=Q2(4)
        DO 140 J=3,0,-1
        U=U*Y+P2(J)
140     V=V*Y+Q2(J)
        W=U/V
        GO TO 170
150     U=P3(0)+(P3(1)+P3(2)/Z)/Z
        V=Q3(0)+(Q3(1)+Q3(2)/Z)/Z
        W=(0.5641896+U/V/Z)/Y
170     IF (X.LT.0.) W=2*EXP(Z)-W
        GO TO 115
        END
C
C--------------------------FUNCTION--ERFC--------------------------
C
C   THIS SUBROUTINE COMPUTES THE COMPLEMENTARY ERROR INTEGRAL,
C   NORMALIZED SO THAT IT EQUALS 1.0 FOR X=0.0 AND 2.0 FOR X
C   IN NEGATIVE INFINITY.FOR FURTHER DEFINITIONS AND DISCUSSIONS
C   SEE  L.ERDELYI,ED. HIGHER TRANSCENDENTAL FUNCTIONS, VOL.2
C   CALLING SEQUENCE : Y = ERFC(X)
C
C   THE RELATIVE ERROR IS BELIEVED TO BE LESS THAN 1.5 E-7
C
C-----------------------------------------------------------------
        FUNCTION ERFC (X)
        DIMENSION P2(0:3),Q2(0:3)
        DATA P10,P11,P12,Q10,Q11,P30,P31,P32,Q30,Q31
     *  /21.34533,     1.722276, 0.3166529,18.95226, 7.843746,
     *   -4.337986E-2,-0.196069,-5.168823E-2,.1509421,.9214524/
        DATA P2,Q2/7.373880, 6.865018, 3.031799, 0.5631696,
     *              7.373961,15.18491, 12.79553,  5.354217/
        IF (X.NE.0.) GO TO 1
        ERFC=1.
        RETURN
1       Y=ABS(X)
        Z=Y*Y
        IF (Y.LE.13.210) GO TO 2
        W=0.
        GO TO 8
2       IF (Y.GT.0.46875) GO TO 4
        U=P10+Z*(P11+Z*P12)
        V=Q10+Z*(Q11+Z)
        W=1.-Y*U/V
        GO TO 8
4       IF (Y.GT.4.) GO TO 6
        U=4.3187795E-5
        V=1.
        DO 5 J=3,0,-1
        U=U*Y+P2(J)
5       V=V*Y+Q2(J)
        W=U/Y
        GO TO 7
6       U=P30+(P31+P32/Z)/Z
        V=Q30+(Q31+1./Z)/Z
        W=(0.5641896+U/V/Z)/Y
7       W=W*EXP(-Z)
8       IF (X.LT.0.) W=2.-W
        ERFC=W
        RETURN
        END
```

Appendix 2

```
C-------------------------MIGR2-- .MAIN---------------------------------
C
C   PROGRAMMING LANGUAGE : DGC FORTRAN-5 FOR DGC NOVA-2 COMPUTER
C
C   HERE IS A SAMPLE PROGRAM TO DEAL WITH SUBROUTINE CYL12 AND TO SOLVE
C   THE MISCIBLE DISPLACEMENT PROBLEM (DISPERSIONLESS) AS DESCRIPTED
C   IN TEXT. ONLY THE I/O INSTRUCTION ARE GIVEN HERE.
C   ALL INPUT IS IN FREE FORMAT; REQUIRED (AND ASKED) PARAMETERS ARE:
C   A : THE VALUE OF INITIAL (PROFILE-CONSTANT) SOLUTE CONCENTRA-
C       TION IN COLUMN.
C   C0 : SOLUTE CONCENTRATION INTRODUCED AT THE TOP OF THE COLUMN.
C   T0 : A MOMENT OF TIME STARTING AT WHICH THE FILTRATE CONCENTRATION
C        IS "MEASURED".
C   X  : A COLUMN LENGTH.
C   BET: VELOCITY OF WATER MOVEMENT IN PORES, THAT IS, THE
C        EXTERNAL VELOCITY DIVIDED BY THE "TRANSFERENT" POROSITY.
C   DT : TIME INTERVAL LENGTH (WITH IT AS A PERIOD A "MEASUREMENTS" WILL
C        BE MADE).
C   LAM: A CONSTANT DESCRIBING THE RATE OF EXCHANGE BETWEEN STAGNANT
C        AND TRANSFERENT ZONES.
C   GAM: A RATIO BETWEEN PORE VOLUMES OF THE TWO ZONES.
C
C   NUMBER OF MEASUREMENTS IS SET TO 40. IF ANOTHER VALUE IS DESIRED,
C   JUST REPLACE THIS IN PARAMETER STATEMENT (M=...).
C
C   AN OUTPUT IS DIRECTED TO THE DISK FILE, NAMED 'MY.LC'.
C   THIS CAN BE EASILY REWRITTEN IF, FOR EXAMPLE, IMMEDIATE PRINTING
C   IS NEEDED.
C
C   WARNING : X MUST BE LESS THAN BET*TIME
C
      PARAMETER M=40
      DOUBLE PRECISION XI,TAU,U1,U2
      DIMENSION Y(0:M),T(0:M)
10    FORMAT (1X//T5,'BET=',F8.4,T18,
   #    'GAM=',F7.4,T30,'LAM=',F8.5/T5,
   #    'A=',F8.4,T18,'X=',F8.4//
   #    T8,'TIME',T24,'FILTRAT'//)
20    FORMAT (1X,T6,F8.5,T20,F3.5)
      ACCEPT 'A=',A,'C0=',C0,'T0=',T0,
   #          'X=',X,'BET=',BET,'DT=',DT,
   #          'LAM=',ALAM,'GAM=',GAM

      Q0=A-C0
      X1-X/BET
      DO 4 I=0,M
4     T(I)=T0+I*DT
      XI=X*GAM*ALAM/BET
      DO 6 I=0,M
      TAU=ALAM*(T(I)-X1)
      CALL CYL12 (XI,TAU,U1,U2)
      Y(I)=C0+Q0*U2
      IF (TAU.LT.0) Y(I)=0.
6     CONTINUE
      OPEN 2,'MY.LC'
      WRITE (2,10) BET,GAM,ALAM,A,X
      WRITE (2,20) (T(I),Y(I),I=0,M)
      CLOSE (2)
      STOP
      END
```

```
C------------------SUBROUTINE--CYL12--------------------------------------
C
C   THIS SUBROUTINE COMPUTES A PAIR OF GOLDSTEIN-TYPE FUNCTIONS, U1 AND
C   U2, WHICH GIVE A SOLUTION OF MISCIBLE DISPERSIONLESS DISPLACEMENT
C   PROBLEM. DEFINITIONS AND RELATIONS AND BIBLIOGRAPHY SEE IN :
C   "INTEGRALS OF BESSEL FUNCTIONS", BY Y.L. LUKE, N.Y.1962, CHAPTER XII.
C   SEE ALSO : "MODELIROVANIE POCHVENNYH PROCESSOV I AVTOMATIZACIYA IH
C   ISSLEDOVANIY", MOSCOW 1976, P. 75-78.
C   CALLING SEQUENCE : CAL CYL12(XA,YA,U1,U2),
C   WHERE XA AND YA ARE ARGUMENTS, U1 AND U2 BEING THE RESULTING
C   FUNCTIONS. SUPPORTING ROUTINES : DERF - FOR COMPUTING OF ERROR
C   FUNCTION AND IZEONE FOR COMPUTING OF THE TWO FIRST MODIFIED BESSEL
C   FUNCTIONS; HOWEVER ONLY ONE OF THEM IS USED HERE.
C
      COMPILER DOUBLE PRECISION
      SUBROUTINE CYL12 (XA,YA,U1,U2)
      DIMENSION CS(12)
      DATA SPI /.56418958354775629/
      DATA CS /
     *  0.99785892323860351,  0.98078528040323045,
     *  0.94693012949510566,  0.89687274153268830,
     *  0.83146961230254524,  0.75183980747897740,
     *  0.65934581510006887,  0.55557023301960222,
     *  0.44228869021900129,  0.32143945530316158,
     *  0.19509032201612827,  6.5403129230143067D-2/
      U1=0.
      IF (XA.NE.YA) GO TO 20
      IF (XA.GT.0.) GO TO 10
      GO TO 15
10    CALL IZEONE (2.*XA,Z0,Z1)
      U1=.5*(1.-Z0)
15    U2=U1
      RETURN
20    Y=DMIN1(XA,YA)
      X=DMAX1(XA,YA)
      XSQ=SQRT(X)
      YSQ=SQRT(Y)
      XI=2.*XSQ*YSQ
      CALL IZEONE (XI,Z0,Z1)
      Z1=XSQ-YSQ
      Z=Z1*Z1
      EZ=0.
      IF (Z.LE.174.67309) EZ=EXP(-Z)
      IF (XI.GT.20.) GO TO 40
C
C   FETTIS-LUKE TYPE 24-POINT TRAPEZOIDAL QUDRATURE
C
      DO 30 J=1,12
      YS=YSQ*CS(13-J)
      XM=YSQ*CS(J)
      XP=XSQ+XM
      XM=XSQ-XM
      T1=0.
      T2=0.
      IF (XP.LE.13.216) T1=EXP(-XP*XP)
      IF (XM.LE.13.216) T2=EXP(-XM*XM)
      U1=U1+CS(13-J)*(T1+T2)*DERF(YS,1)
30    CONTINUE
      U1=7.3552243787729834D-2*U1*YSQ
      GO TO 30
```

```
C   ASYMPTOTICS  I
C
40      Z2=(XSQ+YSQ)/SQRT(2.*XI)
        T=.5*Z/XI
        IF (T.LT.1.) GO TO 60
        R=SPI*Z2/Z1
        U1=1.
        Q=1.
        P=1.
        S=1.
        DO 50 N=1,25
        P=P*(.5-N)/Z
        S=S*T*(.5/N-1.)
        Q=Q+S
        W=Q*P
        IF (DABS(W/U1).LE..5E-15) GO TO 55
        U1=U1+W
50      CONTINUE
55      U1=.5*(R*U1-Z0)*EZ
        GO TO 80
C
C   ASYMPTOTICS  II
C
60      T=.25/XI
        P=1.
        R=DERF(Z1,3)
        U1=R
        S=SPI*Z1
        DO 70 N=1,25
        P=P*(2.-1./N)*T
        R=S-Z*R
        S=S*(N-.5)
        W=P*R
        IF (DABS(W/U1).LT..1D-15) GO TO 75
        U1=U1+W
70      CONTINUE
75      U1=.5*(Z2*U1-Z0)*EZ
80      U2=1.-Z0*EZ-U1
        IF (X.EQ.XA) RETURN
        Z=U2
        U2=U1
        U1=Z
        RETURN
        END

C--------------------FUNCTION--DERF-------------------------------------
C
C   DERF (X,1) = ERF (X)
C   DERF (X,2) = ERFC (X)
C   DERF (X,3) = ERFC (X) * EXP (X**2)
C
C   RATIONAL CHEBYSHEV APPROXIMATION DUE TO W.J.CODY,
C   SEE MATH. COMPUT.1969,VOL.23,#107,PP.631-637
C
        COMPILER DOUBLE PRECISION.
        FUNCTION DERF (X,K)
        LOGICAL NEG,LOG
        DIMENSION P1(0:4),Q1(0:3),P2(0:8),Q2(0:7),P3(0:5),Q3(0:4)
        DATA C /.5641895835477563/
        DATA P1,Q1/3209.3775891384695, 377.48523768530202,
     #           113.86415415105016, 3.1611237438705656,
     #           .18577770618460315, 2844.2368334391706,
```

```
     *           1282.6165260773723, 244.02463793444417,
     *           23.601290952344121/
      DATA P2,Q2/1230.3393547979973, 2051.0783778260715,
     *           1712.0476126340706, 881.95222124176909,
     *           298.63513819740013, 66.119190637141629,
     *           8.8831497943883759, .56418849698867009,
     *      .21531153547440385D-7, 1230.3393548037494,
     *           3439.3676741437216, 4362.6190901432472,
     *           3290.7992357334596, 1621.3895745666902,
     *           537.18110186200986, 117.69395089131250,
     *           15.744926110709835/
      DATA P3,Q3/-.65874916152983780D-3,-.16083785148742277D-1,
     *           -.12578172611122925, -.36034489994980444,
     *           -.30532663496123234, -.16315387137302098D-1,
     *            .23352049762686919D-2, .60518341312441319D-1,
     *            .52790510295142841, 1.8729528499234605,
     *            2.5685201922898224/
      Y=DABS(X)
      Z=Y*Y
      V=1.
      IF (Y.GT.0.46875) GO TO 20
      U=P1(4)
      DO 10 J=3,0,-1
      U=U*Z+P1(J)
      V=V*Z+Q1(J)
10    CONTINUE
      DERF=X*U/V
      IF (K.EQ.1) RETURN
      DERF=1.-DERF
      IF (K.EQ.3) DERF=DERF*DEXP(Z)
      RETURN
20    NEG=X.LT.0.
      LOG=Y.GT.13.216
      IF (LOG) GO TO (80,90) K
      IF (LOG.AND.NEG) GO TO 100
      IF (.NOT.LOG) T=DEXP(Z)
      IF (Y.GT.4) GO TO 40
      U=P2(8)
      DO 30 J=7,0,-1
      U=U*Y+P2(J)
      V=V*Y+Q2(J)
30    CONTINUE
      DERF=U/V
      GO TO 50
40    U=P3(5)
      DO 45 J=4,0,-1
      U=U/Z+P3(J)
      V=V/Z+Q3(J)
45    CONTINUE
      DERF=(C+U/V/Z)/Y
50    IF (K.EQ.3) GO TO 70
      DERF=DERF/T
      IF (K.EQ.2) GO TO 60
      DERF=1.-DERF
55    IF (NEG) DERF=-DERF
      RETURN
60    IF (NEG) DERF=2.-DERF
      RETURN
70    IF (NEG) DERF=2.*T-DERF
      RETURN
80    DERF=1.
      GO TO 55
```

```
 90     DERF=0.
        GO TO 60
100     TYPE 'X.LT.-13.216 IN DERF(X,3)'
        RETURN
        END

C------------------SUBROUTINE--IZEONE------------------------------------
C
C   THIS SUBROUTINE COMPUTES A PAIR OF FUNCTIONS OF ARGUMENT  X :
C   Z0 = EXP(-X) * I0 (X)    AND   Z1 = EXP(-X) * I1 (X) / X ,
C   WHERE I0 AND I1 ARE MODIFIED BESSEL FUNCTIONS OF ORDERS 0 AND 1
C   CORRESPONDENTLY. METHODS USED ARE : FETTIS-TYPE TRAPEZOIDAL
C   QUADRATURE FOR X.LE.20 AND ASYMPTOTIC EXPANSION FOR X.GT.20.
C   ERROR IS BELIEVED TO BE NOT GREATER THAN 5 IN 16-TH SIGNIFICANT
C   DIGIT.
C
      COMPILER DOUBLE PRECISION
      SUBROUTINE IZEONE (X,Z0,Z1)
      DIMENSION CS(12),SN2(12)
      DATA CS,SN2 /
     *   0.99785892323860351,  0.98078528040323045,
     *   0.94693012949510566,  0.89687274153268830,
     *   0.83146961230254524,  0.75183980747897740,
     *   0.65934581510006887,  0.55557023301960222,
     *   0.44228869021900128,  0.32143946530316153,
     *   0.19509032201612827,  6.5403129230143067D-2,
     *   4.2775693130947944D-3,3.8060233744356622D-2,
     *   0.10332332985438242,  0.19561928549563968,
     *   0.30865828381745511,  0.43473690388997420,
     *   0.56526309611002580,  0.69134171618254489,
     *   0.80438071450436932,  0.89667667014561758,
     *   0.96193976625564338,  0.99572243068690521 /
      IF (X.GE.20.) GO TO 20
      Z0=0.
      Z1=0.
      DO 10 J=1,12
      T=DCOSH(X*CS(J))
      Z0=Z0+T
      Z1=Z1+T*SN2(J)
10    CONTINUE
      T=DEXP(-X)/12.
      Z0=Z0*T
      Z1=Z1*T
      RETURN
20    Z0=1.
      Z1=1.
      T=1.
      P=8.*X
      L=3.5+360/X
      R1=1.
      R2=1.
      M=-8
      K=3
      DO 30 N=1,L
        M=M+8
        K=K-M
        R1=DFLOAT(4-K)*R1
        R2=-DFLOAT(K)*R2
        T=T/(P*DFLOAT(N))
        Z0=Z0+R1*T
        Z1=Z1+R2*T
```

```
30      CONTINUE
        T=1.1283791670955126/SQRT(P)
        Z0=Z0*T
        Z1=Z1*T/X
        RETURN
        END

C--------------------.MAIN---(MW0)-------------------------------
C
C   THIS PROGRAM COMPUTES THE REDISTRIBUTION OF MOISTURE IN
C   VERTICAL COLUMN DUE TO THE CAPILLARY CONDUCTIVITY AND GRAVITY.
C   ALL  CALCULATION  ARE  BEING  MADE IN TERMS OF SUCTION, AND
C   MOISTURE CONTENT IS COMPUTED FOR OUTPUT. ALL INPUT AND
C   INITIALIZATION PROVIDES SUBROUTINE INITL. SEE THERE ALSO IN
C   COMMENTARY THE DESCRIPTION OF ALL INPUT VARIABLES.
C   ALPHA (N) AND BETA (N) ARE THE FACTORIZING COEFFICIENTS
C   TO SOLVE THE FINITE DIFFERENCE EQUATION.
C   SUPPORTING ROUTINES : WATN, XFER, OUTPUT, INITL, DERF.
C   A LISTING FOR DERF SEE IN SUPPL. 2
C
        PARAMETER N=53
        COMMON /WATO/ H(0:N),A(N),F1(N)
        DIMENSION ALPHA(N),BETA(N),F(N-1),F2(N-1),HZ(N-1)
        CALL INITL (S,S1,S2,SM,DX,DT,EPS,JT)
        J3=JT
        ALPHA(1)=0.
40      CALL INTRA (J3*DT,BETA(1),FLQ)
        DO 50 J=1,N-1
        F2(J)=S2*F1(J)
        HZ(J)=H(J)
        F(J)=S*(A(J)*(H(J-1)-H(J)-DX)+
     *      A(J+1)*(H(J+1)-H(J)+DX))+DT*FLQ
50      CONTINUE
        J2=0
100     DO 110 J=1,N-1
        P=F2(J)+S1*F1(J)
        R=A(J)*(1.-ALPHA(J))+A(J+1)+P
        ALPHA(J+1)=A(J+1)/R
        BETA(J+1)=(A(J)*BETA(J)+F(J)+P*HZ(J)+DX*(A(J+1)-A(J)))/R
110     CONTINUE
        Z=0.
        R=BETA(N)/(1.-ALPHA(N))
        Z=AMAX1(Z,ABS(H(N)-R)/R)
        H(N)=R
        DO 120 J=N,1,-1
        R=ALPHA(J)*H(J)+BETA(J)
        Z=AMAX1(Z,ABS(H(J-1)-R)/R)
        H(J-1)=R
120     CONTINUE
        DO 130 J=1,N
        HL=.5*(H(J-1)+H(J))
        A(J)=SM*XFER(HL)
        F1(J)=WATN (H(J),2)
130     CONTINUE
        J2=J2+1
        IF (Z.GT.EPS.AND.J2.LT.60) GO TO 100
        J3=J3+1
        CALL OUTPUT (J3,H)
        GO TO 40
        END
```

```
C-------------------SUBROUTINE--INITL----------------------------
C
C   THIS SUBROUTINE INPUTS THE STARTING INFORMATION AND
C   INITIALIZES THE CALULATIONS.
C
C   AN EXPLANATION OF PARAMETERS FOLLOWS :
C
C   H : AN ARRAY OFOF SUCTION VALUES IN EACH X-POINT
C       OF FINITE-DIFFERENCE SCHEME.
C       IT'S DIMENSION IS 0:53, THOUGH THIS VALUE COULD BE RE-
C       PLACED WITH AN APPROPRIATE ONE (THE REST OF ARRAYS IS
C       HANDLED ANALOGOUSLY). TWO WAYS OF INPUT ARE PRESCRIBED :
C       H=CONST AT THE BEGINNING (L=0) OR H IS SUBSTITUTED
C       BY THE VALUES CALCULATED FROM ANY SET OF MOISTURE CONTENT,
C       STORED IN SUBROUTINE INTRA (L=1; SET IS DETERMINED BY THE
C       PARAMETER KB, KB=0,1,...,7).
C   A,F1 - ARRAYS USED IN MAIN PROGRAM; HERE THEY ARE
C       INITIALIZED.
C   A0,B0,C0 : CONSTANTS IN PF-CURVE APPROXIMATION.
C   H1 AND H2,A1,B1,V1,A2,B2,V2 : CONSTANTS IN EXPRESSIONS
C       CAPILLARY CONDUCTIVITY AS A FUNCTION OF SUCTION.
C   DX AND DT - THE INTERVALS OF DISCRETIZATION ALONG X-AXIS
C       AND T (TIME) - AXIS CORRESPONDINGLY.
C   ITME : "TIME-TO-PRINT" ARRAY, THAT IS A SET OF NUMBERS
C       OF TIME INTERVALS AT WCHICH AN OUTPUT IS TO BE DONE.
C   LTIM : THE LENGTH OF THIS ARRAY.
C   KB - NUMBER OF EXPERIMENTAL TIME, STARTING WITH IT
C       A CALCULATION IS PROCEEDED.
C   RQ - AN ARRAY OF EVAPORATION RATES FOR EACH EXPERIMENTAL
C       TIME INTERVAL.
C   S AND S1 - COEFFICIENTS IN FINITE DIFFERENCE SCHEME, S SHOULD
C       BE TAKEN IN INTERVAL FROM 0.5 TO 1.0, S1 - BETWEEN 0.0
C       AND 0.5
C
      PARAMETER N=53
      SUBROUTINE INITL (S,S1,S2,SM,DX,DT1,EPS,JT)
      COMMON /WAT0/ H(0:N),A(N),F1(N)
      COMMON /WAT2/ A0,B0,C0
      COMMON /WAT3/ H1,H2,A1,B1,V1,A2,V2,A3,V3
      COMMON /WAT4/ DT,IT,LMT,ITME(0:12)
      COMMON /WAT5/ W0(11,0:7)
      COMMON /WAT7/ RQ(0:6)
      ACCEPT 'T0=',ITME(0)
      JT=ITME(0)
      DX=0.2
      A0=0.2865
      B0=-1.28596
      C0=.372715
      H1=65.13917
      H2=7241.5994
      A1=1751.
      ACCEPT 'KB=',KB
      ACCEPT 'RQ(0:6) : ',(RQ(I),I=0,6)
      ACCEPT 'S=',S,'S1=',S1,'DT=',DT
      ACCEPT 'EPS=',EPS
      B1=1683.
      V1=2.2
      A2=640.
      V2=2.00
      A3=0.0978
      V3=1.01
      ACCEPT 'LMT=',LMT
```

```
      ACCEPT 'ITME : ',(ITME(I),I=1,LMT)
      ACCEPT 'H0 ? (0 OR 1) ',L
      IF (L.EQ.0) GO TO 200
      DO 100 I=1,10
      WIN=W0(I,KB)
      WR=DX*(W0(I+1,KB)-WIN)
      J1=(I-1)*5
      DO 100 J=0,4
      H(J1+J)=WATN(WIN+J*WR,3)
100   CONTINUE
      DO 150 J=50,53
150   H(J)=WATN(W0(11,KB),3)
      GO TO 300
200   ACCEPT 'H0=',H0
      DO 210 I=0,N
210   H(I)=H0
300   CONTINUE
      DT1=DT
      IT=1
      S2=(1-S1)/S
      S1=S1/S
      S=1/S-1
      SM=DT/(DX*DX)
      DO 30 J=1,N
      F1(J)=WATN (H(J),2)
      A(J)=SM*KFER(H(J))
30    CONTINUE
      RETURN
      END

C----------------FUNCTION--WATN-------------------------------
C
C    THIS SUBROUTINE COMPUTES THE VOLUMETRIC MOISTURE
C    CONTENT WITH SUCTION AS AN ARGUMENT (KEY=1), OR IT'S DERIVATIVE
C    WITH RESPECT TO SUCTION, AS AN ARGUMENT OF SUCTION (KEY=2), OR
C    SUCTION AS AN ARGUMENT OF MOISTURE CONTENT (KEY=3).
C    THE FORMULA IN USE IS THE FIRST APPROXIMATING FORMULA FOR
C    PF-CURVE.
C     CALLING SEQUENCE : Z=WATN(Y,KEY).
C    SUPPORTING ROUTINE : DERF, YNVERF
C
      FUNCTION WATN (Y,KEY)
      DOUBLE PRECISION DERF,YNVERF
      COMMON /WAT2/ A,B,C
      IF (KEY.EQ.3) GO TO 20
      WATN=0
      IF (Y.GT.0) GO TO 10
      IF (KEY.EQ.1) WATN=2*A
      RETURN
10    Z=B+C*ALOG10(Y)
      IF (Z.GT.13.2) RETURN
      S=Z*Z
      IF (KEY.EQ.1) WATN=A*DERF(DBLE(Z),2)
      IF (KEY.EQ.2) WATN=0.4900488*A*C*EXP(-S)/Y
      RETURN
20    WATN=EXP(2.302585*(YNVERF(1.-Y/A,IER)-B)/C)
      RETURN
      END
```

```
C---------------FUNCTION--XFER-------------------------------
C
C  THIS SUBROUTINE COMPUTES THE UNSATURATED CAPILLARY CONDUCTI-
C  VITY (XFER) AS A FUNCTION OF SUCTION (H). FOR
C  H < H1 GARDNER'S FORMULA IS IN USE, FOR H1 < H < H2 AND
C  H > H2 - FORMULA OF VISSER, WIND AND WESSELING (WITH TWO
C  SETS OF PARAMETERS).
C
      FUNCTION XFER (H)
      COMMON /WAT3/ H1,H2,A1,B1,V1,A2,V2,A3,V3
      IF (H.GT.H1) GO TO 30
      P=0.
      IF (H.GT.0) P=EXP(V1*ALOG(H))
      XFER=A1/(P+B1)
      RETURN
30    IF (H.GT.H2) GO TO 40
      XFER=A2*EXP(-V2*ALOG(H))
      RETURN
40    XFER=A3*EXP(-V3*ALOG(H))
      RETURN
      END

C----------------SUBROUTINE--OUTPUT-----------------------------
C
C   THIS IS A SAMPLE SUBROUTINE CONTROLLING THE CALCULATING OF
C   THE MOISTURE CONTENT AND THE OUTPUT OF RESULTING INFORMATION
C   (MOISTURE CONTENT AND SUCTION) TO A PERIPHERAL. HERE OUTPUT
C   IS BEING MADE TO A DISK FILE, NAMED "LISTA.LS".
C   SUPPORTING ROUTINE : WATN.
C
      PARAMETER N=53
      SUBROUTINE OUTPUT (J3,H)
      COMMON /WAT4/ DT,IT,LMT,ITME(0:12)
      DIMENSION H(0:N),W(0:N)
100   FORMAT (1X/1X,"T=",F7.4," DAY"//1X,12F6.4)
      IF (J3.LT.ITME(IT)) RETURN
      TYPE ' >>',J3
      T=J3*DT
      IT=IT+1
      DO 10 J=0,N
10    W(J)=WATN (H(J),I)
      OPEN 1,"LISTA.LS",ATT="A"
      WRITE (10,100) T,(W(5*I),I=0,N/5),W(53)
      WRITE (1,100) T,(W(5*I),I=0,N/5),W(53)
      CLOSE (1)
      IF (IT.GT.LMT) STOP
      RETURN
      END

C---------------------SUBROUTINE--INTRA------------------------------
C
C
C    THIS IS AN INTERPOLATING ROUTINE TO PROVIDE THE INITIAL DATA
C    AT EACH X-POINT OR/AND BOUNDARY CONDITIONS AT EACH T-POINT
C
C     FLW : MOISTURE CONTENT AT THE TOP ACCORDINGLY TO EXPERIMENTAL DATA,
C     FLQ : EVAPORATION RATE.
C
      SUBROUTINE INTRA (T,FLW,FLQ)
      COMMON /WAT6/ TIME(0:7),LAYR,LTIM,T0,T1,W01,W02,W03,W04
     *       /WAT5/ W(11,0:7)
      COMMON /WAT7/ PQ(0:6)
```

```
      DATA TIME /0.,1.,2.125,3.125,4.292,6.292,10.,14.042/
      DATA W/.433,.441,.443,.453,.458,.462,.466,.468,.470,.471,.472,
     *        .374,.381,.388,.395,.399,.402,.406,.409,.413,.415,.417,
     *        .293,.308,.320,.329,.337,.343,.350,.355,.359,.362,.365,
     *        .191,.233,.264,.283,.295,.303,.308,.313,.317,.321,.324,
     *        .114,.195,.232,.253,.267,.277,.285,.292,.297,.299,.302,
     *        .083,.138,.184,.217,.236,.246,.255,.262,.268,.271,.275,
     *        .061,.098,.133,.164,.190,.208,.219,.227,.232,.237,.241,
     *        .055,.085,.116,.142,.163,.178,.190,.200,.206,.212,.214/
C
      DO 10 KB=6,0,-1
      IF (T.GE.TIME(KB)) GO TO 15
10    CONTINUE
15    W01=W(1,KB)
      W02=W(1,KB+1)-W01
      T0=TIME(KB)
      T1=TIME(KB+1)-T0
20    RT=(T-T0)/T1
      FLW=WATM (W01+RT*W02,3)
      FLC=RO(KB)
      RETURN
      END
```

Appendix 3

```
C-----------------RASNA .MAIN--------------------------------------------
C
C    THE PROGRAM WRITTEN IN DGC FORTRAN 5 FOR DGC NOVA-2 COMPUTER
C
C    THIS PROGRAM WAS DESIGNED TO DEAL WITH A SYSTEM OF SIX DIFFERENTIAL
C    EQUATIONS DESCRIBING SODIUM CONVECTIVE DISPERSION AND SORPTIONAL
C    INTERCHANGE WITH CALCIUM. THE EQUATIONS AND ALGORITHM SEE IN TEXT.
C    HERE A BRIEF I/O INSTRUCTION IS PRESENTED.
C    ALL INPUT IS ACCOMPANIED WITH SUPPORTING QUESTIONS AND
C    EVERY VARIABLE IS INPUT IN FREE FORMAT.
C    IN : IF 0 THEN INITIAL CONDITIONS WILL BE INPUT FROM THE
C         CONSOLE, IF YOU SPECIALIZE  1  THEN AS INITIAL WILL BE
C         REGARDED THE INFORMATION, PREVIOUSLY STORED (PREPRODUCED
C         BY THIS PROGRAM) IN DISK FILE NAMED 'RASNA.RB'.
C    OUT : IF  1  THEN RESULTING INFORMATION  WILL  BE  PUT  IN
C          THAT DISK FILE, IF  THIS IS NOT DESIRED, SPECIALIZE  0.
C    B.C : - BOUNDARY CONDITIONS. IF 1 - A CONSTANT CONCENTRATION
C         AT THE TOP OF THE COLUMN WILL BE GIVEN, IF  2 - A BRENNER-
C         CONDITION WITH CONSTANT FLOW.
C    H AND H2 ARE THE COEFFICIENTS IN FINITE-DIFFERENCE SCHEME,
C    OUTPUT PARAMETER, ILMT, IS NUMBER OF PRINTS TO BE MADE
C         (AT GIVEN MOMENTS OF TASK TIME, THAT WILL BE INPUT FURTHER);
C    NITER IS NUMBER OF ITERATION STEPS,
C    CUPA  IS TOTAL SOLUTE CONCENTRATION AT THE TOP OF THE COLUMN,
C    C1UPA IS THE SAME FOR SODIUM;
C    EPSA AND EPSB ARE RELATIVE (VOLUME PER VOLUME) POROSITIES OF
C         TRANSFERENT AND STAGNANT ZONE CORRESPONDENTLY;
C    FLOW VELOCITY, VEL, IS AN EXTERNALLY OBSERVED VELOCITY OF SEEPAGE
C         (HIGHT PER UNIT OF TIME);
C    CAP  IS THE TOTAL ION EXCHANGE CAPACITY,
C    DT AND DX ARE THE FINITE TIME AND VERTICAL COORDINATE INTERVALS,
C         ADOPTED IN FINITE-DIFFERENCE SCHEME;
C    DFS  IS COEFFICIENT OF (MOLECULAR) DIFFUSION, AND
C    DISP IS THE COEFFICIENT OF LONGITUDINAL DISPERSION;
C    AK1  IS THE RATE OF THE TWO ZONES INTERCHANGE,,
C    AK2  IS THE RATE OF SORPTION,
C    AK3  IS THE ISOTHERM'S COEFFICIENT.
C    ITME  IS A SET OF ILMT INTEGERS, PRESENTING NUMBERS OF
C          OF STEPS AFTER WHICH AN OUTPUT OF CURRENT RESULT WILL BE
C          MADE; IN THIS VARIANT OF PROGRAM THAT INFORMATION IS DIRECTED
C          IN PRINTABLE FORMAT TO THE FILE NAMED 'RASNA.LC'.THE
C          CONTENTS OF THIS FILE COULD LATER BE PRINTED, IF DESIRED.
C    THE NEXT IS AN INPUT OF INITIAL VALUES. IT IS DONE FROM THE
C         DISK FILE 'RASNA.BN' (IN=1) OR FROM THE CONSOLE (IN=0).
C         THESE ARE :
C    CA0 - INITIAL (CONSTANT THROUGHOUT) TOTAL SOLUTE CONCENTRATION
C         IN A-ZONE (TRANSFERENT),
C    C1A0  IS THE SAME FOR SODIUM,
C    CB0 AND C1B0 ARE THE CORRESPONDENT VALUES FOR B-ZONE (STAGNANT),
C    SA0 AND SB0 ARE SORBED SODIUM CONCENTRATIONS FOR A- AND B-ZONE
C              CORRESPONDENTLY.
C
      IMPLICIT REAL (L,M)
      PARAMETER N=20
      DIMENSION ITME(20),RHB(0:N),CD(0:N),R(0:N),
     *    GAR(0:N),GBR(0:N),GA(0:N),GB(0:N),HR(0:N),
     *    ALF(N),ALF2(N),BET(N),OCA(0:N),OCB(0:N),
     *    OSA(0:N),OSB(0:N),CA(0:N),C1A(0:N),SLA(0:N),
     *    CB(0:N),C1B(0:N),SLB(0:N),SA(0:N),SB(0:N)
20    FORMAT (1X/T5,'T=',F7.4/T11,'C1',T21,'C2',T31,
```

```
     #    'S1',T41,'CA',T51,'C1A',T61,'C2A',T71,'SA',
     #    T81,'CB',T91,'C1B',T101,'C2B',T111,'SB'//)
30        FORMAT (1X,(T5,11F10.5))
          ACCEPT 'IN ? (0 OR 1) ',IN,'OUT ? (0 OR 1) ',IOUT,
     #    'B.C.? (1 OR 2) ',KBOUND,'H=',H,'H2=',H2,
     #    'OUTPUT QUNTY ?',ILMT,'NITER=',NITER,'CUPA=',
     #    CUPA,'C1UPA=',C1UPA,'EPSA=',EPSA,'EPSB=',EPSB,
     #    'FLOW VELOCITY=',VEL,'CAPACITY=',CAP,'DT=',DT,
     #    'DX=',DX,'DFS=',DFS,'DISP=',DISP,'AK1=',AK1,
     #    'AK2=',AK2,'AK3=',AK3,'ITME:',(ITME(J),J=1,ILMT)
          IF (IN.EQ.0) GO TO 100
          OPEN 2,'RASNA.BN'
          READ BINARY (2) CA,C1A,CB,C1B,SA,SB
          CLOSE (2)
          GO TO 120
100       ACCEPT 'CA0,C1A0,CB0,C1B0,SA0,SB0=',
     #    CA0,C1A0,CB0,C1B0,SA0,SB0
          DO 110 I=0,N
          CA(I)=CA0
          C1A(I)=C1A0
          CB(I)=CB0
          C1B(I)=C1B0
          SA(I)=SA0
110       SB(I)=SB0
120       OPEN 1,'RASNA.LC',ATT='A'
          WRITE FREE (1) 'B.C.=',KBOUND,'H=',H,'H2=',H2,'NITER=',
     #    NITER,'<15>CUPA=',CUPA,'C1UPA=',C1UPA,'EPSA=',EPSA,
     #    'EPSB=',EPSB,'<15>VEL=',VEL,'DT=',DT,'DX=',DX,'DFS=',
     #    DFS,'<15>DISP=',DISP,'AK1=',AK1,'AK2=',AK2,'AK3=',AK3
          H2S=1-H2
          VEL=VEL/EPSA
          D0=(DFS+VEL*DISP)/DX
          TX=DT/DX
          AZ1=EPSA/(EPSA+EPSB)
          AZ2=1-AZ1
          N1=N-1
          AKAP=AK3*CAP*AZ1
          BKAP=AK3*CAP*AZ2
          B0=D0*TX
          A0=VEL*TX+B0
          G0=A0+B0
          A1=A0*H
          A1S=A0-A1
          B1=B0*H
          B1S=B0-B1
          A2=A0*H2
          A2S=A0-A2
          B2=B0*H2
          B2S=B0-B2
          LAM=AK1*DT
          LAM0=LAM/(EPSB+LAM*H)
          LAM1=H*LAM0
          LAM1S=LAM0-LAM1
          LAM2=1-LAM0
          LAM0=LAM0*EPSB/EPSA
          LAM4=LAM*H2
          LAM3=EPSB+LAM4
          LAM4S=LAM-LAM4
          G2=G0+LAM0
          G1=G2*H+1
          G1S=G2-G1
          G2=G0*H2+1
```

```
          G2S=G0-G2
          H2A=H2/EPSA
          M3=AK2*DT
          M0=M3/(1+M0*H2)
          M1=M0*M2
          M1S=M0-M1
          M2=1-M3
          GO TO (125,130) KBOUND
125       ALF(1)=0
          ALF2(1)=0
          GO TO 140
130       F=VEL/(VEL+DS)
          ALF(1)=1-F
          ALF2(1)=ALF(1)
          CUPA=CUPA*F
          C1UPA=C1UPA*F
140       DO 150 I=1,N1
150       ALF(I+1)=B1/(G1-A1*ALF(I))
          J3=1
          IT=0
          IS=NITER
          GO TO 240
C   CA & CB
160       IT=IT+1
          BET(1)=CUPA
          DO 170 I=1,N1
          F=A1S*CA(I-1)-G1S*CA(I)+B1S*CA(I+1)+LAM0*CB(I)
170       BET(I+1)=ALF(I+1)*(A1*BET(I)+F)/B1
          SA(N)=BET(N)/(1-ALF(N))
          DO 180 I=N,1,-1
180       SA(I-1)=ALF(I)*SA(I)+BET(I)
          DO 190 I=0,N
          CB(I)=LAM2*CB(I)+LAM1*SA(I)+LAM1S*CA(I)
190       CA(I)=SA(I)
C   C1A
200       BET(1)=C1UPA
          DO 210 I=1,N1
          Z=1/(G2+H2A*(R(I)+M0*GA(I))-A2*ALF2(I))
          ALF2(I+1)=B2*Z
          F=OCA(I)+(LAM4*HR(I)*(OCB(I)+GBR(I))-
     *      H2S*R(I)*CD(I)+GAR(I))/EPSA
210       BET(I+1)=Z*(A2*BET(I)+F)
          C1A(N)=BET(N)/(1-ALF2(N))
          DO 220 I=N,1,-1
220       C1A(I-1)=ALF2(I)*C1A(I)+BET(I)
C   C1B & SB & SA
          DO 230 I=0,N
          C1B(I)=HR(I)*(GBR(I)+LAM4*C1A(I)+RHB(I))
          SB(I)=OSB(I)+M1*GB(I)*C1B(I)-GBR(I)
230       SA(I)=OSA(I)+M1*GA(I)*C1A(I)-GAR(I)
C
240       DO 250 I=0,N
          CALL SARG (AK3,AKAP,CA(I),C1A(I),SLA(I),GA(I))
          CALL SARG (AK3,BKAP,CB(I),C1B(I),SLB(I),GB(I))
          Z=M1*GB(I)
          HR(I)=1/(LAM3+Z)
          R(I)=LAM*HR(I)*(EPSB+Z)
          GAR(I)=M1*(GA(I)*C1A(I)-SLA(I))
250       GBR(I)=M1*(GB(I)*C1B(I)-SLB(I))
          IS=IS+1
          IF (IS.LT.NITER) GO TO 200
          IS=0
          DO 260 I=0,N
```

```
      OSA(I)=M2*SA(I)+M1S*SLA(I)
      OSB(I)=M2*SB(I)+M1S*SLB(I)
      OCB(I)=CB(I)-OSB(I)+EPSP*C1B(I)
      CD(I)=C1A(I)-C1B(I)
260   RH3(I)=OC2(I)+LAM4S*CD(I)
      DO 270 I=1,N1
270   OCA(I)=A2S*C1A(I-1)-G2S*C1A(I)+
     *  D2S*C1A(I+1)+(SA(I)-OSA(I))/EPSA
      IF (IT.LT.ITME(J3)) GO TO 160
      J3=J3+1
C  OUTPUT
      T=IT*DT
      WRITE (1,20) T
      DO 280 I=0,N
      Z=AZ1*C1A(I)+AZ2*C1B(I)
      F=AZ1*CA(I)+AZ2*CB(I)-Z
      G0=SA(I)+SB(I)
      A0=CA(I)-C1A(I)
      B0=CB(I)-C1B(I)
280   WRITE (1,30) Z,F,G0,CA(I),C1A(I),A0,SA(I),CB(I),C1B(I),B0,SB(I)
      IF (JZ.LE.ILMT) GO TO 160
      CLOSE (1)
      IF (IOUT.EQ.0) STOP 0.
      OPEN 2,'RASNA.BN'
      WRITE BINARY (2) CA,C1A,CB,C1B,SA,SB
      CLOSE (2)
      STOP 1
      END

C---------------SUBROUTINE--SARG-----------------------------------------
C
C   THIS SUBROUTINE IS PROVIDED FOR "RASNA" SYSTEM AND CALCULATES AN
C   EQUILIBRIUM  SORBED SODIUM CONCENTRATION  (SL)  AS WELL AS ITS
C   DERIVATIVE  (G)  WITH RESPECT TO SODIUM SOLUTE CONCENTRATION.
C   INPUT ARGUMENTS : SKAP - ION EXCHANGE CAPACITY,
C                     AK   - COEFFICIENT OF ISOTHERM
C                     C    - TOTAL SOLUTE CONCENTRATION
C                     C1   - SODIUM SOLUTE CONCENTRATION
C   AN ISOTHERM IN USE IS THAT OF GAPON. IF OTHER ISOTHERM IS
C   REQUIRED, THIS SUBROUTINE SHOULD BE REWRITTEN.
C
      SUBROUTINE SARG (AK,SKAP,C,C1,SL,G)
      IF (C1.GE.C) C1=.9999*C
      Z=C-C1
      Z1=SQRT(Z)
      Z2=1/(AK*C1+Z1)
      SL=SKAP*C1*Z2
      G=.5*SKAP*(C+Z)*Z2*Z2/Z1
      RETURN
      END
```