1

Main (1)

COMPACTION PHENOMENA IN SEDIMENTARY SEQUENCES
***SKEMPTON VERSION***

********DATE: DECEMBER 20, 1984**********

NOTES: 1. THIS RUNS ON THE IBM FORTRAN IV COMPLIER G1 RELEASE 2.0.
2. A DATA SET IS APPENDED TO THE END OF THIS PROGRAM.
3. THIS PROGRAM GENERATED THE RESULTS IN SHALES AND ABNORMAL PRESSURES BY JE SMITH, GC DYSINGER AND RL BORST, GEOTECHNICAL AND ENVIROMENTAL ASPECTS OF GEOPRESSURE-ENERGY, ED. SUBENDRA K SAXENA, ENGINEERING FOUNDATION CONF AT SEA ISLAND, GA, JAN 13-18, 1980, PP.69-81.

COMPUTATIONS ARE CARRIED OUT BY THIS PROGRAM TO GIVE, AT CLOSELY SPACED POINTS IN TIME, PROFILES OF POROSITY, SEDIMENT AGE, WATER PRESSURES, FLOW RATES, AND OTHER QUANTITIES DERIVED THEREFROM FOR SEDIMENTARY SEQUENCES. THE COMPUTATIONS BEGIN WITH THE INITIATION OF SEDIMENTATION AT TIME ZERO. THE TOTAL TIME SPAN CONSIDERED IN A PROBLEM IS THE SUM OF A NUMBER OF TIME INTERVALS. ONE AND ONLY ONE OF THE FOLLOWING FIVE THINGS OCCURS DURING A TIME INTERVAL. (1) A SHALE UNIT IS DEPOSITED. (2) A SAND UNIT IS DEPOSITED. (3) THE TIME INTERVAL PASSES WITHOUT ANY SEDIMENTATION. (4) A SHALE UNIT IS ERODED. (5) A SAND UNIT IS ERODED.

IN THIS WAY THE ENTIRE MECHANICAL HISTORY OF ANY SEQUENCE OF SHALES AND SANDS IS RECONSTRUCTED DURING AND FOLLOWING DEPOSITION. TIME INTERVALS OF SHALE DEPOSITION OR EROSION, SAND DEPOSITION OR EROSION, OR NO DEPOSITION MAY OCCUR IN ANY ORDER, EXCEPT THAT (1) A SHALE UNIT MUST BE DEPOSITED IN THE FIRST TIME INTERVAL, AND (2) A UNIT CAN BE ERODED ONLY AFTER IT HAS BEEN DEPOSITED, AND ONLY IF IT IS CURRENTLY THE TOPMOST UNIT IN THE SECTION.

THE WORD "SHALE" IS USED TO MEAN A COMPRESSIBLE LITHOLOGY WITH A LOW PERMEABILITY. A SHALE IS DEFINED BY SPECIFYING (1) THE GRAIN DENSITY, (2) THE POSITIVE OR ZERO CONSTANTS A1 THROUGH A5 IN THE FOLLOWING RELATIONSHIP BETWEEN THE POROSITY, POR, THE MATRIX PRESSURE, SIGMA, AND THE SHALE AGE,

POR= A1*EXP(-A2*(AGE +A3)**A4*SIGMA**A5 ) , (1)

Edward Smith Rowland
3413 Edgewood Rd
Ellicott City MD
2 of 3

AND (3) THE POSITIVE OR ZERO CONSTANTS B1 THROUGH B4 IN THE FOLLOWING RELATIONSHIP BETWEEN PERMEABILITY, K, AND THE POROSITY

K= POR**B1*EXP(B2+B3*POR)/(1-POR)**B4 . (2) Main (2)

THE WORD ''SAND'' IS USED TO MEAN AN INCOMPRESSIBLE LITHOLOGY WITH VERY HIGH PERMEABILITY. A SAND IS DEFINED BY GIVING THE GRAIN DENSITY AND POROSITY. THE PERMEABILITY IS SUPPOSED TO BE GREAT ENOUGH SO THAT WATER FLOW VERTICALLY IN A SAND HAS A NEGLIGIBLE EFFECT ON THE VERTICAL WATER PRESSURE GRADIENT WITHIN THE SAND, G*RHOW. G IS THE GRAVITATIONAL CONSTANT AND RHOW IS THE WATER DENSITY. THE PRESSURE INCREASE VERTICALLY IN A SAND OF THICKNESS H IS THEN G*RHOW*H.

ONE OF THREE TYPES OF BOUNDARY CONDITION MAY BE IMPOSED AT THE BASE OF A SHALE UNIT. THE FIRST IS THAT ALL WATER LEAVING (ENTERING) THE NEXT LOWEST SHALE UNIT ENTERS (LEAVES) THE BASE OF THE SHALE UNIT. THIS IMPLIES THAT ANY INTERVENING SAND(S) ARE PART OF A CLOSED LENS. THIS CONDITION CANNOT BE IMPOSED AT THE BASE OF THE BOTTOM-MOST SHALE UNIT. FOR WHICH AN ALTERNATE CONDITION IS THAT NO WATER CAN ENTER OR LEAVE THE BASE. THE THIRD TYPE OF B. C., WHICH MAY BE IMPOSED AT THE BASE OF ANY SHALE UNIT, IS A SPECIFICATION OF THE WATER PRESSURE THERE BY ASSIGNING VALUES TO CONSTANTS ALPHA AND BETA IN THE EQUATION

P= ALPHA*(S-PN)+ BETA*(PN-PSW) +PSW . (3)

S IS THE OVERBURDEN STRESS, PN IS THE NORMAL FLUID PRESSURE FOR THE DEPTH BELOW THE SEA SURFACE, AND PSW IS THE WATER PRESSURE AT THE SEA FLOOR. S AND PN GENERALLY VARY WITH TIME, DEPENDING ON THE CONDITIONS OF THE PROBLEM. NOTE THAT P=PN FOR ALPHA=0 AND BETA=1 , AND P=S FOR ALPHA=BETA=1. EQUATION (3) DETERMINES WHETHER THE INTERVENING SANDS WILL BE SOURCES OR SINKS FOR SHALE WATERS.

THE COMPUTATIONAL TECHNIQUE IN THIS PROGRAM REQUIRES THAT TIME INTERVALS BE DIVIDED INTO ONE OR MORE SUB-INTERVALS. IF SHALE IS BEING DEPOSITED DURING A TIME INTERVAL THERE MUST BE AT LEAST THREE SUB-INTERVALS. THE ACCURACY AND DETAIL OF SOLUTIONS INCREASES WITH THE NUMBER OF SUB-INTERVALS, AS DOES THE TIME NEEDED FOR THE COMPUTATIONS.

THE TIME LAPSE DURING A GIVEN SUB-INTERVAL, AND THE THICKNESS (IF ANY) OF SAND OR SHALE AT ZERO POROSITY WHICH IS DEPOSITED DURING THAT SUB-INTERVAL, ARE GIVEN BY THE PROGRAM USER. THE

```
C  PRODUCT OF THE NUMBER OF SUB-INTERVALS IN A TIME INTERVAL, NS-
C  UBI, AND THE TIME LAPSE PER SUB-INTERVAL,DDT, GIVES THE TOTAL
C  TIME INTERVAL, TT.

C     TT = NSUBI*DDT     (4)

C  THE TOTAL THICKNESS OF ZERO-POROSITY SEDIMENTS DEPOSITED IN
C  THE TIME INTERVAL, THICK, IS THE PRODUCT OF THE THICKNESS OF
C  ZERO-POROSITY SEDIMENTS DEPOSITED PER SUB-INTERVAL, DTHICK,
C  AND THE NUMBER OF SUB-INTERVALS.

C     THICK= DTHICK*NSUBI.     (5)

C
C  DEFINITE UPPER LIMITS ON THE SIZE OF PROBLEMS WHICH CAN BE
C  RUN WITH THIS PROGRAM HAVE BEEN SET, AS IS REQUIRED IN FORTRAN
C  PROGRAMS. THE MAXIMUM NUMBER OF TIME INTERVALS IS 20. THE SUM
C  OVER ALL SHALE UNITS OF   (NSUBI+1)   MAY NOT EXCEED 130. THERE
C  MAY BE ONE TO FOUR   SETS OF FIVE OR FOUR PARAMETERS RESPEC-
C  TIVELY FOR EQUATIONS (1) AND (2). THESE LIMITS MAY BE CHANGED.
C
C  AT SOME FUTURE TIME IT MAY BE DESIRABLE TO BE ABLE TO CHANGE
C  THE FORM OF EQUATIONS (1) OR (2). TO MAKE THIS EASIER, COMMENT
C  CARDS WITH * AND ** AT ABOUT COLUMN 12 PRECEED THE AFFECTED
C  EQUATIONS.
C
C-----------------------------------------------------------------
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON / READIN / CASEID(20), TEMPT , GRADT, WTPCT, AA(4,5),
     1BB(4,4), ITYPE(20), NSUBI(20),DDT(20),DDZ(20), RHOG(20),
     2PORS(20),IBC(20), ALPHA(20), BETA(20), ISETA(20), ISETB(20),
     3IPAGE(20),     DP(20),DIT(20),GAMMA(20), RHOW, IZ,ITMAX
C
      DATA    IV / 'Z' /,KZ/' '/
C
      COMMON/ABC/DU(129),DUO(129),DUOO(129), YO(129),    AGE(129),
     1    Y(129), YOO(129),YMIN(129),F(129),SMX(129),
     1     LB(20), LT(20), NDTSUM(20), H(20), SSIGMA(20),
     2SMPN(20),GDRHO,LLTM,LO, HLL, GW
C
      COMMON/ERR/ NDT
      COMMON/ERODE/ LNOW(20), ITNOW
      COMMON/XXCO/ITER8S,NEWTYP,NMAX
```

Main (3)

```
      COMMON / SPACES / X(258), A11(129), B(258), XA(258),BRACK,
     1 C(129,129),ET   ,D(129,129),XX(1160),
     1 C(129,129),ET   ,D(129,129),XX(1160),
     2REST,WW ,XA0(258)
```

Math 4

```
      THE WORDS BRACK AND ET   *BRACKET*  C(NMAX,NMAX), ABOVE, SO THAT
      CALL COMLZ CAN LATER  DETERMINE NMAX . NMAX IS THE SIZE OF
      MATRICES  C AND D.


          STATEMENT FUNCTIONS

      F=DECOMPACTION PARAMETER
      A5=SKEMPTON'S LIQUID LIMIT
      PI(A5)=-12.39+.774*A5
      PL(A5)=A5-PI(A5)

      BSIG(F)=DEXP((3.07+(.23*A5+12.4)/(-12.4+.77*A5))/.175)+F-F

      ALPHX(F,SMX)=(.23*A5+12.4)+(-12.4+.77*A5)*(3.07-.175*DLOG(BSIG(F))
     1-.175*(1.-F)*DLOG(1.-DEXP(-(SMX+100.)/BSIG(F))))
      BETX(F)=.175*F*(-12.4+.77*A5)

      W(F,SIG,SMX)=-BETX(F)*DLOG(-DEXP((-SIG-100.)/BSIG(F))+1.)+ALPHX(F,
     1SMX)
      O(F,SIG,SMX)=.027*W(F,SIG,SMX)/(1+.027*W(F,SIG,SMX))

      QIN(PO,F,SMX)=DEXP((ALPHX(F,SMX)-PO/.027/(1.-PO))/BETX(F))

      SIG(PO,F,SMX)=-100.-BSIG(F)*DLOG(1.-QIN(PO,F,SMX))

      SIGO(PO,F,SMX)=-QIN(PO,F,SMX)/(1.-QIN(PO,F,SMX))*BSIG(F)/BETX(F)/.
     &027/(1.-PO)**2

         **     PERMEABILITY
      PERM(PO  ) = PO**B1*DEXP( B2+B3*PO)/(1.-PO)**B4


      EQUATION FOR THE Z-COORDINATES OF ALL INTERVAL AND BASE NODES OF
      ALL SHALE UNITS. D(K)= U(LLT)- U(K).  U(K)= Z COORDINATE OF
      NODE K.

      CHIU( D1,D2, O1,O2, DZ1 )=((1.-0.5*(O1+O2))*( D1-D2)-DZ1 )

          SUBSCRIPT NOTATION
```

```
3
      L= NUMBER OF SHALE UNIT. BOTTOM SHALE UNIT HAS L=1.
      LL= UNIT NUMBER OF TOPMOST EXISTING SHALE AT ANY GIVEN TIME.
      K = GENERAL NODE NUMBER OF SHALE. ALL NODES NUMBERED UPWARD FROM
          BOTTOM OF SEQUENCE WITH K=1 TO K=LLT
      LLT = NODE NUMBER AT TOP OF UNIT LL.
      HLL = THICKNESS OF SAND OVER UNIT LL.
      LLB = NODE NUMBER AT  BOTTOM OF UNIT LL.
      LB =     **     **     ** BOTTOM OF UNIT L.
      LT =     **     **     ** FINAL TOP OF UNIT L.

      READ IN DATA CASE AFTER ZEROING-OUT THE COMMON.
      BQ=0.

      CALL COMLZ ( CASEID(I), ITMAX, NIHIL )

      READ(5,900)     CASEID     , TEMPT, GRADT, WTPCT,
     1 ((AA(I,J), J=1,5 ), I=1,4 ),((BB(I,J) , J=1,4 ), I=1,4 )

      READ (5,901) ITYPE(1), ITYPE(2), NSUBI(1), NSUBI(2), DDT(1),
     1DDT(2), DDZ(1), DDZ(2), RHOG(1), RHOG(2), PORS(1), PORS(2),
     2IBC(1), IBC(2), ALPHA(1), ALPHA(2), BETA(1), BETA(2), ISETA(1),
     3ISETA(2), ISETB(1), ISETB(2), IPAGE(1), IPAGE(2)

      THERE MAY BE ONE TO THREE MORE PAGES OF DATA.
      LOOK AHEAD TO SEE.

      DO 1  J=1,3
      CALL NEXT ( M, KZ )
      IF(M.NE.0) GO TO 2

      IF(KZ. EQ. IV )  READ(5,909) KZ, IZ, DP, DIT
      IF(KZ. EQ. IV )  GO TO 2
      II=  3   +  9*(J-1)
      IL=  II  +  9*(J-1)

      READ (5,902) ( ITYPE(I), I=II,IL ) , ( NSUBI(I), I=II,IL ) ,
     2(DDT(I), I=II,IL ), ( DDZ(I), I=II,IL ), (RHOG(I), I=II,IL) ,
     3(PORS(I), I=II,IL), (IBC(I), I=II,IL) , (ALPHA(I), I=II,IL ),
     4(BETA(I), I=II,IL), (ISETA(I), I=II,IL), (ISETB(I), I=II,IL),
     5(IPAGE(I),I=II,IL)

      CONTINUE

      WRITE INPUT BACK OUT.

      WRITE(6,903)  CASEID, TEMPT, GRADT, WTPCT, (( AA(I,J), J=1,5),
     1     ( BB(I,K), K=1,4), I=1,4 )
```

Main (6)

```
      DO 3  J=1,2
        I= 1+ 10*(J-1)
        L= 10 +10*(J-1)
      IF(J.EQ.2) GO TO 4
      WRITE(6,905)  ( K, K=I,L )
      GO TO 3
      IF(ITYPE(11).EQ.0) GO TO 6
      WRITE(6,906)   ( K ,  K=I,L )
      WRITE(6,907) (ITYPE(K), K=I,L), (NSUBI(K), K=I,L ), (DDT(K),K=I,L)
     1, (DDZ(K), K=I,L), ( RHOG(K), K=I,L), ( PORS(K),K=I,L),
     2(IBC(K), K=I,L ), (ALPHA(K),K=I,L ), (BETA(K),K=I,L),(ISETA(K),K=
     3I,L ), (ISETB(K), K=I,L ) ,(IPAGE(K), K=I,L)

      CONTINUE

      IF(KZ.EQ.IV)    WRITE(6, 914)  IZ,(DP(I),I=1,10),(DIT(I),I=1,10),
     1 (DP(I),I=11,20),(DIT(I),I=11,20)

      CONVERT TO CGS UNITS AND DEG C.


      TEMPT =(TEMPT -32. )* 5./ 9.
      GRADT = GRADT * 5./ 9. / 30.48
      SECPMY= 3.1558*10.0**13.0

      DO  7  J=1,4
          *
          **
      BB(J,3)=BB(J,3)*100.
          **
7     BB(J,2) = BB(J,2) + ALOG (0.98687*10.0**(-11.))
     1+ ( BB(J,1) -BB(J,4))*4.60517

      CM**2 = 1.013*10**8 DARCY= 1.013*10**11 MILLIDARCY.

      DO  9  I=1,20
      DP(I) = DP(I) * 68947.0
9     DIT(I)= DIT(I)* 3.1558* 10.0**7.

      IT= TIME INTERVAL NUMBER
      ITMAX = NUMBER OF TIME INTERVALS SPECIFIED IN THE PROBLEM.
      ITIST=FIRST TIME INTERVAL IN WHICH EROSION OCCURS.

      IB = 20
```

```
 4    ITMAX=0
      ITIST=0
      DO 8 IT= 1,IB
      IF(ITIST.EQ.0.AND.ITYPE(IT).GE.4)ITIST=IT
 8    IF( ITYPE(IT).NE.0) ITMAX =ITMAX+1

      DO 10 IT=1,ITMAX
      DDT(IT)= DDT(IT)*SECPMY/ 1000.0
      DDZ(IT) = DDZ(IT)/.03281
10    PORS(IT)= PORS(IT) * 0.01


      RHOW = 0.9971 +0.0070*WTPCT
      GW   = 980.7 * RHOW

      ------------------------------------------------------------

      PREPARE TO SET UP AND SOLVE EQUATIONS FOR O(K) AND D(K).


      CALL INTPRT(10)
      CALL COMLZ( DU(1), HLL     , NIHIL )
      CALL COMLZ( BRACK , ET , NUMBER )

      NMAX = SQRT (   (NUMBER-16.)/8.)+0.1

      NDTSUM(1)= NSUBI(1)

      DO 14   I=2,ITMAX
      NDTSUM(I)= NDTSUM(I-1)+ NSUBI(I)
14    CONTINUE


      VECTOR LNOW CONTAINS EITHER ZERO OR THE NUMBER. FROM THE
      BOTTOM OF THE SECTION. OF A SHALE UNIT CURRENTLY BEING
      DEPOSITED OR ERODED.

      LLAST=1
      LNOW(1)=1
      DO 118 IT=2,ITMAX
      IF(ITYPE(IT).EQ.1) LLAST=LLAST+1
      IF(ITYPE(IT-1).EQ.4) LLAST=LLAST-1
      LNOW(IT)=0
```

Main (7)

```
3   IF(ITYPE(IT).EQ.1.OR.ITYPE(IT).EQ.4) LNOW(IT)=LLAST
```

Main 8

```
    SET UP INDICES AND INITIAL VALUES PRIOR TO MAIN CALCULATION.

    WW=1.
    TIME=0.0
    IFIRST =0
    LL=0
    LLT=0
    NDT = 0
    ICOUNT=0

    CALL TIMINT(IJI)
        TOLERANCES FOR CONVERGENCE

    ZIP=0.00001
    TOLDU=ZIP
    TOLU= ZIP
    TOLY= ZIP

    INCREMENT TIME. NDT IS THE TOTAL NUMBER OF ELAPSED TIME SUB-
    INTERVALS.

77  NDT = NDT+1

    IF (NDT. GT. NDTSUM(ITMAX).AND.KZ.NE. IV    )  GO TO 999
    IF (NDT. GT. NDTSUM(ITMAX).AND.KZ.EQ. IV    )   GO TO 998

    ITNOW IS THE CURRENT MAXIMUM VALUE OF IT.
    IT IS THE NUMBER OF A TIME INCREMENT.
    NEWTYP =0 GENERALLY, EXCEPT NEWTYP=1 DURING FIRST TIME SUB-
    INTERVAL IN EACH TIME INTERVAL.

    IF(NDT.EQ.1) ITNOW=1
    NEWTYP=0
    IF( NDT.GT.NDTSUM(ITNOW).OR.NDT.EQ.1) NEWTYP =1
    IF(NDT.NE.1 )  ITNOW = ITNOW + NEWTYP
    ITY=ITYPE(ITNOW)

    NEWSHL =0 GENERALLY,EXCEPT NEWSHL=1 DURING FIRST TIME SUB-
    INTERVAL DURING WHICH SHALE IS BEING DEPOSITED.

    NEWSHL = 0
```

```
5    IF(  ITY     .EQ.1  )  NEWSHL= NEWTYP
     LL=   LL +NEWSHL
          *****
     IF(ITY.NE.1) GO TO 8069
     LLT=LLT+1+NEWSHL
     F(LLT)=1.
     SMX(LLT)=0.
     YMIN(LLT)=1.
     IF(NEWSHL.NE.1) GO TO 8069
     F(LLT-1)=1.
     SMX(LLT-1)=0.
     YMIN(LLT-1)=1.
069  CONTINUE
     NEG=1
     IF(ITY.GT.3) NEG=-1
     IF(NEWSHL*ITY.EQ.1) LB(LL)=LLT-1
     IF(NEWSHL*ITY.EQ.1) LT(LL)=LB(LL)+NSUBI(ITNOW)
     IF(ITY.EQ.4) LLT=LLT-1
     IF(NDTSUM(ITNOW).NE.NDT.OR.ITY.NE.4) GO TO 119
     LL=LL-1
     LLT=LLT-1
19   CONTINUE
          *****
     DT = DDT( ITNOW)
     GO TO  29

98   CALL DEPLET( DT,     ICOUNT,&999 )

     ITY=ITYPE(ITMAX)
9    TIME = TIME + DT
     KK= LLT
     IF(  ITYPE(ITNOW).EQ.1) KK=LLT-1
     DO  30 K=1,KK
0    AGE(K)= AGE(K)+ DT

     IF SAND DEPOSITION IS OCCURRING, UPDATE THE TOTAL THICKNESS
     OF SANDS OVERLYING SHALE UNIT LL, H(LL), THE MATRIX PRESSURE
     ABOVE SHALE UNIT LL, SSIGMA(LL), AND Y(LLT).

     IF((ITY-2)*(ITY-5).NE.0) GO TO 32
     H(LL)= H(LL)+ DDZ(ITNOW)/ (1.-PORS(ITNOW) ) *NEG
     SSIGMA(LL)= SSIGMA(LL) +980.7*(RHOG(ITNOW)-RHOW)*DDZ(ITNOW)*NEG
     SMX(LLT)=DMAX1(SMX(LLT),SSIGMA(LL))
```

Main (9)

```
      EVALUATE O(LLT) WHEN SHALE,SAND,OR NOTHING IS BEING DEPOSITED.
2     IF( ITYPE(ITNOW).NE.1) YO(LLT)   = Y(LLT)
      HLL=H(LL)
      CALL ABGIT(A1,A2,A3,A4,A5,B1,B2,B3,B4,GDRHO, IT, LL)

      IF(NEG.EQ.-1)F(LLT)=A4
      YTRY=O(F(LLT),SSIGMA(LL),SMX(LLT))
      IF(YTRY.LE.YMIN(LLT))F(LLT)=1.
        Y(LLT)= O(F(LLT),SSIGMA(LL),SMX(LLT))


      MAKE INITIAL ESTIMATES OF POROSITIES  O(I) THROUGH O(LLT-1).
      AND  COORDINATE DIFFERENCES DU(I) THROUGH DU(LLT-1), PRIOR
      TO ITERATING TO FIND GOOD VALUES OF THESE  2*(LLT-1)
      QUANTITIES. DU(K)= U(LLT)- U(K). DU(1)=U(LLT). DU(LLT)=0.



      CALCULATE, FOR THE BASE OF EACH SHALE UNIT, THE CURRENT VALUE
      FOR THE DIFFERENCE
      SMPN(L)=( OVERBURDEN PRESSURE - NORMAL WATER PRESSURE ).
      THIS WILL BE CONSTANT IN THE FOLLOWING ITERATIONS AT FIXED
      TIME.
      DO   47  L=1,LL
7     IF( ITY.NE.3)
     1SMPN(L)= SMPN(L)+ 980.7*(RHOG(ITNOW)-RHOW )*DDZ(ITNOW)*NEG
      IF(LL.LT.20) SMPN(LL+1)=0.0

      CALL ESTIM8(LL,LLT,NEWSHL,NEWTYP,ITNOW,ICOUNT,IFAST,NEG)
      WRITE(6,951)
      WRITE(6,952) (AGE(II),Y(II),DU(II),II=1,LLT)
951   FORMAT(1H0,31HTHESE ARE THE INITIAL ESTIMATES)
952   FORMAT(1H0,3(G12.4,5X))
      WRITE( 6,923) NDT,ICOUNT,IFAST,NEG,(Y(I),I=1,LLT)


      TOP OF ITERATION FOR FIXED TIME AT STATEMENT 333.

      LDIAG=LOWER DIAGONALS IN D WHEN DGELB IS CALLED.
      LDIAG=MIN0(2,LLT-2)
      IF(LL.GT.1) LDIAG=3
      LUP=MIN0(3,LLT-2)
      IGOOD=0
      ITER8S = 0
```

Main (10)

```
33  CONTINUE
    ITER8S= ITER8S +1
6
    MATRIX INVERSION BY DYSINGER'S METHOD.SOLVE THE 2*(LLT-1)
    LINEAR EQUATIONS

     ( A(U,U)  B(U,O) )*( DDU ) = (-X(U) ) .
     ( C(O,U)  D(O,O) )*(  DO )   (-X(O) )

    A,B,C,D, ARE SQUARE, (LLT-1).  A*AI= I= A*A2I*A1I. A2I IS
    UPPER TRIANGULAR, FILLED WITH +1.

    COMPUTE  A1I, B , X(U), X(O). REPLACE X BY -X.

    CALL COMLZ( X(I), REST, NIHIL )

    DO  50  L=1,LL
    CALL ABGIT(A1,A2,A3,A4,A5, B1,B2,B3,B4, GDRHO, IT, L )


    LBL= LB(L)
    LBLP= LB(L)+1
    LBLM=LB(L)-1
    LTL = LT(L)
    LTLM = LT(L)-1
    LLTM = LLT-1
    LLTM2=LLT-2
    LX = MINO( LTLM, LLTM )
    LXX=LX+1
         *
    IF((Q(F(LLT),BQ,SMX(LLT))-Y(LXX)),LE. .000006) LXX=LX
    LP= L+1
    LM= L-1

    THE ABOVE WILL SIMPLIFY SUBSCRIPTS IN THIS LOOP, L=1,LL.

    DO 52  K= LBL, LX
    A1I(K)=1.-.5*(Y(K)+Y(K+1))
    X(K) = CHIU( DU(K), DU(K+1), Y(K), Y(K+1), DDZ(IT) ) /A1I(K)
    B( 2*K-1) =  (-DU(K) + DU(K+1))*.5/A1I(K)
2   IF( K.NE. LLTM )  B(2*K) = B(2*K-1 )

    PUT TOGETHER CONDITION ON  DU(LTL) AND ITS DERIVATIVES.

    WRITE( 6,922) NDT,LLT,NEWSHL,NEWTYP,(X(I),B(2*I-1),B(2*I),I=1,LLT)
    IF(L.EQ.LL) GO TO 53
    K=LTL
```

```
      X(K)=-H(L)-DU(K+1)+DU(K)
      A11(K)=+1.

      COMPUTE THE X(0) , L=1,LL.  FIRST, NODE LB(L).

3     CONTINUE
      I=LBL
      IF(IBC(IT).NE.2) GO TO 200
          X( LLTM + I   ) = SIG( Y( I   ),F(I),SMX(I)    )
     1 -(1.-ALPHA(IT))*SMPN(L) = (1.-BETA(IT))*GW*( DU(I   )+ HLL)
     2 -GAMMA(IT)
      C(I,I)=-(1.-BETA(IT))*GW
      D(I,I)=SIGO(Y(I),F(I),SMX(I))
      WRITE( 6,922) NDT,L,LL,IT,C(I,I),D(1,1)
      GO TO 55

200   ILX=MIN0(3,LX-LBL+1)
      ILZ=MIN0(I+3,LXX)
      CALL XCD(I,I,ILZ,ILX,L,ANS)
      IF(ITER8S.EQ.1.AND.I.NE.1 ) LUP=MIN0(LLTM2,MAX0(LUP,ILZ-I+1))

      IF ( IBC(IT).NE.3 ) GO TO 55
      K=LBLM
      LDIAG=4
      CALL XCD(I,K-3,K,6,LM,ANS)

      INTERNAL NODES IN UNIT L, IF ANY.

5     CONTINUE

      IF ( LBLP.GT.LX ) GO TO 101

      DO 57  K=LBLP, LX

      X(LLTM+K)= -(DU(K)-DUO(K))/DT

      C(K,K)=C(K,K)-1./DT
      IF(LBLP.EQ.LX) CALL XCD(K,K-1,LXX,4,L,ANS)
      IF(LBLP.EQ.LX) GO TO 57

      IF(K.EQ.LX) CALL XCD(K,K-2,LXX,8,L,ANS)
      M=MIN0(LXX,K+2)
      IF(K.NE.LX) CALL XCD(K,K-1,M,7,L,ANS)

57    CONTINUE
```

```
7   DO NECESSARY SUBTRACTION OF X'S BETWEEN  LT(L)-1 AND LB(L)+1.
    WRITE( 6,922) M,LX,ITNOW

    M=  LX+1
    I2 = M -LBL -2
    IF(I2.LT.1)  GO TO 102
    DO 60     I= 1,I2
    IG1=MAXO(1,M-I-4)
    IG2=MINO(LLTM,M-I+LUP)

    DO 61 J=IG1,IG2
    C(M-I,J)=C(M-I,J)-C(M-I-1,J)
1   D(M-I,J)=D(M-I,J)-D(M-I-1,J)
0   X(LLTM+M-I) = X(LLTM+M-I)- X(LLTM+M-I-1)

02  CONTINUE

    FINISH COMPUTING X FOR NODE LB(L)+I

    IF(LBLP.GE.LLT) GO TO 692

    I=LBLP
    II1=MINO(LXX,I+2)
    II2=MINO(10,8+LX-LBL)
    CALL  XCD(I,LBL,II1,II2,L,ANS)
    X(LLTM+I)=X(LLTM+I)+(DU(I-1)-DUO(I-1))/DT

    C(I,I-1)=C(I,I-1)+1./DT
92  CONTINUE

    COMPUTE X FOR POROSITY AT TOP-MOST NODE OF ALL SHALE UNITS BURIED
    BENEATH ANOTHER SHALE.

    IF(L.EQ.LL) GO TO 101

    I=LTL
    X(I+LLTM)=SIG(Y(I),F(I),SMX(I))-SSIGMA(L)
    D(I,I)=SIGO(Y(I),F(I),SMX(I))
    CALL  ABGIT(A1,A2,A3,A4,A5,B1,B2,B3,B4,GDRHO,IT,LP)
    X(I+LLTM)=X(I+LLTM)-SIG(Y(I+1),F(I+1),SMX(I+1))
    D(I,I+1)=-SIGO(Y(I+1),F(I+1),SMX(I+1))
    CALL  ABGIT(A1,A2,A3,A4,A5,B1,B2,B3,B4,GDRHO,IT,L)

    ALL  X'S HAVE BEEN COMPLETED. REPLACE X'S BY  -X'S.

01  CONTINUE
```

Main (13)

```
      IH= MIN0( LTL, LLTM )
      DO  62  I= LBL, IH
      X(I)= -X(I)
2     X( LLTM+I)  = - X( LLTM+I)

      END OF  LOOP  OVER L'S .

0     CONTINUE

      WW=0.
      DO 981 I=1,LLTM
81    WW=WW+DABS( X(I+LLTM))
      IF(WW.LT.10.**(-12.)) WW=10.**(-12.)
      WW=1.

      ELIMINATE SUB-DIAGONALS OF C .

      IF( LLTM2.EQ.0 ) GO TO 104

      II=0
      DO 64    J=  1,LLTM2
      I1 =  J+1
      I2 =  MIN0( LLTM, J+LDIAG )

      II=II+2
      DO  64    I = I1,I2
      IF( C(I,J).EQ.0.0) GO TO 64
      C(I,J+1) = C(I,J+1)  + C(I,J )
      D(I,J+1) = D(I,J+1)  - C(I,J )* B(II  )
      D(I,J )  = D(I,J )  - C(I,J )* B(II-1)
      X(LLTM+I)= X(LLTM+I) - C(I,J )* X( J )
4     CONTINUE

      ELIMINATE SUPER-DIAGONALS OF C.


      II=2*LLTM
      DO 66 JJ= 1,LLTM2

      J  = LLTM +1 -JJ
      I1 =  MAX0(  J-LUP,1)
      I2 =   J - 1
      II=II-2

      DO  66  I= I1,I2
      IF( C(I,J).EQ.0.0 )  GO TO 66
```

Main

```
&   C( I,J-1) = C(I,J-1)   + C(I,J)
&   D( I,J-1) = D(I,J-1)   + C(I,J)* B(II-1)
    D( I,J )  = D(I,J )    + C(I,J)* B(II)
    X(LLTM+I) = X(LLTM+I)+ C(I,J)* X( J-1)

6   CONTINUE
04  CONTINUE
```

Main (19)

```
    TURN C INTO IDENTITY WITH SOME ZEROS.

    DO  411    I=1,LLTM

    IF(C(I,I).EQ.0.) GO TO 411
    X(LLTM+I) = X(LLTM+I)/ C(I,I)
    J1 =  MAX0( 1, I-LDIAG )
    J2 =  MIN0( LLTM, I+LUP)

    DO  402    J= J1,J2

02  D(I,J) = D(I,J) / C(I,I)
11  CONTINUE

    REPLACE ANY 0.0 WITH 1 IN DIAGONAL OF C.

    IF(C(1,1).NE.0.) GO TO 845

    IF(LLT.NE.2) GO TO 847
    X(2)=X(2)+X(1)
    D(1,1)=D(1,1)+B(1)
    GO TO 845

47  X(LLT)=X(LLT)+X(1)+X(LLT+1)
    DO 846 I=1,LLTM
    D(1,I)=D(1,I)+D(2,I)
46  IF(I.LE.2) D(1,I)=D(1,I)+B(I)

45  CONTINUE
    IF(2.GT.LLTM) GO TO 482
    DO 483 I=2,LLTM
    IF(C(I,I).NE.0.) GO TO 483
    D(I,I-1)=D(I,I-1)-B(2*I-3)
    D(I,I)=D(I,I)-B(2*I-2)
    X(LLTM+I)=X(LLTM+I)-X(I-1)+X(LLTM+I-1)

    J1=MAX0(1,I-1-LDIAG)
```

```
      J2=MIN0(LLTM,I+2)
      DO 484 J=J1,J2
484   D(I,J)=D(I,J)+D(I-1,J)

483   CONTINUE
482   CONTINUE
      WRITE( 6,922) ITER8S

      PUT  -AD INTO C , THEN ADD B TO C TO OBTAIN (B-AD).

      CALL COMLZ(BRACK,ET,NIHIL)

      II=0
      DO  661  I=1, LLTM
      J1 =  MAX0(  I-LDIAG , 1 )
      J2 = MIN0(  I+LUP+1,LLTM)
      II=II+2
      C(I,I)=B( II-1)
      IF(I.LT.LLTM)  C(I,I+1)=B( II)

      DO  661   J= J1,J2
      C(I,J)   = C(I,J)  -D(I,J)
561   IF( I.NE.LLTM) C(I,J) = C(I,J) +D(I+1,J )


      PUT C INTO VECTOR  XX FOR DGELB.

      NU = 0

      DO  401   I= 1,LLTM
      J1 =  MAX0( 1, I-LDIAG)
      J2 = MIN0( LLTM, I+LUP)

      DO  401  J= J1,J2
      NU = NU +1
401   XX( NU) =  C(I,J)

      FORM ((-X1) -A(-X2)) AND PUT INTO XA.

      IF(  LLTM2.EQ.0 ) GO TO 403

      DO  404   I=1,LLTM2
404   XA( I) = -X(LLTM+I) + X(LLTM+I+1)  + X(I)
403   XA(LLTM) =  -X(LLTM*2) + X(LLTM)

      EPS = 10.0**(-12.)

      CALL DGELB( XA, XX, LLTM,1,  LUP,LDIAG,EPS,IER)
```

Main (16)

9

```
      XA CONTAINS THE POROSITY CORRECTIONS IN  I=1,LLTM

      IF(IER.NE.0 ) WRITE(6,910)

      PUT CORRECTIONS TO DU INTO XA, I=1,LLTM.  ((-X2)-D*XA ).

      DO    406     I=1,LLTM
406   XA( LLTM+I) =  XA(I)
      DO 405 I=1,LLTM
      J1 =  MAX0( 1, I-LDIAG )
      J2 =  MIN0( LLTM, I+LUP)
      XA(I)=X(I+LLTM)
      DO    405     J= J1,J2
405   XA(I) =  XA(I) - D(I,J)* XA(LLTM+J )

      XA CONTAINS IN SEQUENCE THE DU AND O CORRECTIONS.

      BIGU = LARGEST CORRECTION TO DU'S.  TOLU= TOLERENCE.
      BIGY = LARGEST CORRECTION TO Y'S.   TOLY= TOLERENCE.

      BIGDU=0.0
      BIGU =0.
      L = 1

      BIGY =0.
      DO   420  I= 1,LLTM
      IF(DU(I).NE.DU(I+1))
     1BIGU = DMAX1( BIGU,DABS(  XA(I)/(DU(I)-DU(I+1))  )   )
      BIGDU=DMAX1( BIGDU,DABS(XA(I)/DU(I)  ))
      BIGY = DMAX1( BIGY,DABS(  XA(I+LLTM)/Y(I)  ))


          CORRECT THE POROSITIES AND DEPTHS AT ALL NODES

      DU(I)= DU(I)+ XA(I)
      Y(I) = Y(I) + XA(I+LLTM)

      IF(I.EQ.LB(L)) CALL ABGIT(A1,A2,A3,A4,A5,B1,B2,B3,B4,GDRHO,IT,L)
      IF( I.EQ.LB(L)) L = L + 1
      IF(MOD(ITER8S,2).NE.0)GO TO 1831
      IF(Y(I).GT.YMIN(I)*1.003.AND.ITNOW.GE.ITIST)F(I)=A4
      IF(Y(I).LE.YMIN(I)*.997)F(I)=1.
      IF(F(I).EQ.1.)
     1 SMX(I) = DMAX1(SMX(I),SIG(Y(I),F(I),SMX(I)))
831   CONTINUE

20    CONTINUE
```

Main (17)

```
      WRITE( 6,922) LLTM

      IF(ITER8S. GT.37 ) WRITE(6,920) ITER8S,ITNOW,(Y(I),DU(I),I=1,LLT )
      IF(ITER8S.GT.40) GO TO 999

      WRITE( 6,922) NDT,IJ3,ITER8S,ITNOW,(Y(I),DU(I),I=1,LLT)

      JUDGE=0
      IF(BIGU.LE.TOLU.OR.BIGDU.LE.TOLDU) JUDGE=1
      IF(JUDGE.NE.1.OR.BIGY.GT.TOLY) GO TO 333
      IGOOD=IGOOD+1
      IGD=1
      IF(ICOUNT.NE.0) IGD=4
      IF(IGOOD.LT.IGD) GO TO 333

      CALL TIMINT(IJ2)
      IJ3=IJ1-IJ2
      WRITE( 6,924) IJ3,ITER8S,ITNOW,NDT
      WRITE(6,922) IJ3 ,ITER8S

      CALL OUTPUT(LLT,LL,IFIRST,DT,TIME,ITNOW,NDT,ICOUNT)

      DO 70 I=1,LLT
      SMX(I) = DMAX1(SMX(I),SIG(Y(I),F(I),SMX(I)))
70    YMIN(I)=DMIN1(Y(I),YMIN(I))

      GO TO 777

          *READ* FORMAT STATEMENTS

900   FORMAT ( 20A4 / 3F10.4 / 4(5F10.0/ ) ,  3(4F10.0/ ), 4F10.0    )

901   FORMAT ( 2I1, 2I3, 2F6.0, 6F4.0,  2I1, 4F5.0 ,6I1   )

909   FORMAT (A1,I2/  20F4.0 /   20F4.0 )

902   FORMAT (  9(I1,5X)/  9(I3,3X) / 9(F6.0) /  3(9(F4.0,2X) / ) ,
     1 9( I1,5X) /  2( 9(F5.0, 1X) / ) ,  2( 9(I1,5X)/ ),
     2 9( I1,5X)  )

          *WRITE* FORMAT STATEMENTS
```

Main 18

```
0003  FORMAT( '2',    T62,'INPUT DATA' /  T27,20A4 // T41,
     1'EXTRAPOLATED SEA FLOOR TEMPERATURE (DEG F)', T87, F8.1  /
     2 T41,'GEOTHERMAL GRADIENT (DEG F / FT )',  T87, F8.4 /
     3 T41,'SODIUM CHLORIDE IN PORE WATER (WEIGHT % )', T87, F8.2 //
     4 T6 ,'CONSTANTS IN POROSITY - MATRIX STRESS - SEDIMENT AGE '  ,
     5  'RELATIONSHIP' ,T80, 'CONSTANTS IN PERMEABILITY - POROSITY '  ,
     6  'RELATIONSHIP'  /    T15, 'A1', T27,'A2', T39,'A3', T51,'A4'  ,
     7 T63, 'A5', T90,'B1', T102,'B2',  T114,'B3', T126, 'B4' /
     8 T6, 'SET 1 ',  5( G11.5,1X ) , T80, 'SET 1', 4(1X, G11.5    ) /
     9 T6, 'SET 2 ',  5( G11.5,1X ) , T80, 'SET 2', 4(1X, G11.5    ) /
     1 T6, 'SET 3 ',  5( G11.5,1X ) , T80, 'SET 3', 4(1X, G11.5    ) /
     2 T6, 'SET 4 ',  5( G11.5,1X ) , T80, 'SET 4', 4(1X, G11.5    )  )


0005  FORMAT ('0' , T40, 'EACH COLUMN OF TWELVE ITEMS DESCRIBES A ',
     1 'TIME INTERVAL'  /  T10, 'INTERVAL NUMBER= ',
     2  9(I1,10X ),  I2    )

0006  FORMAT ('0' ,T9 , '  INTERVAL NUMBER= ',  9( I2,9X), I2    )


0007  FORMAT ( ' ' , T6  ,'TIME INTERVAL TYPE' ,10( 4X,I1,6X) /
     2 T11,'SUB-INTERVALS'  ,    10(2X,I3, 6X)   /
     3 T3 ,'SUB-INTERVAL DURATION' ,10( G11.5)    /
     4 T2 ,'SUB-INTERVAL THICKNESS',10( 1X, G10.3)  /
     5 T11,'GRAIN DENSITY'  ,    10( 1X ,G10.3    ) /
     6 T5 ,'   PERCENT POROSITY',  10( 1X ,G10.3    ) /
     7 T6 ,'BOUNDARY CONDITION' ,  10( 4X,I1,6X)    /
     8 T19,'ALPHA' ,    10(1X,  G10.4    ) /
     9 T20,'BETA'  ,    10(1X,  G10.4    ) /
     9 T2 ,'POROSITY PARAMETER SET',10( 4X,I1,6X)   /
     1 T2 ,'PERMEABILITY PARA. SET',10( 4X,I1,6X)   /
     2 T5 ,'PRINT-OUT FREQUENCY' ,  10( 4X,I1,6X) )

0014  FORMAT ('1' ,T17,'PRESSURE DECREASES (FIRST LINE) AND ',
     1  'CORRESPONDING TIME INCREMENTS (SECOND LINE) JUST BELOW '/
     3 T7 ,10G10.1/T8,10G10.2///T7,10G10.1/T8,10G10.2)


0010  FORMAT (///// T51,'**** WARNING - MATRIX **C** WAS SINGULAR ****'
```

```
     1/////)

920  FORMAT ('1' , T20 ,'NO CONVERGENCE AFTER ', I2 ,' ITERATIONS.' /
     1 T20 , 'ITNOW= ', I2  /   T20,'POROSITY    DU' /
     2 ( T20 , G11.4 ,2X , G11.4   )   )

922  FORMAT(' ', 4I10/ 5(G12.4,2X))
923  FORMAT(' ',//4I10/5(G12.4,2X))
924  FORMAT(' ',6I10)

999  CALL EXIT
     STOP
     END
```

Main 20

```
CT*  NOTERM,NOID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP,NOTEST
CT*  NAME = MAIN    , LINECNT =      50
SOURCE STATEMENTS =      382,PROGRAM SIZE =    19854
 DIAGNOSTICS GENERATED
```

Main 21

```
C
C  OBTAIN PARAMETERS FOR POROSITY - MATRIX PRESSURE -AGE , AND
C  PERMEABILITY - POROSITY EQUATIONS FOR SHALE UNIT L. ALSO
C  OBTAIN  G*( RHOG(L)- RHOW ) AND IT, GIVEN L ONLY.
C
      SUBROUTINE ABGIT(A1,A2,A3,A4,A5,B1,B2,B3,B4,GDRHO, IT, L)
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON/ERODE/ LNOW(20),ITNOW
C
      COMMON  / READIN / CASEID(20), TEMPT, GRADT, WTPCT, AA(4,5),
     1 BB(4,4), ITYPE(20), NSUBI(20),DDT(20), DDZ(20), RHOG(20),
     2 PORS(20), IBC(20), ALPHA(20), BETA(20), ISETA(20), ISETB(20),
     3 IPAGE(20),      DP(20), DIT(20), GAMMA(20),RHOW, IZ,ITMAX
C
C  FIND IT, GIVEN L, ITNOW, AND LNOW.
C
      DO 1 I=1,ITNOW
      IT=ITNOW+1-I
      IF(LNOW(IT).EQ.L) GO TO 4
    1 CONTINUE
    4 CONTINUE
C
      M = ISETA(IT)
      N = ISETB(IT)
C
      A1=   AA( M,1)
      A2=   AA( M,2)
      A3=   AA( M,3)
      A4=   AA( M,4)
      A5=   AA( M,5)
C
      B1=   BB( N,1)
      B2=   BB( N,2)
      B3=   BB( N,3)
      B4=   BB( N,4)
C
      GDRHO = 980.7*( RHOG(IT) - RHOW )
C
      RETURN
      END
```

ABGIT

13

```
      SUBROUTINE  DEPLET( DT,  ICOUNT,  * )
      IMPLICIT REAL*8 (A-H,O-Z)

      COMMON / READIN / CASEID(20), TEMPT, GRADT, WTPCT, AA(4,5),
     1 BB(4,4), ITYPE(20), NSUBI(20), DDT(20), DDZ(20), RHOG(20),
     2 PORS(20), IBC(20), ALPHA(20), BETA(20), ISETA(20), ISETB(20),
     3 IPAGE(20),      DP(20), DIT(20),GAMMA(20),RHOW, IZ,ITMAX

      COMMON/ERODE/ LNOW(20), ITNOW

      COMMON / ABC / DU(129), DUO(129), DUOO(129), YO(129), AGE(129),
     1    Y(129), YOO(129),YMIN(129),F(129),SMX(129),
     1 LB(20), LT(20), NDTSUM(20),   H(20),  SSIGMA(20),
     2 SMPN(20),GDRHO,LLTM,LO,HLL , GW

      F=DECOMPACTION PARAMETER
      A5=SKEMPTON'S LIQUID LIMIT
      PI(A5)=-12.39+.774*A5
      PL(A5)=A5-PI(A5)

      BSIG(F)=DEXP((3.07+(.23*A5+12.4)/(-12.4+.77*A5))/.175)+F-F

      ALPHX(F,SMX)=(.23*A5+12.4)+(-12.4+.77*A5)*(3.07-.175*DLOG(BSIG(F
     1))-.175*(1.-F)*DLOG(1.-DEXP(-(SMX+100.)/BSIG(F))))
      BETX(F)=.175*F*(-12.4+.77*A5)

      QIN(PO,F,SMX)=DEXP((ALPHX(F,SMX)-PO/.027/(1.-PO))/BETX(F))

      SIG(PO,F,SMX)=-100.-BSIG(F)*DLOG(1.-QIN(PO,F,SMX))

      IF(  ICOUNT.NE. 0 )  GO TO 10

      IBC(IZ) = 2
      BETA(IZ)= 0.0

      L=LNOW(IZ)

      CALL  ABGIT( A1,A2,A3,A4,A5, B1,B2,B3,B4, GDRHO, IK, L )

      COMPUTE ALPHA(IK) JUST ONCE.
```

```
ASE 2.0                ABGIT              DATE = 84355          14/52/28          PAGE 0002

T*  NOTERM,NOID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP,NOTEST
T*  NAME = ABGIT    , LINECNT =       50
OURCE STATEMENTS =        23,PROGRAM SIZE =       900
DIAGNOSTICS GENERATED
```

ABGIT 2

13

```
      *
      ALPHA(IK)=  1.-  1./ SMPN(L)*(  SIG(Y(LB(L)),F(LB(L)),SMX(LB(L)))
     1                              - GW*( DU(LB(L))+ HLL) )

      ITYPE(ITMAX) =3
      IPAGE(ITMAX) =2

0     CONTINUE

      ICOUNT  = ICOUNT +1
      IF(ICOUNT.GT.20) RETURN 1
      DT   =  DIT(ICOUNT)
      IF(DT.EQ.0.0) RETURN 1
      GAMMA(IK)=GAMMA(IK)+DP(ICOUNT)

      RETURN
      END
```

Deplet [redacted] 2

```
*  NOTERM,NOID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP,NOTEST
*  NAME = DEPLET  , LINECNT =     50
URCE STATEMENTS =       26,PROGRAM SIZE =      2126
IAGNOSTICS GENERATED
```

Deplet [3]

14

```
      SUBROUTINE ESTIM8(LL,LLT,NEWSHL,NEWTYP,ITNOW,ICOUNT,IFAST,NEG)

      IMPLICIT REAL*8 (A-H,O-Z)

      COMMON / READIN / CASEID(20), TEMPT , GRADT, WTPCT, AA(4,5),
     1BB(4,4), ITYPE(20), NSUBI(20),DDT(20),DDZ(20), RHOG(20),
     2PORS(20),IBC(20), ALPHA(20), BETA(20), ISETA(20), ISETB(20),
     3IPAGE(20),      DP(20),DIT(20),GAMMA(20),RHOW, IZ,ITMAX


      COMMON/ABC/DU(129),DUO(129),DUOO(129), YO(129),    AGE(129),
     1    Y(129), YOO(129),YMIN(129),F(129),SMX(129),
     1           LB(20), LT(20), NDTSUM(20), H(20), SSIGMA(20),
     2SMPN(20),GDRHO,LLTM,LO, HLL, GW
           *
      PERM(PO)=PO**B1*DEXP(B2+B3*PO)/(1.-PO)**B4

      F=DECOMPACTION PARAMETER
      A5=SKEMPTON'S LIQUID LIMIT
      PI(A5)=-12.39+.774*A5
      PL(A5)=A5-PI(A5)

      BSIG(F)=DEXP((3.07+(.23*A5+12.4)/(-12.4+.77*A5))/.175)+F-F

      ALPHX(F,SMX)=(.23*A5+12.4)+(-12.4+.77*A5)*(3.07-.175*DLOG(BSIG(F
     1))-.175*(1.-F)*DLOG(1.-DEXP(-(SMX+100.)/BSIG(F))))
      BETX(F)=.175*F*(-12.4+.77*A5)

      W(F,SIG,SMX)=-BETX(F)*DLOG(-DEXP((-SIG-100.)/BSIG(F))+1.)+ALPHX
     1(F,SMX)
      O(F,SIG,SMX)=.027*W(F,SIG,SMX)/(1+.027*W(F,SIG,SMX))
      QIN(PO,F,SMX)=DEXP((ALPHX(F,SMX)-PO/.027/(1.-PO))/BETX(F))

      SIG(PO,F,SMX)=-100.-BSIG(F)*DLOG(1.-QIN(PO,F,SMX))


      BQ=0.
      L1=LL

      LLTM=LLT-1
      LLTM2=LLT-2
      LLTM3=LLT-3
      LLM=LL-1
      IFAST=IFAST+1
      IF(ICOUNT.NE.0) IFAST=1
```

Estimate-1

Estimator 2

```
      IF(NEWTYP.EQ.1) IFAST=1

      LTOP=LL
      KTOP=LLT-1

      DUOO(LLT)=DUO(LLT)
      DUO(LLT)=DU(LLT)
      DU(LLT)=0.0

      IF(ITYPE(ITNOW).NE.1.OR.IFAST.GT.2) GO TO 1

      IF(IFAST.GT.1) GO TO 6
      YO(LLTM)=Y(LLT)
      DUO(LLTM)=0.0
      CALL ABGIT(A1,A2,A3,A4,A5,B1,B2,B3,B4,GDRHO,IT,LL)
      BURDEN=GDRHO*DDZ(ITNOW)
      Y(LLTM)=     O(F(LLTM),BURDEN,BURDEN)
      IF(LL.EQ.1) GO TO 111
      IF(SSIGMA(LLM).NE.0.) GO TO 111
      FUDG=DDZ(IT)/DDT(IT)/PERM(O(F(LLT),BQ,SMX(LLT)))
      CALL ABGIT(A1,A2,A3,A4,A5,B1,B2,B3,B4,GDRHO,IT,LLM)
      FUDG=FUDG/DDZ(IT)*DDT(IT)*PERM(O(F(LLTM2),BURDEN,SMX(LLTM2)))
      OBTAIN YO(LLT-3)/YNO(LLT-3)
      BURDEN=SSIGMA(LLM)+GDRHO*DDZ(IT)
      YNOM3=O(F(LLTM3),BURDEN,SMX(LLTM3))
      Y(LLTM)=Y(LLTM)*(1.+(Y(LLT-3)/YNOM3-1.)*FUDG)
      CALL ABGIT(A1,A2,A3,A4,A5,B1,B2,B3,B4,GDRHO,IT,LL)
11    DU(LLTM)=DDZ(ITNOW)/(1.-(Y(LLT)+Y(LLTM))/2.)
      IF(IBC(IT).EQ.2) CALL BC2(LL,LLTM)

      IF(LL.EQ.1) RETURN

      BURDEN=SSIGMA(LLM)+SIG(Y(LLTM),F(LLTM),SMX(LLTM))
      CALL ABGIT(A1,A2,A3,A4,A5,B1,B2,B3,B4,GDRHO,IT,LLM)

      YOO(LLTM2)=YO(LLTM2)
      YO(LLTM2)=Y(LLTM2)
      Y(LLTM2)=O(F(LLTM2),BURDEN,SMX(LLTM2))

      DUO(LLTM2)=0.0
      DU(LLTM2)=H(LLM)+DU(LLTM)
      LTOP=LLM
      KTOP=LLT-3
      GO TO 1

5     CONTINUE

      YO(LLTM)=Y(LLTM)
```

```
      Y(LLTM)=Y(LLTM2)
      DUO(LLTM)=DU(LLTM)
      DU(LLTM)=DU(LLTM2)

      CALL ABGIT(A1,A2,A3,A4,A5,B1,B2,B3,B4,GDRHO,IT,LL)
      BURDEN= SIG(Y(LLTM),F(LLTM),SMX(LLTM))
      BURD2=2.*BURDEN
      S1=GDRHO*DDZ(IT)
      PN1=O(F(LLTM),S1,SMX(LLTM))
      PN2=O(F(LLTM2),2.*S1,SMX(LLTM2))
      YOO(LLTM2)=YO(LLTM2)
      YO(LLTM2)=Y(LLTM2)
      Y(LLTM2)=     O(F(LLTM2),BURD2,SMX(LLTM2))
      Y(LLTM2)=PN2*Y(LLTM)/PN1
      DUO(LLTM2)= DU(LLTM2)
      DU(LLTM2)=DU(LLTM)+DDZ(ITNOW)/(1.-(Y(LLTM)+Y(LLTM2))/2.)
      IF(IBC(IT).EQ.2) CALL BC2(LL,LLTM2)
      IF(LL.EQ.1) RETURN
      CALL ABGIT(A1,A2,A3,A4,A5,B1,B2,B3,B4,GDRHO,IT,LLM)
      YOO(LLTM3)=YO(LLTM3)
      YO(LLTM3)=Y(LLTM3)
      BURDEN=SSIGMA(LLM)+SIG(Y(LLTM2),F(LLTM2),SMX(LLTM2))
      Y(LLTM3)=O(F(LLTM3),BURDEN,SMX(LLTM3))
      DUOO(LLTM3)=DUO(LLTM3)
      DUO(LLTM3)=DU(LLTM3)
      DU(LLTM3)=H(LLM)+DU(LLTM2)

      LTOP=LLM
      KTOP=LLT-4

      FRAC=1.

      CONTINUE
      L1=LTOP

      DO 2 I=1,KTOP
      K=KTOP+1-I

      SAVEY=Y(K)
      IF(IFAST.LE.3.AND.ITNOW.GT.1) GO TO 77
      IF(ITYPE(ITNOW).EQ.4) GO TO 77
      IF(ICOUNT.NE.0) GO TO 77
      IBB=0
      IF(K.GT.LLT-4.AND.ITYPE(ITNOW).EQ.1.AND.K.GT.LB(LL)) IBB=1
      IF(IBB.EQ.1)   Y(K)=Y(K)+     (Y(K-1)-Y(K))
      IF(  IBB.NE.1.AND.IFAST.LE.3)     Y(K)=Y(K)+     (Y(K)-YO(K))
      IF(IBB.NE.1.AND.IFAST.GE.4) Y(K)=(3.*(Y(K)-YO(K))+YOO(K) )
     1 /2.+(2.*Y(K)-YO(K))/2.
```

```
      GO TO 78
7     DTIM=DDT(ITNOW)
      IF(ICOUNT.NE.0) DTIM=DIT(ICOUNT)
      IF(K.EQ.KTOP) GO TO 50
      IF((K-LB(L1))*(LT(L1)-K).GE.0) GO TO 44
      L1=L1-1
0     CALL ABGIT(A1,A2,A3,A4,A5,B1,B2,B3,B4,GDRHO,IT,L1)
4     CONTINUE
      SO=SIG(Y(K),F(K),SMX(K))
      IF(ITYPE(ITNOW).EQ.3.OR.ICOUNT.NE.0) GO TO 79
      SO=SO+980.7*(RHOG(ITNOW)-RHOW)*DDZ(ITNOW) *NEG
9     Y(K)=O(F(K),SO,SMX(K))
8     CONTINUE
      YOO(K)=YO(K)
      YO(K)=SAVEY

      IF(IFAST*ITYPE(ITNOW).EQ.1.AND.LLT.GT.2)
     1  Y(LLT-3)=0.5*(Y(LLT-2)+Y(LLT-4))

      FIND VALUES FOR DU(K) AND DUO(K) .

      CALL ABGIT(A1,A2,A3,A4,A5,B1,B2,B3,B4,GDRHO,IT,LTOP)

      DO 3  I=1,KTOP
      K=KTOP+1-I
      HL=0.
      IF(LTOP.EQ.1) GO TO 4
      IF(K.NE.LT(LTOP-1)) GO TO 4
      LTOP=LTOP-1
      CALL ABGIT(A1,A2,A3,A4,A5,B1,B2,B3,B4,GDRHO,IT,LTOP)
      HL=H(LTOP)
      CONTINUE

      SAVEU=DU(K)

      IF(K.EQ.LT(LTOP)) DU(K)=DU(K+1)+HL
      IF(K.NE.LT(LTOP)) DU(K) = DU(K+1)+DDZ(IT)/
     1  (1.-(Y(K)+Y(K+1))/2.)
      IF(K.NE.LB(LTOP).OR.IBC(IT  ).NE.2) GO TO 88
      BU=SMPN(LTOP)*(1.-ALPHA(IT))+GW*(DU(K)+HLL)*(1.-BETA(IT))+
     1GAMMA(IT)
      Y(K)=O(F(K),BU,SMX(K))
      CALL BC2(LTOP,LB(LTOP))
      IF(K.NE.1) CALL ABGIT(A1,A2,A3,A4,A5,B1,B2,B3,B4,GDRHO,IT,LTOP-1)
      IF(K.NE.1)Y(K-1)=O(F(K-1),SIG(Y(K),F(K),SMX(K))+SSIGMA(LTOP-1),SMX
     1(K-1))

38    DUOO(K)=DUO(K)
```

Estimate 4

```
3 16  DUO(K)=SAVEU

      RETURN
      END
```

Estimates

```
:T*  NOTERM,NOID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP,NOTEST
:T*  NAME = ESTIM8   , LINECNT =      50
OURCE STATEMENTS =      133,PROGRAM SIZE =     7092
 DIAGNOSTICS GENERATED
```

Estimate 6

17

```
      COMPUTES UPWARD VOLUME FLOW AT NODE K AND SHALE UNIT L
      AND PUTS THE RESULT INTO W(1), ANS, AND X(LLTM+K). NNEQ
      DEFINES THE EXPRESSION FOR THE FLOW WHICH IS USED IN XCD.
      IF KF IS NOT ZERO, DERIVATIVES WITH RESPECT TO DU(KI)
      THROUGH DU(KF) ARE ADDED TO C(K,KI), ETC. AND DERIVATIVES
      WITH RESPECT TO Y(KI) THROUGH Y(KF) ARE ADDED TO D(K,KI),
      ETC.
```

XCD1

```
      SUBROUTINE XCD(K,KI,KF,NNEQ,L,ANS)
      IMPLICIT REAL*8 (A-H,O-Z)
      ------------------------------------------------------------

      DIMENSION W(2) ,FXO(129),FXU(129)

      COMMON / READIN / CASEID(20), TEMPT , GRADT, WTPCT, AA(4,5),
     1BB(4,4), ITYPE(20), NSUBI(20),DDT(20),DDZ(20), RHOG(20),
     2PORS(20),IBC(20), ALPHA(20), BETA(20), ISETA(20), ISETB(20),
     3IPAGE(20),     DP(20),DIT(20),GAMMA(20),RHOW, IZ,ITMAX

      COMMON/ SPACES/ X(258), A1I(129), B(258), XA(258),BRACK,
     1 C(129,129), ET, D(129,129), XX(1160),
     1 C( 50, 50), ET, D( 50, 50), XX(1160),
     2REST,WW ,XAO(258)

      COMMON/XXCD/ ITER8S,NEWTYP,NMAX
      COMMON/ERR/NDT

      COMMON/ABC/DU(129),DUO(129),DUOO(129), YO(129),   AGE(129),
     1    Y(129), YOO(129),YMIN(129),FR(129),SMX(129),
     1    LB(20), LT(20), NDTSUM(20), H(20), SSIGMA(20),
     2SMPN(20),GDRHO,LLTM,LO, HLL, GW


      FR=DECOMPACTION PARAMETER
      A5=SKEMPTON'S LIQUID LIMIT
      PI(A5)=-12.39+.774*A5
      PL(A5)=A5-PI(A5)

      BSIG(FR)=DEXP((3.07+(.23*A5+12.4)/(-12.4+.77*A5))/.175)-FR+FR

      ALPHX(FR,SMX)=(.23*A5+12.4)+(-12.4+.77*A5)*(3.07-.175*DLOG(BSIG
     &(FR))-.175*(1.-FR)*DLOG(1.-DEXP(-(SMX+100.)/BSIG(FR))))
```

XCD 2

```
      BETX(FR)=.175*FR*(-12.4+.77*A5)

      WC(FR,SIG,SMX)=-BETX(FR)*DLOG(-DEXP((-SIG-100.)/BSIG(FR))+1.)+ALPH
     &X(FR,SMX)

      O(FR,SIG,SMX)=.027*WC(FR,SIG,SMX)/(1+.027*WC(FR,SIG,SMX))

      QIN(PO,FR,SMX)=DEXP((ALPHX(FR,SMX)-PO/.027/(1.-PO))/BETX(FR))

      SIG(PO,FR,SMX)=-100.-BSIG(FR)*DLOG(1.-QIN(PO,FR,SMX))


          **     PERMEABILITY
      PERM(PO ) = PO**B1*DEXP( B2+B3*PO)/(1.-PO)**B4   /WW


      COMPUTE MOLES NACL PER 1000 G OF WATER. CALL IT SALT. WEIGHT PER
      CENT = WTPCT.
      SALT (WTPCT)= WTPCT/ (1.-0.01*WTPCT)/ 5.8454

      COMPUTE  TEMPERATURE , TDC, AND REDUCED VISCOSITY, VR.
      THIS IS A SALINITY CORRECTION GIVEN IN OSW R&D REPORT NO. 384.
      GRADT= DEG C /CM AND DEPTH= CM.
      TDC(DK )    = TEMPT +GRADT*(DK+HLL)

      X WILL STAND IN FOR WTPCT.

      AAA(X)= SALT(X)*(.0355 +.00231*SALT(X)- .00003*SALT(X)*SALT(X) )
      BBB(X)= SALT(X)*(-.04753 + .01598*SALT(X)-.00194*SALT(X)*SALT(X))



      COMPUTE VISCOSITY, V      , IN  POISE, OR DYNE-SEC/ CMSQ. INPUT
      VARIABLES ARE DEPTH BELOW SEDIMENT-WATER INTERFACE, DK+HLL,
      IN CM, AND  WEIGHT % NACL IN PORE WATER, X.  LANGE, P-1632.
      GLASSTONE, P-95.

      V(DK,X)= 10.**( AAA(X)+(1.+BBB(X))*(-1.64779+262.37/
     1( 139.18+ TDC( DK ) ) )  )  * 0.01

      LOGARITHMIC DERIVATIVE WRT DK. UNITS ARE DYNE-SEC/CC. GRADT
      HAS UNITS  DEG C / CM.
```

18

XCD3

```
      FZ(DK, DK, WTPCT)    = PERM(DK)/ V(DK,WTPCT)

      NUMBERS GO FROM BOTTOM TO TOP OF SHALES IN FOLLOWING FUNCTIONS.
            1
      QU1(D1,D2,O1,O2,T1,T2,S1,S2 ,WTPCT )= FZ(D1,O1,WTPCT)*( GDRHO*
     1 (1.-0.5*(O2+O1)) + ( SIG(O2,T2,S2) -SIG(O1,T1,S1))/ (D1-D2)  )
            2
      QU2(D1,D2,D3,O1,O2,O3,T1,T2,T3,S1,S2,S3,WTPCT)=FZ(D1,O1,WTPCT)*(
     1      (( SIG(O2,T2,S2)- SIG(O1,T1,S1) )*(D1-D3)**2 - (
     2 SIG(O3,T3,S3)-SIG(O1,T1,S1))*(D1-D2)**2)/(D1-D2)/(D1-D3)/(D2-D3
     3 )-GDRHO*(1.-.5*(O1+O2))*((D1-D2)*(D1-D2)/(D1-D3)/(D2-D3)-
     4 (D1-D3)/(D2-D3))-GDRHO*(1.-.5*(O2+O3))*(D1-D2)/(D1-D3))

            4
      QC2(D1,D2,D3,O1,O2,O3,T1,T2,T3,S1,S2,S3,WTPCT)=FZ(D2,O2,WTPCT)*(
     1       (  (SIG(O3,T3,S3)-SIG(O2,T2,S2))*(D1-D2)**2 -
     2 (SIG(O1,T1,S1)- SIG(O2,T2,S2))*( D2-D3)**2  )/ (D2-D3)/ (D2-D1)/
     3 (D3-D1) +GDRHO/ (D1-D3)*((1.-.5*(O2+O3))*(D1-D2)+
     4 (1.-.5*(O1+O2))*(D2-D3)) )


      BRU2( D1,D3,D4) =      (D1-D3)*(D1-D4)*(D3-D4)
      BRU3( D1,D2,D4) = BRU2( D1,D2,D4)
      BRU4( D1,D2,D3) = BRU2( D1,D2,D3)
      BRU ( D1,D2,D3,D4 ) =  BRU2(D1,D3,D4) -BRU3(D1,D2,D4)
     1    +BRU4(D1,D2,D3)

            3
      QU3(D1,D2,D3,D4,O1,O2,O3,O4,T1,T2,T3,T4,S1,S2,S3,S4,WTPCT)=
     1FZ( D1,O1,WTPCT)    / BRU( D1,D2,D3,D4)* (
     2BRU2(D1,D3,D4)*(SIG(O2,T2,S2)-SIG(O1,T1,S1)-(1.-.5*(O2+O1))*(D2-D1
     2 )*GDRHO)/(D1-D2)
     3-BRU3(D1,D2,D4)*(SIG(O3,T3,S3)-SIG(O1,T1,S1)-(1.-.5*(O2+O1))*(D2-D
     3 1)*GDRHO-(1.-.5*(O2+O3))*(D3-D2)*GDRHO)/(D1-D3)
     4+BRU4(D1,D2,D3)*(SIG(O4,T4,S4)-SIG(O1,T1,S1)-(1.-.5*(O2+O1))*(D2-D
     4 1)*GDRHO-(1.-.5*(O2+O3))*(D3-D2)*GDRHO-(1.-.5*(O3+O4))*(D4-D3)
     5 *GDRHO)/(D1-D4))


      BRC1(D2,D3,D4) = (D2-D3)*(D2-D4)*(D3-D4)
      BRC3(D1,D2,D4) = BRC1( D1,D2,D4 )
      BRC4(D1,D2,D3) = BRC1( D1,D2,D3 )
      BRC( D1,D2,D3,D4) = - BRC1(D2,D3,D4) -BRC3(D1,D2,D4)
     1   +BRC4(D1,D2,D3)
```

```
      7
     QCU3(D1,D2,D3,D4,O1,O2,O3,O4,T1,T2,T3,T4,S1,S2,S3,S4,WTPCT)=
    1 FZ( D2,O2,WTPCT)     /BRC(D1,D2,D3,D4)*(
    2-BRC1(D2,D3,D4)*(SIG(O2,T2,S2)-SIG(O1,T1,S1)-(1.-.5*(O1+O2))*(D2-D
    21)*GDRHO) /(D1-D2)
    3-BRC3(D1,D2,D4)*(SIG(O3,T3,S3)-SIG(O2,T2,S2)-(1.-.5*(O2+O3))*(D3-D
    32)*GDRHO) /(D2-D3)
    4+BRC4(D1,D2,O3)*(SIG(O4,T4,S4)-SIG(O2,T2,S2)-(1.-.5*(O3+O2))*(D3-D
    42)*GDRHO-(1.-.5*(O3+O4))*(D4-D3)*GDRHO) /(D2-D4))

      ALTERNATE SIGN OF DIFFERENTIALS

      SIGN=1.
      IF(MOD(ITER8S,2).EQ.0) SIGN=-1.

      FXU AND FXO ADJUST THE SIZE OF THE DIFFERENTIALS

      IF(ITER8S*NEWTYP.NE.1.AND.MOD(NDT,5 ).NE.0) GO TO 440
      IF(NDT.EQ.NDTO.AND.NDT.GT.2) GO TO 440
      NDTO=NDT
      DO 441 I=1,NMAX
      FXO(I)=1.
.41   FXU(I)=1.
.40   CONTINUE

      NEQ=NNEQ
      M=1
      DIFDU=( DU(KI)-DU(KI+1) ) * (-0.01     )
      IX =  2*( KF-KI+1) + 1
      IF(KF.EQ.0) IX=1
      IF(L.EQ.LY.AND.LLTM.NE.1) GO TO 500
      LY=L
       CALL ABGIT(A1,A2,A3,A4,A5,B1,B2,B3,B4,GDRHO,IT,L)

500   CONTINUE

      NS=1
      IF(NEQ.EQ.8) NS=-1
      IF(NEQ.EQ.8.OR.NEQ.EQ.7) NEQ=5
      KO=K-1*NS
      K1=K
      K2=K+1*NS
      K3=K+2*NS
      K4=K+3
      SYN=1.0
      IF(NEQ.EQ.6.OR.NEQ.EQ.9.OR.NEQ.EQ.10) SYN=-1.
      IF(NEQ.LE.8) GO TO 50
      K1=K-1
```

19

```
      K2=K
      K3=K+1
      K4=K+2
      NEQ=NEQ-7
50    IF(NEQ.NE.6) GO TO 55
      K1=K-1
      K2=K-2
      K3=K-3
      K4=K-4
      NEQ=3
55    CONTINUE
      DO 100  I=1,IX

      IF(I.EQ.1)  GO TO(1,2,3,4,7) ,NEQ
      M=2

      IF( I.GT.(KF-KI+2) )  GO TO 200
      KD =  KI + ( I-2)
      IP = 1
330   DIFU=DIFDU*FXU(KD)*SIGN
      DU(KD) =  DU(KD) + DIFU

200   IF( I.LE.(KF-KI+2) )  GO TO(1,2,3,4,7), NEQ

      KD=  KI + I - (KF-KI+3 )
      IP = 2
      IF((KD+1).LE.KF.OR.KD.EQ.KI)  VT=-0.001*(Y(KD)-Y(KD+1))
      IF(KD.EQ.KF.AND.KD.NE.KI)  VT=-0.001*(Y(KD)-Y(KD-1))
      IF(KD.EQ.KI)  VTO=0.
      IF(DABS(VT).GE.DABS(VTO))  VF=VT
      IF(VF.NE.VT)  VF=-VTO
      VTO=VT
150   VFF=VF*FXO(KD)*SIGN
      Y(KD) = Y(KD) + VFF

      GO TO (1,2,3,4,7), NEQ


      CONTINUE
      W(M)=QU1(DU(K),DU(K+1),Y(K),Y(K+1),FR(K),FR(K+1),SMX(K),SMX(K+1),
     1 WTPCT)
      GO TO 300
      CONTINUE
      C1=FR(K1)
      C2=FR(K2)
      C3=FR(K3)
      C4=FR(K4)
```

```
      W(M) =QU2( DU(K1),DU(K2 ), DU(K3 ),Y(K1),Y(K2 ),Y(K3 ),
     1 C1,C2,C3,SMX(K1),SMX(K2),SMX(K3), WTPCT ) *SYN
      GO TO 300

      CONTINUE
      C1=FR(K-1)
      C2=FR(K)
      C3=FR(K+1)
      W(M) = QC2( DU(K-1),DU(K),DU(K+1), Y(K-1),Y(K),Y(K+1), C1,
     1 C2,C3,SMX(K-1),SMX(K),SMX(K+1), WTPCT )
      GO TO 300

    3 D1=DU(K1)
      D2=DU(K2)
      D3=DU(K3)
      D4=DU(K4)
      Y1=Y(K1)
      Y2=Y(K2)
      Y3=Y(K3)
      Y4=Y(K4)
      C1=FR(K1)
      C2=FR(K2)
      C3=FR(K3)
      C4=FR(K4)
962   CONTINUE
      W(M)=QU3(D1,D2,D3,D4,Y1,Y2,Y3,Y4,C1,C2,C3,C4,
     1SMX(K1),SMX(K2),SMX(K3),SMX(K4),WTPCT)*SYN
900   CONTINUE
      GO TO 300


7     D1=DU(K0)
      D2=DU(K1)
      D3=DU(K2)
      D4=DU(K3)
      Y1=Y(K0)
      Y2=Y(K1)
      Y3=Y(K2)
      Y4=Y(K3)
      C1=FR(K0)
      C2=FR(K1)
      C3=FR(K2)
      C4=FR(K3)
985   CONTINUE
      W(M)=QCU3(D1,D2,D3,D4,Y1,Y2,Y3,Y4,C1,C2,C3,C4,
     1SMX(K0),SMX(K1),SMX(K2),SMX(K3),WTPCT)
```

XCD6

30

```
300   CONTINUE
      IF(I.NE.1) GO TO 310
      AW1=DABS(W(1))
      GO TO 100
310   DIF=W(2)-W(1)
      AW2=DABS(W(2))
      IF(IP.NE.1) GO TO 400

      IF(DABS(DIF).GT.DMAX1(AW1,AW2)*10.**(-6.)) GO TO 320
      FXU(KD)=FXU(KD)*3.
      GO TO 330
320   C(K,KD)=C(K,KD)+DIF/DIFU
      DU(KD)=DU(KD)-DIFU
      GO TO 100

400   IF(DABS(DIF).GT.DMAX1(AW1,AW2)*10.**(-6.)) GO TO 340
      FXD(KD)=FXD(KD)*3.
      GO TO 350
340   D(K,KD)=D(K,KD)+DIF/VFF
      Y(KD)=Y(KD)-VFF

.00   CONTINUE

      ANS=W(1)
      X(K+LLTM)=X(LLTM+K)+W(1)
      RETURN
      END
```

XCD7

```
:ASE 2.0                XCD                DATE = 84355          14/52/28          PAGE 0008

:T*   NOTERM,NOID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP,NOTEST
:T*   NAME = XCD      , LINECNT =      50
SOURCE STATEMENTS =      165,PROGRAM SIZE =    16150
 DIAGNOSTICS GENERATED
```

XCD8

21

```
      SUBROUTINE OUTPUT (LLT,LL, IFIRST, DT,    TIME, ITNOW, NDT,ICOUNT)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON / PUTOUT /  YN(129), PN(129), S(129), SIGMA(129), P(129),
     1    VG(129), Z(129),    DEPTH(129),Q(129),
     2 PERME(129),VWREL(129), SIGX, BQTOT(20), TQTOT(20),    QUPTOT

      COMMON/ READIN /  CASEID(20), TEMPT, GRADT, WTPCT, AA(4,5),
     1 BB(4,4), ITYPE(20), NSUBI(20), DDT(20), DDZ(20), RHOG(20),
     2 PORS(20), IEC(20), ALPHA(20), BETA(20), ISETA(20), ISETB(20),
     3 IPAGE(20),    DP(20), DIT(20), GAMMA(20),RHOW, IZ,ITMAX

      COMMON / ABC / DU(129), DUO(129),DUOO(129), YO(129), AGE(129),
     1    Y(129), YOO(129),YMIN(129),F(129),SMX(129),
     1 LB(20), LT(20), NDTSUM(20), H(20), SSIGMA(20), SMPN(20),GDRHO,
     2 LLTM,LO,HLL,GW


      *
      F=DECOMPACTION PARAMETER
      A5=SKEMPTON'S LIQUID LIMIT
      PI(A5)=-12.39+.774*A5
      PL(A5)=A5-PI(A5)

      BSIG(F)=DEXP((3.07+(.23*A5+12.4)/(-12.4+.77*A5))/.175)+F-F

      ALPHX(F,SMX)=(.23*A5+12.4)+(-12.4+.77*A5)*(3.07-.175*DLOG(BSIG(F))
     1-.175*(1.-F)*DLOG(1.-DEXP(-(SMX+100.)/BSIG(F))))

      BETX(F)=.175*F*(-12.4+.77*A5)

      W(F,SIG,SMX)=-BETX(F)*DLOG(-DEXP((-SIG-100.)/BSIG(F))+1.)+ALPHX(F,
     1SMX)
      O(F,SIG,SMX)=.027*W(F,SIG,SMX)/(1+.027*W(F,SIG,SMX))

      QIN(PO,F,SMX)=DEXP((ALPHX(F,SMX)-PO/.027/(1.-PO))/BETX(F))

      SIG(PO,F,SMX)=-100.-BSIG(F)*DLOG(1.-QIN(PO,F,SMX))
      **
      PERM(PO)= PO**B1*DEXP(B2+B3*PO)/(1.-PO)**B4
```

```
      BQ=0.
      IF(NDT.EQ.1) CALL COMLZ( YN(1),QUPTOT,NIHIL)

      CALL COMLZ(YN(1),SIGX,NIHIL)

      ISET=0

      DO 550 L=1,LL
      CALL ABGIT(A1,A2,A3,A4,A5,B1,B2,B3,B4,GDRHO,IT,L)


      I=LB(L)
      J=LT(L)
      IP=I+1
      IM=I-1
      JP=J+1
      JM=J-1
       JX=MIN0(JM,LLTM)
      JXP=JX+1
      JXM=JX-1

      OBTAIN FLOWS AT BOTTOM AND TOP NODES OF EACH SHALE UNIT.


      ILX=MIN0(3,JX-IM)
      CALL  XCD(I,1,0,ILX,L,ANS)
      Q(I)= ANS
      IF(IBC(L).EQ.1) Q(I)=0.0
      IF(ILX*ITYPE(ITNOW).EQ.I) BQTOT(L)=0.0
      IF(ILX*ITYPE(ITNOW).EQ.1) TQTOT(L)=0.0
      BQTOT(L)=BQTOT(L)+Q(I)*DT

      MS=0
      IF(ITYPE(ITNOW).EQ.1.AND.L.EQ.LL) MS=1

           *
      Q(JXP)= Q(I)-(DU(I)-DU(JXP-MS)-DUO(I)+DUO(JXP-MS))/DT
     1 +MS*(DDZ(IT)/(1.-O(F(JXP),BQ,SMX(JXP))) -DU(JX)) /DT

      TQTOT(L)= TQTOT(L)+Q(JXP)*DT
      IF(L.EQ.LL) QUPTOT=QUPTOT+Q(JXP)*DT

      IS OUTPUT DESIRED?
```

Output 2

```
      IPG=IPAGE(ITNOW)
      IF(IPG.EQ.0)  ISET=1
      IF(IPG.EQ.1.AND.NDT.NE.NDTSUM(ITNOW)) ISET=1
      IF(IPG.GT.2.AND.MOD(NSUBI(ITNOW),IPG).NE.0.AND.NDT.
     1NE.NDTSUM(ITNOW)) ISET=1
      IF(ISET.EQ.1) GO TO 550


      SIGX=SMPN(L)+GDRHO*DDZ(IT)

      DO 563 K=I,JXP

      SIGX=SIGX-GDRHO*DDZ(IT)
           *
      YN(K)=O(F(K),BQ,SMX(K))
      IF(SIGX.GE.0.000001) YN(K)=O(F(K),SIGX,SMX(K))

      PN(K)=GW*(DU(K)+HLL)
      S(K)=SIGX+PN(K)
           *
      SIGMA(K)=0.0
      IF(Y(K).LT.(O(F(K),BQ,SMX(K))-0.000001)) SIGMA(K)=SIG(Y(K),
     1F(K),SMX(K))
      P(K)=S(K)-SIGMA(K)
      PERME(K)=PERM(Y(K))

      GRAIN VELOCITY

563   VG(K)=(DU(1)-DU(K)-DUO(1)+DUO(K))/DT



      RELATIVE WATER FLOW

      IF(JX.EQ.I) GO TO 552
      IF(JX.NE.IP) GO TO 553

      CALL XCD(IP,1,0,4,L,ANS)
      Q(IP)=ANS
      GO TO 552

553   CALL XCD(JX,1,0,8,L,ANS)
      Q(JX)=ANS

      DO 554  M=IP,JXM
      CALL XCD(M,1,0,7,L,ANS)
```

```
654   Q(M)=ANS

652   CONTINUE

650   CONTINUE

      IF(ISET.EQ.1) RETURN

      IF(ITYPE(ITNOW).EQ.1)  VG(LLT)=VG(LLTM)-Q(LLT)+Q(LLTM)

      CONVERT FROM CGS TO COMMON UNITS:
      PRESSURES IN PSI. POROSITY IN %. VELOCITIES AND FLOW RATES IN
      FEET PER THOUSAND YEARS. PERMEABILITY IN MILLIDARCIES.

      FTPCM=0.03281
      FTPTY=0.03281*3.1558*10.0**10.0
      DARCY =   0.98687 * 10.0**(-11.)
      PSI =   1.4504 * 10.0**(-5.)
      PERCT = 100.0
      SECPMY =   3.1558 * 10.0**(+13.)


      DO 300  K=1,LLT
      DEPTH(K)=(DU(K)+HLL)*FTPCM
      YN(K) = YN(K)* PERCT
      Z(K)  = Y(K) * PERCT
      PN(K) = PN(K)* PSI
      P(K)  = P(K) * PSI
      S(K)  = S(K) * PSI
      SIGMA(K) =  SIGMA(K)* PSI
      VG(K)    = VG(K)*FTPTY
      Q(K)    =  Q(K) *FTPTY
      PERME(K) =  PERME(K) / DARCY
      AGE(K)   =  AGE(K) / SECPMY
300   VWREL(K) = Q(K) / Y(K)


      QUPTOT=QUPTOT*FTPCM
      TIME = TIME / SECPMY

      DO 200 L=1,LL
      BQTOT(L)=BQTOT(L)*FTPCM
200   TQTOT(L)=TQTOT(L)*FTPCM

      WRITE OUTPUT BEGINNING AT TOP OF SEDIMENTARY SECTION
```

Output 4

23

```
      IF( IFIRST.EQ.0 )  WRITE(6,400)
      IFIRST= 2

      ITXXX=ITNOW+ICOUNT

      WRITE(6, 500)  TIME, ITXXX, NDT , QUPTOT


      DO 600 ML=1,LL
      L= LL+1-ML
      I= LB(L)
      J= LT(L)

      JX = MIN0( LLT ,J )

      WRITE(6,700)  L, L, TQTOT(L)

      IS =  JX +1 -I
      DO  800  MK= 1, IS
      K =  JX +1  -MK

800   WRITE(6, 900)  Z(K), YN(K), P(K), PN(K), S(K), SIGMA(K),
     1 PERME(K), VG(K), VWREL(K), Q(K), AGE(K), DEPTH(K)

      WRITE(6,901)L,BQTOT(L)
500   CONTINUE

      CHANGE UNITS BACK TO CGS.

      DO 910  L=1,LL
      BQTOT(L)= BQTOT(L)/ FTPCM
910   TQTOT(L)= TQTOT(L) / FTPCM

      QUPTOT = QUPTOT / FTPCM
      TIME=TIME*SECPMY

      DO 920  K=1,LLT
920   AGE(K) = AGE(K) * SECPMY


400   FORMAT ('1'  //  T54,'START OF COMPUTED OUTPUT' //
     1 T59,'SYSTEM OF UNITS' //
     2 T32,'POROSITY',     T64,'PERCENT' /
     3 T32,'PRESSURE',     T64,'PSIA'    /
     4 T32,'PERMEABILITY',   T64,'MILLIDARCYS' /
```

Outputs

```
     5 T32,'VELOCITY',     T64,'FEET / THOUSAND YEARS' /
     6 T32,'FLOW RATE',T64,'CUBIC FEET / SQUARE FOOT / THOUSAND YEARS'/
     7 T32,'CUMULATIVE FLOW',T64,'CUBIC FEET / SQUARE FOOT' /
     8 T32,'SEDIMENT AGE',     T64,'MILLIONS OF YEARS'    /
     9 T32,'DEPTH BELOW SEA FLOOR', T64,'FEET '     )

500  FORMAT ('1'  // T33,'TIME SINCE INITIATION OF SEDIMENTARY SEC',
     1    'TION = ', G10.3   ,   T91,'(MILLION YEARS)' /
     2 T50,'CURRENT TIME INCREMENT NUMBER = ' , I2  /
     3 T41,'TOTAL NUMBER OF ELAPSED TIME SUB-INCREMENTS = ', I4   /
     4 T17,'TOTAL FLOW OUT OF THE TOP OF THE SEDIMENTARY SECTION',
     5 ' AS A WHOLE = ', G12.4   ,  T97, '(CUBIC FEET / SQUARE FOOT)'    )

600  FORMAT( ' ' ///  T55,'COMPUTATIONS FOR SHALE UNIT ', I2  /
     1 T38,'CUMULATIVE FLOW OUT OF TOP OF SHALE UNIT ',    I2, ' = ',
     2  G12.4    //   T3,'SEDIMENT    NORMAL     WATER     NORMAL    ',
     3 'OVERBURDEN   MATRIX    PERMEA-      MATRIX  RELATIVE WATER    ' ,
     4 'FLOW     SEDIMENT   SUB SEAFLOOR' /
     5 T3,'POROSITY  POROSITY  PRESSURE  PRESSURE    PRESSURE   ',
     6 'PRESSURE     BILITY  VELOCITY     VELOCITY     RATE     ',
     7 'AGE     DEPTH'    )

900  FORMAT (  ' '  , T2, F7.2, T12,F7.2, T22, F8.2, T33,F8.2,
     1 T42,F8.2, T53,F8.2, T63, G10.3, T73, G10.3,  T84, G10.3,
     2 T97, G10.3,  T109, G10.3, T119, F9.2    )

901  FORMAT(' ' /  T37,' CUMULATIVE FLOW INTO BOTTOM OF SHALE UNIT ',
     1     I2, ' = ' , G 12.4    )

      RETURN
      END
```

Outputs

24

CT* NOTERM,NOID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP,NOTEST
CT* NAME = OUTPUT , LINECNT = 50
SOURCE STATEMENTS = 125,PROGRAM SIZE = 7428
DIAGNOSTICS GENERATED

Output7

```
      SUBROUTINE BC2(L,M)
C
C     GIVEN THE SHALE UNIT, L, AND THE BOTTOM NODE, M ,
C     FIND Y(M) AND DU(M) WHEN THE PRESSURE AT M IS DICTATED.
C
      IMPLICIT REAL*8 (A-H,O-Z)                                  BC2 1
C
      COMMON / READIN /  CASEID(20), TEMPT , GRADT, WTPCT, AA(4,5),
     1BB(4,4), ITYPE(20), NSUBI(20),DDT(20),DDZ(20), RHOG(20),
     2PORS(20),IBC(20), ALPHA(20), BETA(20), ISETA(20), ISETB(20),
     3IPAGE(20),     DP(20),DIT(20),GAMMA(20),RHOW, IZ,ITMAX
C
C
C
      COMMON/ABC/DU(129),DUO(129),DUOO(129), YO(129),     AGE(129),
     1    Y(129), YOO(129),YMIN(129),F(129),SMX(129),
     1     LB(20), LT(20), NDTSUM(20), H(20), SSIGMA(20),
     2SMPN(20),GDRHO,LLTM,LO, HLL, GW
C
C      *
C     F=DECOMPACTION PARAMETER
C     A5=SKEMPTON'S LIQUID LIMIT
C     PI(A5)=-12.39+.774*A5
C     PL(A5)=A5-PI(A5)
C
      BSIG(F)=DEXP((3.07+(.23*A5+12.4)/(-12.4+.77*A5))/.175)+F-F
C
      ALPHX(F,SMX)=(.23*A5+12.4)+(-12.4+.77*A5)*(3.07-.175*DLOG(BSIG(F))
     1-.175*(1.-F)*DLOG(1.-DEXP(-(SMX+100.)/BSIG(F))))
C
      BETX(F)=.175*F*(-12.4+.77*A5)
C
      W(F,SIG,SMX)=-BETX(F)*DLOG(-DEXP((-SIG-100.)/BSIG(F))+1.)+ALPHX(F,
     1SMX)
      O(F,SIG,SMX)=.027*W(F,SIG,SMX)/(1+.027*W(F,SIG,SMX))
C
      QIN(PO,F,SMX)=DEXP((ALPHX(F,SMX)-PO/.027/(1.-PO))/BETX(F))
C
      SIG(PO,F,SMX)=-100.-BSIG(F)*DLOG(1.-QIN(PO,F,SMX))
C
C      *
C
      OSIG(F,SI,SMX)=.027*BETX(F)/BSIG(F)/(1.+.027*W(F,SI,SMX))**2
     1/(1.-DEXP((SI+100.)/BSIG(F)))
      ITER= 0
      MP= M+1
```

25

```
      CALL ABGIT(A1,A2,A3,A4,A5,B1,B2,B3,B4,GDRHO, IT, L )

      S=  SIG( Y(M), F(M),SMX(M) )

      U = DDZ(IT)/ (1. -.5*( Y(MP)+ O(F(M),S,SMX(M)) ) )
      X= -S + (1.-ALPHA(IT))*SMPN( L)+(1.-BETA(IT))*GW*
     1 ( DU(MP) + HLL + U ) +GAMMA(IT)

      XD =  -1.+ GW*(1.-BETA(IT))*U*U / DDZ(IT)/ 2. *DSIG(F(M),S,SMX(M))

      DS = - X / XD
      S = S + DS

      Y(M) =  O( F(M), S,SMX(M) )
      DU(M)= DU(MP) + DDZ(IT)/ (1.-0.5*(Y(MP)+Y(M)))

      ITER = ITER +1
      IF(ITER.LT.40.AND. DABS( DS/S).GT. 0.0001) GO TO 1

      RETURN
      END
```

BC2-2

```
EASE 2.0                BC2                 DATE = 84355          14/52/28          PAGE 0003
    36
CT*  NOTERM,NOID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP,NOTEST
CT*  NAME = BC2      , LINECNT =      50
SOURCE STATEMENTS =       27,PROGRAM SIZE =     3308
 DIAGNOSTICS GENERATED
```

BC2 → 3

26

DGELB 1

```
      SUBROUTINE DGELB

      PURPOSE
         TO SOLVE A SYSTEM OF SIMULTANEOUS LINEAR EQUATIONS WITH A
         COEFFICIENT MATRIX OF BAND STRUCTURE.

      USAGE
         CALL DGELB(R,A,M,N,MUD,MLD,EPS,IER)

      DESCRIPTION OF PARAMETERS
         R    -DOUBLE PRECISION M BY N RIGHT HAND SIDE MATRIX
          (DESTROYED). ON RETURN R CONTAINS THE SOLUTION
          OF THE EQUATIONS.
         A    -DOUBLE PRECISION M BY M COEFFICIENT MATRIX WITH
          BAND STRUCTURE (DESTROYED).
         M    -THE NUMBER OF EQUATIONS IN THE SYSTEM.
         N    -THE NUMBER OF RIGHT HAND SIDE VECTORS.
         MUD    -THE NUMBER OF UPPER CODIAGONALS (THAT MEANS
          CODIAGONALS ABOVE MAIN DIAGONAL).
      MLD    -THE NUMBER OF LOWER CODIAGONALS (THAT MEANS
          CODIAGONALS BELOW MAIN DIAGONAL).
         EPS    -SINGLE PRECISION INPUT CONSTANT WHICH IS USED AS
          RELATIVE TOLERANCE FOR TEST ON LOSS OF
          SIGNIFICANCE.
         IER    -RESULTING ERROR PARAMETER CODED AS FOLLOWS
          IER=0 - NO ERROR.
          IER=-1 - NO RESULT BECAUSE OF WRONG INPUT PARAME-
          TERS M,MUD,MLD OR BECAUSE OF PIVOT ELEMENT
          AT ANY ELIMINATION STEP EQUAL TO 0.
          IER=K=WARNING DUE TO POSSIBLE LOSS OF SIGNIFI-
          CANCE INDICATED AT ELIMINATION STEP K+1.
          WHERE PIVOT ELEMENT WAS LESS THAN OR
          EQUAL TO THE INTERNAL TOLERANCE EPS TIMES
          ABSOLUTELY GREATEST ELEMENT OF MATRIX A.

      REMARKS
         BAND MATRIX A IS ASSUMED TO BE STORED ROWWISE IN THE FIRST
         THE SUCCESSIVE STORAGE LOCATIONS OF TOTALLY NEEDED MA
         STORAGE LOCATIONS, WHERE
          MA=M*MC-ML*(ML+1/2    AND    ME=MA-MU*(MU+1)/2    WITH
          MC=MIN(M,1+MUD+MLD),    ML=MC-1-MLD, MU=MC-1-MUD.
       RIGHT HAND SIDE MATRIX R IS ASSUMED TO BE STORED COLUMNWISE
         IN N*M SUCCESSIVE STORAGE LOCATIONS. ON RETURN SOLUTION
```

```
      MATRIX R IS STORED COLUMNWISE TOO.
      INPUT PARAMETERS M, MUD, MLD SHOULD SATISFY THE FOLLOWING
      RESTRICTIONS  MUD NOT LESS THAN ZERO
       MLD NOT LESS THAN ZERO
       MUD+MLD NOT GREATER THAN 2*M-2.
      NO ACTION BESIDES ERROR MESSAGE IER=-1 TAKES PLACE IF THESE
      RESTRICTIONS ARE NOT SATISFIED.
      THE PROCEDURE GIVES RESULTS IF THE RESTRICTIONS ON INPUT
      PARAMETERS ARE SATISFIED AND IF PIVOT ELEMENTS AT ALL
      ELIMINATION STEPS ARE DIFFERENT FROM C. HOWEVER WARNING
      IER=K - IF GIVEN - INDICATES POSSIBLE LOSS OF SIGNIFICANCE.
      IN CASE OF A WELL SCALED MATRIX A AND APPROPRIATE TOLERANCE
      EPS, IER=K MAY BE INTERPRETED THAT MATRIX A HAS THE RANK K.
      NO WARNING IS GIVEN IF MATRIX A HAS NO LOWER CODIAGONAL.

  SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
      NONE

   METHOD
      SOLUTION IS DONE BY MEANS OF GAUSS ELIMINATION WITH
      COLUMN PIVOTING ONLY, IN ORDER TO PRESERVE BAND STRUCTURE
      IN REMAINING COEFFICIENT MATRICES.

  ..................................................................

  SUBROUTINE DGELB(R,A,M,N,MUD,MLD,EPS,IER)


  DIMENSION R(1),A(1)
  DOUBLE PRECISION R,A,PIV,TB,TOL,EPS

  TEST ON WRONG INPUT PARAMETERS
  IF(MLD)47,1,1
1 IF(MUD)47,2,2
2 MC=1+MLD+MUD
  IF(MC+1-M-M)3,3,47

  PREPARE INTEGER PARAMETERS
      MC=NUMBER OF COLUMNS IN MATRIX A
      MU=NUMBER OF ZEROS TO BE INSERTED IN FIRST ROW OF MATRIX A
      ML=NUMBER OF MISSING ELEMENTS IN LAST ROW OF MATRIX A
      MR=INDEX OF LAST ROW IN MATRIX A WITH MC ELEMENTS
      MZ=TOTAL NUMBER OF ZEROS TO BE INSERTED IN MATRIX A
      MA=TOTAL NUMBER OF STORAGE LOCATIONS NECESSARY FOR MATRIX A
      NM=NUMBER OF ELEMENTS IN MATRIX R
3 IF(MC-M)5,5,4
4 MC=M
5 MU=MC-MUD-1
```

```
      ML=MC-MLD-1
      MR=M-ML
      MZ=(MU*(MU+1))/2
      MA=M*MC-(ML*(ML+1))/2
      NM=N*M
C
C     MOVE ELEMENTS BACKWARD AND SEARCH FOR ABSOLUTELY GREATEST ELEMENT
C     (NOT NECESSARY IN CASE OFA MATRIX WITHOUT LOWER CODIAGONALS)
      IER=0
      PIV=0.D0
      IF(MLD)14,14,6
    6 JJ=MA
      J=MA-MZ
      KST=J
      DO 9 K=1,KST
      TB=A(J)
      A(JJ)=TB
      TB=DABS(TB)
      IF(TB-PIV)8,8,7
    7  PIV=TB
    8 J=J-1
    9 JJ=JJ-1
C
C     INSERT ZEROS IN FIRST MU ROWS(NOT NECESSARY IN CASE MZ=C)
      IF(MZ)14,14,10
   10 JJ=1
      J=1+MZ
      IC=1+MUD
      DO 13 I=1,MU
      DO 12K=1,MC
      A(JJ)=0.00
      IF(K-IC)11,11,12
   11 A(JJ)=A(J)
      J=J+1
   12 JJ=JJ+1
   13 IC=IC+1
C
C     GENERATE TEST VALUE FOR SINGULARITY
   14 TOL=EPS*PIV
C
C
C     START DECOMPOSITION LOOP
      KST=1
      IDST=MC
      IC=MC-1
      DO 38 K=1,M
      IF(K-MR-1)16,16,15
   15 IDST=IDST-1
```

DGELB 3

```
16    ID=IDST
      ILR=K+MLD
      IF(ILR-M)18,18,17
17    ILR=M
18    II=KST

      PIVOT SEARCH IN FIRST COLUMN (ROW INDEXES FROM I=K UP TO I=ILR)
      PIV=0.00
      DO 22 I=K,ILR
      TB=DABS(A(II))
      IF(TB-PIV)20,20,19
19    PIV=TB
      J=I
      JJ=II
20    IF(I-MR)22,22,21
21    ID=ID-1
22    II=II+ID

      TEST ON SINGULARITY
      IF(PIV)47,47,23
23    IF(IER)26,24,26
24    IF(PIV-TOL)25,25,26
25    IER=K-1
26    PIV=1.00/A(JJ)

      PIVOT ROW REDUCTION AND ROW INTERCHANGE IN RIGHT HAND SIDE R
      ID=J-K
      DO 27 I=K,NM,M
      II=I+ID
      TB=PIV*R(II)
      R(II)=R(I)
27    R(I)=TB

      PIVOT ROW REDUCTION AND ROW INTERCHANGE IN COEFFICIENT MATRIX A
      II=KST
      J=JJ+IC
      DO 28 I=JJ,J
      TB=PIV*A(I)
      A(I)=A(II)
      A(II)=TB
28    II=II+1

      ELEMENT REDUCTION
      IF(K-ILR)29,34,34
29    ID=KST
      II=K+1
      MU=KST+1
      MZ=KST+IC
```

DGELB 4

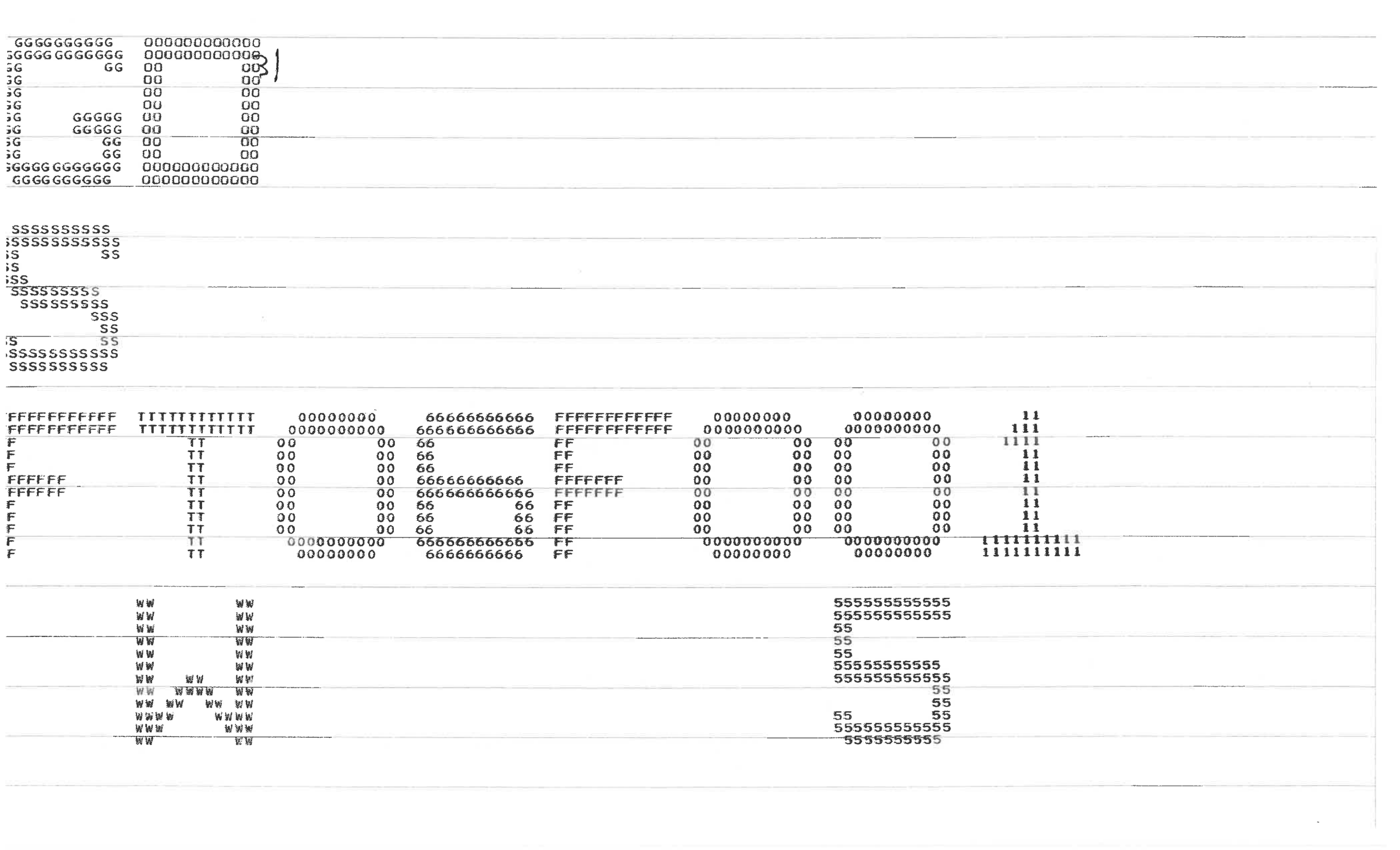
38

```
      DO 33 I=II,ILR

      IN MATRIX A
      ID=ID+MC
      JJ=I-MR-1
      IF(JJ)31,31,30
30    ID=ID-JJ
31    PIV=-A(ID)
      J=ID+1
      DO 32 JJ=MU,MZ
      A(J-1)=A(J)+PIV*A(JJ)
32    J=J+1
      A(J-1)=0.00

      IN MATRIX R
      J=K
      DO 33 JJ=I,NM,M
      R(JJ)=R(JJ)+PIV*R(J)
33    J=J+M
34    KST=KST+MC
      IF(ILR-MR)36,35,35
35    IC=IC-1
36    ID=K-MR
      IF(ID)38,38,37
37    KST=KST-ID
38    CONTINUE
      END OF DECOMPOSITION LOOP


      BACK SUBSTITUTION
      IF(MC-1)46,46,39
39    IC=2
      KST=MA+ML-MC+2
      II=M
      DO 45 I=2,M
      KST=KST-MC
      II=II-1
      J=II-MR
      IF(J)41,41,40
40    KST=KST+J
41    DO 43 J=II,NM,M
      TB=R(J)
      MZ=KST+IC-2
      ID=J
      DO 42 JJ=KST,MZ
      ID=ID+1
42    TB=TB-A(JJ)*R(ID)
43    R(J)=TB
```

DGELB 5

```
      IF(IC-MC)44,45,45
44    IC=IC+1
45    CONTINUE
46    RETURN

      ERROR RETURN
47    IER=-1
      RETURN
      END
```

DGELB 6


GGGGGGGGG 000000000000
GG GG 00 00
GG 00 00
GG GGGGG 00 00
GG GGGGG 00 00
GG GG 00 00
GGGGGGGGGGG 000000000000
GGGGGGGGG 000000000000
SSSSSSSSS
SSSSSSSSSSS
SS SS
SS
SSSSSSSSS
SSSSSSSS
SSS
SS
SS SS
SSSSSSSSSSS
SSSSSSSSS
FFFFFFFFFFF TTTTTTTTTTTT 00000000 66666666666 FFFFFFFFFFFF 00000000 00000000 11
FFFFFFFFFFF TTTTTTTTTTTT 0000000000 666666666666 FFFFFFFFFFFF 0000000000 0000000000 111
FF TT 00 00 66 FF 00 00 00 00 1111
FF TT 00 00 66 FF 00 00 00 00 11
FF TT 00 00 66 FF 00 00 00 00 11
FFFFFF TT 00 00 66666666666 FFFFFFF 00 00 00 00 11
FFFFFF TT 00 00 666666666666 FFFFFFF 00 00 00 00 11
FF TT 00 00 66 66 FF 00 00 00 00 11
FF TT 00 00 66 66 FF 00 00 00 00 11
FF TT 00 00 66 66 FF 00 00 00 00 11
FF TT 0000000000 666666666666 FF 0000000000 0000000000 1111111111
FF TT 00000000 6666666666 FF 00000000 00000000 1111111111
WW WW 555555555555
WW WW 555555555555
WW WW 55
WW WW 55
WW WW 55
WW WW 55555555555
WW WW WW 555555555555
WW WWWW WW 55
WW WW WW WW 55
WWWW WWWW 55 55
WWW WWW 555555555555
WW WW 5555555555

"Last" Page

TIME SINCE INITIATION OF SEDIMENTARY SECTION = 0.750D-01 (MILLION YEARS)
CURRENT TIME INCREMENT NUMBER = 1
TOTAL NUMBER OF ELAPSED TIME SUB-INCREMENTS = 3
TOTAL FLOW OUT OF THE TOP OF THE SEDIMENTARY SECTION AS A WHOLE = 1407. (CUBIC FEET / SQUARE FOOT)

Output=3

COMPUTATIONS FOR SHALE UNIT 1
CUMULATIVE FLOW OUT OF TOP OF SHALE UNIT 1 = 1407.

| SEDIMENT POROSITY | NORMAL POROSITY | WATER PRESSURE | NORMAL PRESSURE | OVERBURDEN PRESSURE | MATRIX PRESSURE | PERMEA-BILITY | MATRIX VELOCITY | RELATIVE WATER VELOCITY | FLOW RATE | SEDIMENT AGE | SUB SEAFLOOR DEPTH |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 79.34 | 79.34 | -0.00 | 0.0 | -0.00 | 0.0 | 0.613D-01 | -21.4 | 27.0 | 21.4 | 0.0 | 0.0 |
| 69.03 | 52.47 | 622.32 | 436.20 | 623.80 | 1.48 | 0.201D-01 | -11.8 | 17.1 | 11.8 | 0.250D-01 | 968.31 |
| 58.68 | 48.56 | 1074.95 | 747.77 | 1122.97 | 48.02 | 0.549D-02 | -2.48 | 4.23 | 2.48 | 0.500D-01 | 1659.93 |
| 52.47 | 45.96 | 1376.54 | 1001.28 | 1564.08 | 187.54 | 0.225D-02 | 0.0 | 0.0 | 0.0 | 0.750D-01 | 2222.69 |

CUMULATIVE FLOW INTO BOTTOM OF SHALE UNIT 1 = 0.0

HESE ARE THE INITIAL ESTIMATES

| | | |
|---|---|---|
| 0.3156D 13 | 0.4965 | 0.8331D 05 |
| 0.2367D 13 | 0.5247 | 0.6774D 05 |
| 0.1578D 13 | 0.5868 | 0.5059D 05 |
| 0.7889D 12 | 0.6903 | 0.2951D 05 |
| 0.0 | 0.7934 | 0.0 |

There is about another inch-thick of output pages

| NAME | ORIGIN | LENGTH |
|---|---|---|
| IHCFMAXI* | 12A88 | C9 |
| IHCSSQRT* | 12B58 | 145 |
| IHCEFNTR* | 12CA0 | 4A2 |
| IHCEFIOS* | 13148 | 13F2 |
| IHCETRCH* | 14540 | 26E |
| IHCUATBL* | 147B0 | 348 |
| IHCUOPT * | 14AF8 | 330 |
| $$EUFT * | 14E28 | D0 |
| PRTBUF * | 14EF8 | 6E |
| SYSFLAG * | 14F68 | 6 |
| UNITAB * | 14F70 | 68 |
| READIN | 14FD8 | 968 |
| ABC | 15940 | 2B40 |
| ERR | 18480 | 4 |
| ERODE | 18488 | 54 |
| XXCD | 184E0 | C |
| SPACES | 184F0 | 458B8 |
| PUTOUT | 5DDA8 | 2DA8 |

| NAME | LOCATION | NAME | LOCATION | NAME | LOCATION | NAME | LOCATION |
|---|---|---|---|---|---|---|---|
| ERRMON | 123C0 | IHCERRE | 123D8 | SSDD51 | 12A5C | | |
| MAX0 | 12A88 | MIN0 | 12A9E | AMAX0 | 12AB4 | AMIN0 | 12ACA |
| SQRT | 12B58 | | | | | | |
| ARITH# | 12CA0 | | | | | | |
| FIOCS# | 13148 | FIOCSBEP | 1314C | | | | |
| IHCTRCH | 14540 | ERRTRA | 14548 | | | | |

Module Map 2

ENTRY ADDRESS 00

TOTAL LENGTH 60B50

***X DOES NOT EXIST BUT HAS BEEN ADDED TO DATA SET

UTHORIZATION CODE IS 0.

=64-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,LET,MAP
DEFAULT OPTION(S) USED - SIZE=(196608,65536)

30

MODULE MAP

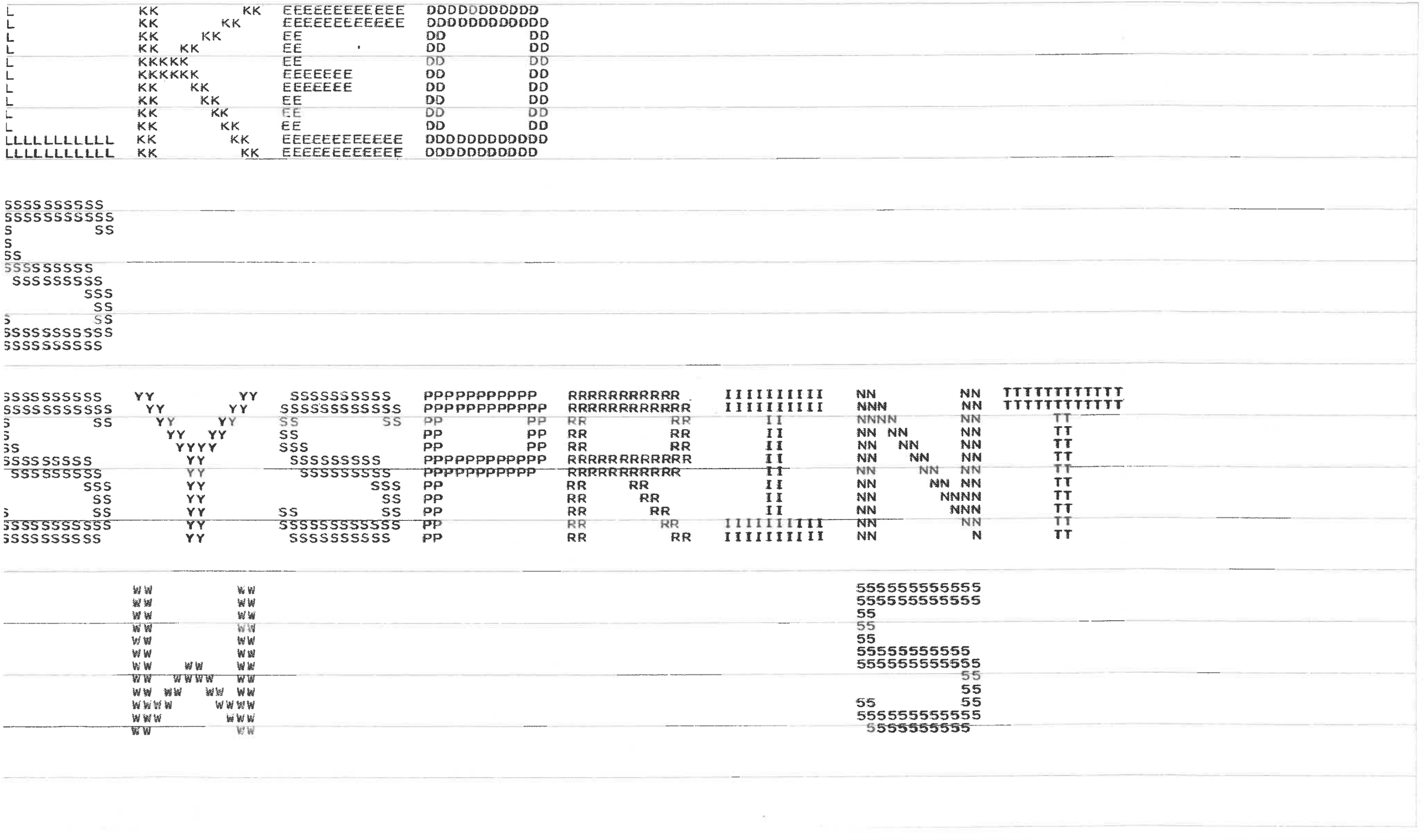
Module Map 1

CONTROL SECTION

| NAME | | ORIGIN | LENGTH |
|---|---|---|---|
| MAIN | | 00 | 4D8E |
| ABGIT | | 4D90 | 384 |
| DEPLET | | 5118 | 84E |
| ESTIM8 | | 5968 | 18B4 |
| XCD | | 7520 | 3F16 |
| OUTPUT | | 8438 | 1D04 |
| BC2 | | D140 | CEC |
| DGELB | | DE30 | 8DA |
| $COML | * | EA10 | 118 |
| IHCLEXP | * | E828 | 288 |
| IHCLLOG | * | EDB0 | 1F8 |
| IHCFEXIT | * | EFA8 | 1C |
| IHCFDXPD | * | EFC8 | 1A0 |
| IHCFRXPR | * | F168 | 183 |
| IHCECOMH | * | F2F0 | 11A2 |
| IHCCOMH2 | * | 10498 | 511 |
| $ONOFF | * | 109B0 | A0 |
| MSINEXT | * | 10A50 | 2EC |
| $TMINT | * | 10D40 | C8 |
| IHCFCVTH | * | 10E08 | 114D |
| IHCSLOG | * | 11F58 | 1AA |
| $CSER | * | 12108 | B0 |
| IHCFMAXD | * | 12188 | 60 |
| IHCSEXP | * | 12228 | 192 |
| IHCERRM | * | 123C0 | 6C4 |

ENTRY

| NAME | LOCATION | NAME | LOCATION | NAME | LOCATION | NAME | LOCATION |
|---|---|---|---|---|---|---|---|
| COMLZ | EA10 | COML | EA26 | | | | |
| DEXP | E828 | | | | | | |
| DLOG10 | EDB0 | DLOG | EDC8 | | | | |
| EXIT | EFA8 | | | | | | |
| FDXPD# | EFC8 | | | | | | |
| FRXPR# | F168 | | | | | | |
| IBCOM# | F2F0 | FDIOCS# | F3AC | IGETERR1 | FDA4 | IGETLGTH | FDB0 |
| INTSW | 10046 | | | | | | |
| SEQDASD | 10712 | | | | | | |
| INTPRT | 109B0 | | | | | | |
| NEXTV | 10A50 | INEXTV | 10A50 | NEXT | 10A84 | INEXT | 10A84 |
| XNEXTF | 10A84 | | | | | | |
| TIMINT | 10D40 | | | | | | |
| ADCON# | 10E08 | FCVAOUTP | 10E82 | FCVLOUTP | 10F3A | FCVZOUTP | 1107A |
| FCVIOUTP | 11408 | FCVEOUTP | 118FA | FCVCOUTP | 11B0C | | |
| ALOG10 | 11F58 | ALOG | 11F70 | | | | |
| CSERR | 12108 | $CSER$ | 1215A | | | | |
| DMAXI | 12188 | DMIN1 | 121CE | | | | |
| EXP | 12228 | | | | | | |


KK KK EEEEEEEEEEEE DDDDDDDDDD
KK KK EEEEEEEEEEEE DDDDDDDDDDD
KK KK EE DD DD
KK KK EE DD DD
KKKKK EE DD DD
KKKKKK EEEEEEE DD DD
KK KK EEEEEEE DD DD
KK KK EE DD DD
KK KK EE DD DD
KK KK EE DD DD
KK KK EEEEEEEEEEE DDDDDDDDDDD
KK KK EEEEEEEEEEE DDDDDDDDDD
SSSSSSSSS
SSSSSSSSSS
S SS
S
SS
SSSSSSSSS
SSSSSSSS
SSS
SS
S SS
SSSSSSSSSS
SSSSSSSSS
YY YY SSSSSSSSS PPPPPPPPPP RRRRRRRRR IIIIIIIII NN NN TTTTTTTTTT
YY YY SSSSSSSSSS PPPPPPPPPP RRRRRRRRRR IIIIIIIII NNN NN TTTTTTTTTT
SS YY YY SS SS PP PP RR RR II NNNN NN TT
YY YY SS PP PP RR RR II NN NN NN TT
YYYY SSS PP PP RR RR II NN NN NN TT
YY SSSSSSSS PPPPPPPPPP RRRRRRRRR II NN NN NN TT
YY SSSSSSSS PPPPPPPPPP RRRRRRRRR II NN NN NN TT
YY SSS PP RR RR II NN NN NN TT
YY SS PP RR RR II NN NNNN TT
YY SS SS PP RR RR II NN NNN TT
YY SSSSSSSSSS PP RR RR IIIIIIIII NN NN TT
YY SSSSSSSSS PP RR RR IIIIIIIII NN N TT
WW WW
WW WW
WW WW
WW WW
WW WW
WW WW
WW WWWW WW
WW WW WW WW
WWWW WWWW
WWW WWW
WW WW
55555555555
555555555555
55
55
55
55555555555
555555555555
55
55
55 55
555555555555
5555555555

39

```
*OPTIONS IN EFFECT*   NOTERM,NOID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP,NOTEST
*OPTIONS IN EFFECT*   NAME = DGELB    , LINECNT =      50
*STATISTICS*     SOURCE STATEMENTS =      133,PROGRAM SIZE =     3034
*STATISTICS*   NO DIAGNOSTICS GENERATED

*STATISTICS*   NO DIAGNOSTICS THIS STEP
```

DGELB 7