89-06

ANALYTICAL PARAMETRIC STUDY OF A TIED-BACK SYSTEM FOR REINFORCING DISTRESSED BRIDGE ABUTMENTS

FINAL REPORT

Volume III

Source Listing of Programs BART, PLOT, and PLOT2

Submitted to
South Dakota Department of Transportation

by

Sangchul Bang, Ph.D., P.E.
and
Hyeong I. Yeon

Department of Civil Engineering
South Dakota School of Mines and Technology
Rapid City, SD 57701

December, 1990

ANALYTICAL PARAMETRIC STUDY OF A TIED-BACK SYSTEM FOR REINFORCING DISTRESSED BRIDGE ABUTMENTS

FINAL REPORT

VOLUME III

Source Listing of Programs BART, PLOT, and PLOT2

Submitted to

South Dakota Department of Transportation

by

Sangchul Bang, Ph.D., P. E.

and

Hyeong I. Yeon

Department of Civil Engineering
South Dakota School of Mines and Technology
Rapid City, SD 57701

December, 1990

The contents of this report reflect the view of the authors who are responsible for the facts and the accuracy of the data presented herein. The contents do not ncessarily reflect the official views or policies of the South Dakota Department of Transportation, the State Transportation Commission, or the U. S. Federal Highway Administration. This report does not constitute a standard, specification, or regulation.

TABLE OF CONTENTS

LISTING OF PROGRAM BART

A complete listing of the program is given in this section. The program is written in FORTRAN 5 for IBM compatible personal computers. Compilation of the program was done with the Microsoft Fortran Compiler 5.0. It is recommended to store and run the program from a hard disk. The size of the source program, executable program, input, and output files are approximately:

Input file :	770 bytes
Outpuf file :	146,000 bytes
.FOR program :	106,000 bytes
.EXE program :	288,000 bytes

The command necessary to compile the .FOR file using the Microsoft Fortran Compiler is

```
FL /FPi  BART.FOR
```

This command compiles the .FOR file with or without math coprocessor.

```
C
C     THIS PROGRAM DESIGNS THE TIE-ROD ANCHOR SYSTEM WITH DRILLED-IN
C     CONCRETE ANCHOR FOR REINFORCING DISTRESSED BRIDGE ABUTMENT BASED
C     ON LIMITING EQUILIBRIUM METHOD AND ANALYZES THE SYSTEM BY A
C     GENERALIZED 3-DIMENSIONAL PLANE STRAIN INCREMENTAL AND ITERATIVE
C     FINITE ELEMENT METHOD DEVELOPED BY S. BANG, SDSM&T, OCT. 1990
C
C
C     INPUT: IF DATA FILE EXISTS, PROGRAM WILL READ DATA FROM EXISTING FI
LE
C            AND ANY CHANGE CAN BE MADE. A NEW DATA FILE IS THEN CREATED
C            IF DATA FILE DOES NOT EXIST, PROGRAM WILL CREATE ONE
C     OUTPUT:IF OUTPUT FILE EXISTS, NEW RESULTS WILL OVERWRITE IT
C            IF OUTPUT FILE DOES NOT EXIST, PROGRAM WILL CREATE ONE
C
C
C     FILE  4: FILE FOR FINITE ELEMENT GRID PLOT
C     FILE  5: KEYBOARD INPUT
C     FILE  6: MONITOR OUTPUT
C     FILE  7: OUTPUT FILE
C     FILE  8: NEW INPUT DATA FILE
C     FILE  9: EXISTING INPUT DATA FILE
C     FILE 10: EDITTED INPUT DATA FILE
C
C
      COMMON /BLK1/AE(10),EIY(10),E(10),POIS(10),EX(102),GX(102),INR
      COMMON /BLK17/EIZ(10),GJX(10),IPILE,XPILE,INCR,XXS(102,5),
     +              YYS(102,5),IFILE
      COMMON /BLK12/ZZS(102,5),XYT(102,5),XZT(102,5),YZT(102,5)
      COMMON /BLK7/DUN(10,9),PA
      COMMON /BLK10/ID(100,6),PN(6),FFX(10),FFY(10),TTX(102),TTY(102)
      COMMON/BLK11/AX,GAM,T1,T2,T3
      COMMON/BLK13/CD
      CHARACTER TITLE*80
      CHARACTER*12 FILEOUT,FILEIN,FNEW
      CHARACTER   CD*1
      CHARACTER*1 YES/'Y'/
C
C
C     FORMAT STATEMENTS
C
    6 FORMAT(A1)
   10 FORMAT(A80)
   11 FORMAT(A12)
   18 FORMAT(F20.9)
   19 FORMAT(I2)
   20 FORMAT(//1X,A80)
   30 FORMAT(//1X,'**INPUT DATA AS FOLLOWS**',//,
     +         3X,'HEIGHT OF ABUTMENT WALL          =',F10.3,' FT',/,
     +         3X,'DEPTH TO TIE ROD                 =',F10.3,' FT',/,
     +         3X,'BACKFILL SLOPE (V/H)             =',F10.3,/,
     +         3X,'SURCHARGE LOADING                =',F10.3,' PSF',/,
     +         3X,'WALL FRICTION ANGLE              =',F10.3,' DEG',/,
```

```
     +         3X,'BRIDGE SKEW ANGLE                    =',F10.3,' DEG',/,
     +         3X,'TIE ROD SPACING                      =',F10.3,' FT',/,
     +         3X,'ALLOWABLE STRESS OF TIE ROD          =',E10.3,' PSI',//,
     +         1X,'BACKFILL SOIL PROPERTIES')
  31 FORMAT(/ 3X,'UNIT WEIGHT                          =',F10.3,' PCF',/,
     +         3X,'FRICTION ANGLE                       =',F10.3,' DEG',//,
     +         1X,'FOUNDATION SOIL PROPERTIES',/,
     +         3X,'UNIT WEIGHT                          =',F10.3,' PCF',/,
     +         3X,'COHESION                             =',F10.3,' PSF',/,
     +         3X,'FRICTION ANGLE                       =',F10.3,' DEG',/,
     +         3X,'UNDRAINED SHEAR STRENGTH             =',F10.3,' PSF',
     +         1X,'AT',F5.1,' FT',//,
     +         1X,'CONCRETE ANCHOR DIAMETER               =',F10.3,' FT',/,
     +         1X,'FACTOR OF SAFETY                       =',F10.3)
  40 FORMAT(//1X,'**OUTPUT DATA AS FOLLOWS**',//,
     +         3X,'MINIMUM DIAMETER OF TIE ROD    =',F10.3,' IN')
  50 FORMAT(/1X,'**CALCULATED MINIMUM LENGTH OF DRILLED IN',1X,
     +        'CONCRETE ANCHOR =',F10.3,' FT')
  80 FORMAT(//3X,'**FRICTION ANGLE OF FOUNDATION SOIL CAN NOT BE',
     +           ' ESTIMATED FROM SU',/,3X,'REENTER FOUNDATION SOIL',
     +           ' PROPERTIES')
  90 FORMAT(//3X,'**ESTIMATED FRICTION ANGLE OF FOUNDATION SOIL IS',
     +           F10.3,/3X,'PROBABLY TOO HIGH -- USE CAUTION')
 100 FORMAT(/1X,'HEIGHT OF ABUTMENT WALL IN FT?',\)
 101 FORMAT(/1X,'DEPTH TO TIE ROD FROM GROUND SURFACE IN FT?',\)
 102 FORMAT(/1X,'BACKFILL SLOPE V/H?',\)
 103 FORMAT(/1X,'SURCHARGE LOADING IN PSF?',\)
 110 FORMAT(/1X,'WALL FRICTION ANGLE IN DEG?',\)
 111 FORMAT(/1X,'BRIDGE SKEW ANGLE IN DEG?',\)
 112 FORMAT(/1X,'TIE ROD SPACING IN FT?',\)
 113 FORMAT(/1X,'ALLOWABLE STRESS OF TIE ROD IN PSI?',\)
 114 FORMAT(/1X,'FACTOR OF SAFETY?',\)
 120 FORMAT(/1X,'BACKFILL SOIL PROPERTIES')
 121 FORMAT(/1X,'UNIT WEIGHT IN PCF?',\)
 122 FORMAT(/1X,'SOIL DESIGNATION NUMBER IN INTEGER TO USE REFERENCE',
    +/1X,'OR HIT RETURN TO INPUT PARAMETERS DIRECTLY',/1X,'NUMBER=?',\)
 130 FORMAT(/1X,'FOUNDATION SOIL PROPERTIES')
 131 FORMAT(/1X, 'UNIT WEIGHT IN PCF?',\)
 230 FORMAT(/1X,'NEW VALUE OF DRILLED IN CONC ANCHOR LENGTH',/1X,
    +          'OR HIT RETURN TO KEEP SAME VALUE',/1X,
    +          'MUST BE GREATER THAN THE WALL HEIGHT',/1X,
    +          'LENGTH=?',\)
 330 FORMAT(//1X,'MINIMUM REQUIRED DIMENSIONS ARE',//,
    +            3X,'LENGTH OF TIE ROD          =',F10.3,' FT',/,
    +            3X,'LENGTH OF CONCRETE ANCHOR =',F10.3,' FT')
 340 FORMAT(  /1X,'ENTER TITLE OF PROBLEM?',\)
 350 FORMAT(/1X,'**LENGTH OF CONCRETE ANCHOR IS LESS THAN THE HEIGHT',
    +  1X,'OF ABUTMENT',/1X,'INCREASE THE LENGTH BIGGER THAN',F10.3)
 360 FORMAT(/1X,'**THE FOLLOWING WILL BE USED IN FINITE ELEMENT ',
    +              'ANALYSIS',//,
    +            4X,'DIAMETER OF TIE ROD    =',F10.3,' IN',/,
    +            4X,'LENGTH OF TIE ROD      =',F10.3,' FT',/,
    +            4X,'SURCHARGE LOADING      =',F10.3,' PSF',/)
```

```
  370 FORMAT(/1X,      'DIAMETER OF CONCRETE ANCHOR IN FT',/,
     +        1X,'HIT RETURN IF 2.5 FT IS USED',/1X,'DIAMETER=?',\)
  380 FORMAT(/1X,'NAME OF OUTPUT FILE',/1X,'IF THIS NAME ALREADY ',
     + 'EXISTS RESULTS WILL OVERWRITE ON THE EXISTING FILE',/1X,
     + 'OTHERWISE A NEW OUTPUT FILE WILL BE CREATED',/1X,'NAME=?',\)
  381 FORMAT(/1X,'NEW VALUE OF TIE ROD DIAMETER',
     +      /1X,'OR HIT RETURN TO KEEP SAME VALUE',/1X,'DIAMETER=?',\)
  382 FORMAT(/1X,'NEW VALUE OF TIE ROD LENGTH',
     +      /1X,'OR HIT RETURN TO KEEP SAME VALUE',/1X,'LENGTH=?',\)
  383 FORMAT(/1X,'NEW VALUE OF SURCHARGE LOADING',
     +      /1X,'OR HIT RETURN TO KEEP SAME VALUE',/1X,'SURCHARGE=?',\)
  384 FORMAT(/1X,'NAME OF INPUT FILE',/1X,'IF THIS NAME ALREADY EXISTS',
     + ' PROGRAM WILL READ DATA DIRECTLY FROM THIS FILE',/1X,
     + 'IF NOT THE PROGRAM WILL CREATE THE DATA FILE AS YOU ENTER',
     + /1X,'NAME=?',\)
  385 FORMAT(/1X,'DATA WILL BE READ FROM EXISTING FILE',
     +      /1X,'IF NO CHANGE IN DATA NO FURTHER INPUT IS NEEDED',
     + ' AFTER THIS',/1X,'ANY CHANGE IN DATA(Y/N)?',\)
  386 FORMAT(/1X,'NEW DATA FILE IS CREATED')
  387 FORMAT(/1X,'NAME OF NEW DATA FILE AFTER CHANGES?',\)
  388 FORMAT( /1X,'FRICTION ANGLE IN DEGREES?',\)
  389 FORMAT(/1X,'COHESION IN PSF?',\)
  390 FORMAT(/1X,'FRICTION ANGLE IN DEGREES?',\)
  391 FORMAT(/1X,'UNDRAINED STRENGTH IN PSF?',\)
  392 FORMAT(/1X,'UNDRAINED STRENGTH MEASURED AT WHAT DEPTH IN FT?',\)
  393 FORMAT(/1X,'**INPUT COHESION AND FRICTION AS ZERO IF UNKNOWN '
     +         ,'AND TO BE CALCULATED FROM UNDRAINED STRENGTH',/,
     +        1X,'**INPUT UNDRAINED STRENGTH AS ZERO IF UNKNOWN',
     +         ' AND TO BE CALCULATED FROM COHESION AND FRICTION')
  394 FORMAT(//1X,'******* FINAL DESIGN SELECTION *******',
     +  /1X,'TIE ROD TENSION              =',F10.3,1X,'LBS PER TIE',
     +  /1X,'TIE ROD DIAMETER             =',F10.3,1X,'IN',
     +  /1X,'TIE ROD LENGTH               =',F10.3,1X,'FT',
     +  /1X,'CONCRETE ANCHOR LENGTH       =',F10.3,1X,'FT',
     +  /1X,'CONCRETE ANCHOR DIAMETER     =',F10.3,1X,'FT',//)
  395 FORMAT(/1X,'**ERROR**REENTER IN INTEGER(S)',\)
  396 FORMAT(/1X,'EDITTED DATA FILE NAME=',A12)
  397 FORMAT(//1X,'INPUT DATA FILE NAME  =',A12,
     +        /1X,'OUTPUT DATA FILE NAME =',A12)
  398 FORMAT(/1X,'**TYPE ANY ALPHABET LETTER IN CAPITAL**',
     +/1X,'**ANY NUMERIC DATA MUST HAVE A DECIMAL UNLESS SPECIFIED**')
C
C     READ INPUT AND OUTPUT FILE NAMES
C
      WRITE(6,398)
      WRITE(6,384)
      READ(5,11)FILEIN
      WRITE(6,380)
      READ(5,11)FILEOUT
      OPEN(UNIT=7,FILE=FILEOUT,STATUS='UNKNOWN')
      WRITE(6,340)
C
C     READ TITLE OF THE PROBLEM
```

```
C
      READ(5,10)TITLE
      WRITE(7,20)TITLE
      WRITE(7,397)FILEIN,FILEOUT
C
C     OPEN EXISTING DATA FILE
C
      NAMEI=9
      OPEN(UNIT=NAMEI,FILE=FILEIN,STATUS='OLD',ERR=2)
C
C     READ DATA DIRECTLY FROM EXISTING FILE
C
      IFILE=0
      WRITE(6,385)
      READ(5,6)CD
      IF(CD .NE. YES)GO TO 4
      WRITE(6,387)
      READ(5,11)FNEW
      WRITE(7,396)FNEW
      OPEN(UNIT=10,FILE=FNEW,STATUS='NEW')
      GO TO 4
C
C     NEW DATA FILE IS CREATED
C
    2 IFILE=1
      WRITE(6,386)
      CLOSE(UNIT=NAMEI,STATUS='KEEP')
      OPEN(UNIT=8,FILE=FILEIN,STATUS='NEW')
    4 CONTINUE
C
C     READ DATA
C
      WRITE(6,100)
      IF(IFILE .EQ. 1)GO TO 500
      CALL EDIT1(H)
      GO TO 501
  500 READ(5,*)H
      IF(IFILE .EQ. 1)WRITE(8,*)H
  501 WRITE(6,101)
      IF(IFILE .EQ. 1)GO TO 502
      CALL EDIT1(HPP)
      GO TO 503
  502 READ(5,*)HPP
      IF(IFILE .EQ. 1)WRITE(8,*)HPP
  503 WRITE(6,102)
      IF(IFILE .EQ. 1)GO TO 504
      CALL EDIT1(SLOPE)
      GO TO 505
  504 READ(5,*)SLOPE
      IF(IFILE .EQ. 1)WRITE(8,*)SLOPE
  505 WRITE(6,103)
      IF(IFILE .EQ. 1)GO TO 506
      CALL EDIT1(Q)
```

```
      GO TO 507
  506 READ(5,*)Q
      IF(IFILE .EQ. 1)WRITE(8,*)Q
  507 WRITE(6,110)
      IF(IFILE .EQ. 1)GO TO 508
      CALL EDIT1(DEL)
      GO TO 509
  508 READ(5,*)DEL
      IF(IFILE .EQ. 1)WRITE(8,*)DEL
  509 WRITE(6,111)
      IF(IFILE .EQ. 1)GO TO 510
      CALL EDIT1(TH)
      GO TO 511
  510 READ(5,*)TH
      IF(IFILE .EQ. 1)WRITE(8,*)TH
  511 WRITE(6,112)
      IF(IFILE .EQ. 1)GO TO 512
      CALL EDIT1(SH)
      GO TO 513
  512 READ(5,*)SH
      IF(IFILE .EQ. 1)WRITE(8,*)SH
  513 WRITE(6,113)
      IF(IFILE .EQ. 1)GO TO 514
      CALL EDIT1(FA)
      GO TO 515
  514 READ(5,*)FA
      IF(IFILE .EQ. 1)WRITE(8,*)FA
  515 WRITE(6,114)
      IF(IFILE .EQ. 1)GO TO 516
      CALL EDIT1(FS)
      GO TO 517
  516 READ(5,*)FS
      IF(IFILE .EQ. 1)WRITE(8,*)FS
  517 CONTINUE
      TTH=TH
C
C     CONCRETE ANCHOR DIMENSION - 2.5 FT CYLINDER UNLESS SPECIFIED
C
      WRITE(6,370)
      IF(IFILE .EQ. 1)GO TO 518
      CALL EDIT2(D)
      GO TO 519
  518 READ(5,18)D
      IF(D .EQ. 0.)D=2.5
      IF(IFILE .EQ. 1)WRITE(8,18)D
C
C     READ DATA
C
  519 WRITE(6,120)
      WRITE(6,122)
      K=1
      IF(IFILE .EQ. 1)GO TO 520
      CALL EDIT3(KCODE1)
```

```
      GO TO 521
  520 READ(5,19,ERR=551)KCODE1
      GO TO 552
  551 WRITE(6,395)
      GO TO 520
  552 CONTINUE
      IF(IFILE .EQ. 1)WRITE(8,19)KCODE1
  521 IF(KCODE1 .EQ. 0)GO TO 14
      CALL DUNPRT(KCODE1,K)
      CALL SOIL(KCODE1,K)
      GAM1=-FFY(K)
      PH1=DUN(K,6)
      GO TO 15
   14 WRITE(6,121)
      IF(IFILE .EQ. 1)GO TO 522
      CALL EDIT1(GAM1)
      GO TO 533
  522 READ(5,*)GAM1
      IF(IFILE .EQ. 1)WRITE(8,*)GAM1
  533 WRITE(6,390)
      IF(IFILE .EQ. 1)GO TO 534
      CALL EDIT1(PH1)
      GO TO 15
  534 READ(5,*)PH1
      IF(IFILE .EQ. 1)WRITE(8,*)PH1
   15 CONTINUE
    7 WRITE(6,130)
      WRITE(6,122)
      K=1
      IF(IFILE .EQ. 1)GO TO 523
      CALL EDIT3(KCODE2)
      GO TO 524
  523 READ(5,19,ERR=553)KCODE2
      GO TO 554
  553 WRITE(6,395)
      GO TO 523
  554 CONTINUE
      IF(IFILE .EQ. 1)WRITE(8,19)KCODE2
  524 IF(KCODE2 .EQ. 0)GO TO 16
      CALL DUNPRT(KCODE2,K)
      CALL SOIL(KCODE2,K)
      GAM2=-FFY(K)
      PH2=DUN(K,6)
      C2=DUN(K,5)
      SU=0.
      GO TO 17
   16 WRITE(6,393)
      WRITE(6,131)
      IF(IFILE .EQ. 1)GO TO 525
      CALL EDIT1(GAM2)
      GO TO 535
  525 READ(5,*)GAM2
      IF(IFILE .EQ. 1)WRITE(8,*)GAM2
```

```
  535 WRITE(6,389)
      IF(IFILE .EQ. 1)GO TO 536
      CALL EDIT1(C2)
      GO TO 537
  536 READ(5,*)C2
      IF(IFILE .EQ. 1)WRITE(8,*)C2
  537 WRITE(6,390)
      IF(IFILE .EQ. 1)GO TO 538
      CALL EDIT1(PH2)
      GO TO 539
  538 READ(5,*)PH2
      IF(IFILE .EQ. 1)WRITE(8,*)PH2
  539 WRITE(6,391)
      IF(IFILE .EQ. 1)GO TO 540
      CALL EDIT1(SU)
      GO TO 541
  540 READ(5,*)SU
      IF(IFILE .EQ. 1)WRITE(8,*)SU
  541 WRITE(6,392)
      IF(IFILE .EQ. 1)GO TO 542
      CALL EDIT1(ZT)
      GO TO 543
  542 READ(5,*)ZT
      IF(IFILE .EQ. 1)WRITE(8,*)ZT
  543 CONTINUE
   17 CONTINUE
      PH2=PH2*0.0174533
      PI=4.*ATAN(1.0)
      ITEST=0
C
C     IF SU IS NOT KNOWN, ESTIMATE SU AT THE ABUT BASE FROM C2 AND PH2
C
      IF(SU .EQ. 0. .AND. C2 .NE. 0.) SU=GAM2*H/2.
     +          *(1.-TAN(PI/4.-PH2/2.)**2)+C2*TAN(PI/4.-PH2/2.)
C
C     IF C2 AND PH2 ARE NOT KNOWN, ESTIMATE FROM SU ASSUMING NORMALLY
C     CONSOLIDATED SOIL (C2=0.)
C
      PH2=PH2/0.0174533
      IF(C2 .NE. 0. .OR. PH2 .NE. 0.)GO TO 60
      TEM=1.-2.*SU/GAM2/ZT
      IF(TEM .GT. 0.)GO TO 70
      WRITE(6,80)
      GO TO 7
   70 PH2=PI/2.-2.*ATAN(SQRT(TEM))
      PH2=PH2/0.0174533
      IF(PH2 .GT. 50.)WRITE(6,90)PH2
   60 WRITE(7,30)H,HPP,SLOPE,Q,DEL,TH,SH,FA
      WRITE(7,31)GAM1,PH1,GAM2,C2,PH2,SU,ZT,D,FS
C
C     CONVERT ANGLES INTO RADIANS
C
      TH=TH*0.0174533
```

```
      DEL=DEL*0.0174533
      PH1=PH1*0.0174533
      PH2=PH2*0.0174533
C
C     CALCULATE AT-REST LATERAL EARTH PRESSURE
C
      XK=1.-SIN(PH1)
      PR=(GAM1*H**2/2.+Q*H)*XK*COS(DEL)
C
C     TIE ROD TENSION - FACTOR OF SAFETY IS CONSIDERED
C
      T=FS*PR/COS(TH)
      TMAX=T*SH
C
C     TIE ROD DIAMETER
C
      DA=SQRT(4.*TMAX/FA/PI)
      WRITE(7,40)DA
C
C     DRILLED-IN CONCRETE DEADMAN AND TIE-ROD LENGTH
C
      F1=TAN(PI/4.-PH1/2.)
      F2=TAN(PI/4.+PH1/2.)
      F3=TAN(PI/2.-PH1)
      F4=TAN(PI/4.+PH2/2.)
      F5=TAN(PI/4.-PH2/2.)
C
C     MINIMUM TIE-ROD LENGTH REQUIRED
C
      XLMIN=H*F3
      HT=H*(1.-SLOPE*F1)/(1.+SLOPE*F2)
      YT=H*(F2+F1)/(1.+SLOPE*F2)
      XB=H*(F1+F2-F4*(1.-SLOPE*F1))/(1.+SLOPE*F2)
C
C     INCREASE THE TIE-ROD LENGTH UNTIL ENOUGH RESISTANCE IS MET
C
      XL=XLMIN-1.
  140 XL=XL+1.
      Z=0.
  142 Z=Z+1.
      IF(PH1 .GT. ATAN(SLOPE))GO TO 144
      HPMIN=(XL-XB)*F5
      GO TO 148
  144 IF(XL .GT. YT)GO TO 146
      HPMIN=(XL-H*F1)*F1
      GO TO 148
  146 HPMIN=(XL-XB)*F5
  148 IF(Z .GT. HPMIN)GO TO 140
C
C     CALCULATE ULTIMATE PILE RESISTANCE
C
      PU1=0.
      PU2=0.
```

```
      ZZ=0.
  149 ZZ=ZZ+1.
      ZD=(ZZ-0.5)/D
      AX=COEFA(ZD)
      IF(ZZ .GT. Z)GO TO 160
      IF(PH1 .GT. ATAN(SLOPE))GO TO 150
      PH=PH2
      GAM=GAM2
      GO TO 154
  150 IF(XL .GT. YT)GO TO 152
      PH=PH1
      GAM=GAM1
      GO TO 154
  152 TEM=H-XL*SLOPE
      IF(ZZ .LT. TEM)GO TO 156
      PH=PH2
      GAM=GAM2
      GO TO 154
  156 PH=PH1
      GAM=GAM1
  154 CONTINUE
      BE=PI/4.+PH/2.
      AL=PH/2.
      YK=1.-SIN(PH)
      E1=TAN(BE)
      E2=TAN(PH)
      E3=TAN(BE-2.*AL)
      E4=TAN(BE-PH)
      E5=TAN(AL)
      E6=SIN(BE)
      T1=E3**2*(E1**8-1.)+YK*(E2*E1**4)
      T2=YK*E2*E6/E4/COS(AL)+YK*E1*(E2*E6-E5)+E1**2*E5/E4
      T3=E1/E4-E3**2
C
C     NUMERICAL INTEGRATION
C
      CALL GAUSS(1,ZZ,A1,D)
      CALL GAUSS(2,ZZ,A2,D)
      PU1=PU1+A1*D
      PU2=PU2+A2*D
      GO TO 149
C
C     ULTIMATE RESISTANCE IS SMALLER OF LONG AND SHORT PILE BEHAVIORS
C     PU1 - SHORT PILE BEHAVIOR
C     PU2 - LONG PILE BEHAVIOR
C
  160 PU=MIN(PU1,PU2)
C
C     CHECK IF THE CONCRETE ANCHOR IS DEEP ENOUGH
C
      IF(PU .GT. TMAX)GO TO 200
      GO TO 142
  200 HP=Z
```

```
      WRITE(6,50)HP
      IF(HP .GT. H)GO TO 285
C
C     INCREASE THE LENGTH OF THE CONCRETE ANCHOR IF DESIRED
C
      WRITE(6,350)H
  285 CONTINUE
      WRITE(6,230)
      IF(IFILE .EQ. 1)GO TO 162
      CALL EDIT2(XHP)
      HP=XHP
      GO TO 164
  162 READ(5,18)XHP
      IF(XHP .EQ. 0.)XHP=HP
      IF(XHP .NE. 0.)HP=XHP
      IF(IFILE .EQ. 1)WRITE(8,18)XHP
  164 CONTINUE
C
C     INCREASE THE TIE ROD LENGTH IF NECESSARY
C
      IF(HP .GT. HT)GO TO 166
C
C     CONCRETE ANCHOR LIES ENTIRELY WITHIN THE BACKFILL
C
      XL=H*F1+HP*F2
      IF(XL .LT. XLMIN)XL=XLMIN
      GO TO 168
C
C     CONCRETE ANCHOR LIES WITHIN THE BACKFILL AND FOUNDATION SOIL
C
  166 XL=XB+HP*F4
      IF(XL .LT. XLMIN)XL=XLMIN
  168 CONTINUE
C
C     ACCOUNT FOR SKEW ANGLE
C
      XL=XL/COS(TH)
      WRITE(7,330)XL,HP
      HPH=HP
C
C
C     GENERALIZED 3-D PLANE STRAIN FINITE ELEMENT ANALYSIS
C
C
  400 CONTINUE
      WRITE(6,360)DA,XL,Q
      WRITE(6,381)
      IF(IFILE .EQ. 1)GO TO 528
      CALL EDIT2(XDA)
      DA=XDA
      GO TO 529
  528 READ(5,18)XDA
      IF(XDA .EQ. 0.)XDA=DA
```

```
      IF(XDA .NE. 0.)DA=XDA
      IF(IFILE .EQ. 1)WRITE(8,18)XDA
  529 WRITE(6,382)
      IF(IFILE .EQ. 1)GO TO 530
      CALL EDIT2(XXL)
      XL=XXL
      GO TO 531
  530 READ(5,18)XXL
      IF(XXL .EQ. 0.)XXL=XL
      IF(XXL .NE. 0.)XL=XXL
      IF(IFILE .EQ. 1)WRITE(8,18)XXL
  531 WRITE(6,383)
      IF(IFILE .EQ. 1)GO TO 532
      CALL EDIT2(XQ)
      Q=XQ
      GO TO 410
  532 READ(5,18)XQ
      IF(XQ .EQ. 0.)XQ=Q
      IF(XQ .NE. 0.)Q=XQ
      IF(IFILE .EQ. 1)WRITE(8,18)XQ
  410 CONTINUE
      WRITE(7,394)TMAX,DA,XL,HPH,D
      XL=XL*COS(TH)
      CALL GPS(H,HPP,Q,TTH,XL,SH,DA,D,HPH,KCODE1,KCODE2)
 1000 STOP
      END
C
C
      SUBROUTINE GAUSS(IC,Z,AREA,D)
C
C     THIS SUBROUTINE PERFORMS TWO POINT GAUSSIAN NUMERICAL INTEGRATION
C     STARTING WITH 2 SUBINTERVALS.  CONTINUE TO DOUBLE-INCREASE THE
C     SUBINTERVALS UNTIL ERROR IS LESS THAN 1 % OR 50 ITERATIONS
C
      COMMON/BLK11/AX,GAM,T1,T2,T3
      DIMENSION C(2),U(2)
C
C     WEIGHTING FUNCTIONS
C
      C(1)=1.
      C(2)=1.
C
C     GAUSSIAN POINTS
C
      U(1)=0.57735
      U(2)=-0.57735
      B=Z
      A=Z-1.
      IG=0
      M=2
    5 AM=M
      DELX=(B-A)/AM
      X2=A
```

```
      AREA=0.
      DO 10 K=1,M
      X1=X2
      X2=X1+DELX
      DO 10 J=1,2
      X=(DELX*U(J)+X1+X2)/2.
      IF(IC .EQ. 2)F=AX*GAM*X*T1
      IF(IC .EQ. 1)F=AX*GAM*X**2/D*(T2+T3*D/X)
   10 AREA=AREA+C(J)*F
      AREA=AREA*DELX/2.
      IF(IG)12,12,13
   12 AREA1=AREA
      M=2*M
      IG=1
      GO TO 5
   13 IF(ABS((AREA-AREA1)/AREA1)-0.01)15,15,16
   16 IF(M-50)12,12,15
   15 RETURN
      END
C
C
      FUNCTION COEFA(ZD)
C
C     THIS SUBROUTINE CALCULATES THE COEFFICIENT A IN ULTIMATE SOIL
C     RESISTANCE EXPRESSION
C
      IF(ZD .GE. 5.)GO TO 10
      COEFA=0.035777*(5.-ZD)**2.5+0.88
      GO TO 20
   10 COEFA=0.88
   20 RETURN
      END
C
C
      SUBROUTINE GPS(HH,HPP,QQ,ANGLE,XXL,Z10,Z11,Z12,ZDM,KCODE1,KCODE2)
C
C
C      GENERALIZED PLANE STRAIN THREE DIMENSIONAL ANALYSIS
C      BY FINITE ELEMENT METHOD.  INCLUDED ARE CONTINUUM
C      ELEMENT, REINFORCED CONTINUUM AND 3-D BEAM-COLUMN ELEMENT
C
C      THIS PROGRAM HAS FOLLOWING LIMITATIONS
C      MAXIMUM NO. OF NODES      =100
C      MAXIMUN NO. OF ELEMENTS   =102
C      MAXIMUM NO. OF MATERIALS = 10
C      MAXIMUM NO. OF UNKNOWNS   =323
C      MAXIMUM BANDWIDTH         =100
C
C
C     COMMON AND DIMENSION STATEMENTS
C
      COMMON /BLK1/AE(10),EIY(10),E(10),POIS(10),EX(102),GX(102),INR
      COMMON /BLK17/EIZ(10),GJX(10),IPILE,XPILE,INCR,XXS(102,5),
```

```
     +              YYS(102,5),IFILE
      COMMON /BLK12/ZZS(102,5),XYT(102,5),XZT(102,5),YZT(102,5)
      COMMON /BLK2/XQ(4),YQ(4),B(6,12),XN(4),ALP,EST,PHP,PHR,BA(6,6),
     +            BAP (6,6)
      COMMON /BLK3/LM(102,6),X(100),Y(100),ARE(102),EF(4,12)
      COMMON /BLK4/FX,FY,IOPT,EP(12),EK(12,12),TX,TY,ICONT,IOPTE,ICONTE
      COMMON /BLK5/DISN(100,6),SIGMA(6)
      COMMON /BLK6/SMX(5),SMY(5),SMZ(5),TXY(5),TXZ(5),TYZ(5),STR(5,6)
      COMMON /BLK7/DUN(10,9),PA
      COMMON /BLK8/A1,A2,A3,A4,A5,A6,B1,B2,C1,C2,C3,C4,C5,C6,C7,C8,C9,
     +            C10,C11,C12,C13,D1,D2,E1,E2,E3,E4,E5,E6,E7,F1,F2,F3,
     +            F4,F5,G1,SSH,DDA,DDM,D3,D4
      COMMON /BLK9/GK(323,100),GP(323)
      COMMON /BLK10/ID(100,6),PN(6),FFX(10),FFY(10),TTX(102),TTY(102)
      COMMON/BLK13/CD
      DIMENSION IEQ(100,6),DGP(323),MNO(10),GGP(323),SG(6),
     +          IIQ(100,6),IID(100,6)
      CHARACTER*1 YES/'Y'/
      CHARACTER   CD*1,ICONTD*1,JPL*1
      CHARACTER ICONT*1,  ICNEQ*1 ,IOPT*1,ICONTE*1,IOPTE*1
C
C     FORMAT STATEMENTS
C
    3 FORMAT(//,36X,'NODAL POINT DATA'//1X,'NODAL POINT COORDINATES',
     +   10X,'BOUNDARY CONDITION CODES',18X,'GLOBAL EQUATION NUMBER',
     +   //35X,'**IN INCREMENT',I4,'**',//1X,
     +   'NODE',5X,'X(N)',5X,'Y(N)',4X,'D-X',3X,'D-Y',3X,'D-Z',2X,
     +   'TH-X',2X,'TH-Y',2X,'TH-Z',9X,'D-X',3X,'D-Y',3X,'D-Z',3X,
     +   'TH-X',2X,'TH-Y',2X,'TH-Z',/)
    4 FORMAT(//,3X,'**TOTAL NUMBER OF EQUATIONS =',I4,' IN INCR=',I3)
    5 FORMAT(//,3X,'**BANDWIDTH OF MATRIX =',I4,' IN INCR=',I3)
    6 FORMAT(A1)
    7 FORMAT(F20.9)
    8 FORMAT(I1)
   26 FORMAT(I4,2F10.3,6I6,6X,6I6)
   31 FORMAT(//,31X,'NODAL POINT LOAD TO BE INCLUDED IN INCREMENT 2',
     +          //,5X,'NODE',15X,'PN(X)',15X,
     +   'PN(Y)',15X,'PN(Z)',15X,'M(X)',16X,'M(Y)',16X,'M(Z)',/)
   38 FORMAT(I9,6E20.5)
   53 FORMAT(                                /1X,'MAT.',4X,'AE',5X,'GJ1',
     +   5X,'EI2',5X,'EI3',7X,'E',5X,'POIS',4X,'FX',9X,'FY',/)
   57 FORMAT(/,10X,'MATERIAL PROPERTIES',//,1X,'MAT.',
     +   3X,'K',7X,'KUR',8X,'N',6X,'RF',
     +   6X,'C',6X,'PHI',5X,'DPHI',5X,'KB',7X,'M',5X,'PA',6X,'FX',8X,
     +   'FY',/)
   62 FORMAT(I4,1X,5E8.2,F5.2,2X,F7.4,2X,F9.4)
  108 FORMAT(I4,46X,9F6.1,1X,F6.1,F7.4,2X,F9.4)
  109 FORMAT(I4,1X,10F8.2,F9.4,1X,F9.4)
  206 FORMAT(//,11X,'SOLUTION DID NOT CONVERGE AFTER',I4,
     +   ' ITERATIONS'/11X,'RESULTING ERROR IS',E13.5)
  208 FORMAT(//,11X,'SOLUTION CONVERGED AFTER',I4,' ITERATIONS',
     +   /11X,'RESULTING ERROR IS',E13.5)
  210 FORMAT(I5,2F10.3,I5,4F10.3)
```

```
  216 FORMAT(I5,13F5.1)
  222 FORMAT(//3X,'**BANDWIDTH IS TOO BIG --- CHANGE THE SECOND INTEGER'
     +    ,' OF GK ARRAY IN BLK9 AND NBAND ACCORDINGLY',//)
  223 FORMAT(//3X,'**TOO MANY UNKNOWNS --- CHANGE THE FIRST INTEGER',
     +    ' OF GK AND GP ARRAY IN BLK9 DGP AND GGP ARRAY AND NUNKN ',
     +    'ACCORDINGLY',//)
  350 FORMAT(//,50X,'ELEMENT STRAINS AND STRESSES',//,2X,'EL.NO.',
     +    2X,'LOC',6X,'DEXX',6X,'DEYY',
     +    6X,'DGXY',6X,'DGXZ',6X,'DGYZ',7X,'SXX',7X,'SYY',7X,'SZZ',7X,
     +    'TXY',7X,'TXZ',7X,'TYZ')
  573 FORMAT(2X,'***MAT',I3,1X,'HAS CONSTANT POISSON RATIO OF',F7.3)
  575 FORMAT(//55X,'DISPLACEMENTS AT NODES',//,5X,'NODE',12X,
     +    'X-TRAN',14X,'Y-TRAN',14X,'Z-TRAN',14X,'X-ROT',14X,'Y-ROT',14X,
     +    'Z-ROT',/)
  580 FORMAT(I9,6E20.5)
  642 FORMAT(10X,'REINFORCEMENT STRESS =',E12.4)
  653 FORMAT(//,3X,'**DEVELOPED FORCES AND MOMENTS PER',
     +     ' FOOT OF ABUTMENT**',/,
     + 1X,'FOR AXIAL THRUST TENSION IS POSITIVE',/1X,
     + 'FOR BENDING MOMENT TENSION ON THE BACKFILL SIDE IS POSITIVE',//)
  654 FORMAT(1X,'TIE-ROD LEVEL AXIAL THRUST OF ABUT STEM',
     +       4X,E10.3,1X,'LB/FT')
  656 FORMAT(15X,'SHEAR FORCE',18X,E10.3,1X,'LB/FT')
  657 FORMAT(15X,'BENDING MOMENT',15X,E10.3,1X,'FT-LB/FT',/1X,//)
  659 FORMAT(1X,'AT THE BOTTOM OF ABUT STEM AXIAL THRUST',
     +       4X,E10.3,1X,'LB/FT')
  660 FORMAT(28X,'SHEAR FORCE',5X,E10.3,1X,'LB/FT')
  662 FORMAT(28X,'BENDING MOMENT',2X,E10.3,1X,'FT-LB/FT',/1X,//)
  700 FORMAT(//3X, 'RELAXATION                    =',F5.2,/,3X,
     +                 'MAXIMUM ERROR FOR ITERATION  =',F5.2,/,3X,
     +                 'MAXIMUM NO. OF ITERATION     =',I3,//)
  901 FORMAT(//,25X,'ELEMENT DATA', //,
     +    10X,' IN INCR=1 ONLY ELEMENT 1 THRU 55 ARE CONSIDERED',//,
     +    1X,'ELEM',2X,'NODE1',1X,'NODE2',1X,'NODE3',1X,'NODE4',4X,
     +    'REINF',3X,'MAT',7X,'ANGLE',7X,'TX',7X,'TY',/)
  903 FORMAT(5I6,I9,I6,F12.2,2F9.2)
  906 FORMAT(2F10.3)
  907 FORMAT(////,1X,'**GENERALIZED PLANE STRAIN FINITE ELEMENT '
     +    ,'ANALYSIS**')
  908 FORMAT(/1X,'DO YOU NEED STRAINS AND STRESSES ALONG THE SIDES ',
     +    'OF ELEMENT(Y/N)?',\)
  910 FORMAT(/1X,'TOTAL WIDTH OF ABUTMENT BASE IN FT?',\)
  911 FORMAT(/1X,'FRONT ROW PILE BATTER TOWARD BERM IN V/H?',\)
  912 FORMAT(/1X,'DIST FROM FRONT OF BASE TO FRONT PILES IN FT?',\)
  914 FORMAT(/1X,      'GIRDER SUPPORT LOAD IN LBS PER FT?',\)
  915 FORMAT(/1X,'SOIL PROPERTIES BEHIND ABUTMENT',/,
     +            1X,'ENTER KEL=1 FOR ELASTIC SOIL',/,
     +           11X,'ZERO FOR NONLINEAR SOIL')
  916 FORMAT(/1X,'ENTER DUNCAN SOIL PARAMETERS IN THE ORDER OF',/,
     +            7X,'K KUR N RF C PHI DPHI KB M POIS GX GY',/,
     +           10X,'**NOTE POIS OF NONZERO IS FOR CONSTANT POISSON',
     +        1X,'RATIO',/)
  917 FORMAT(/1X,'SOIL PROPERTIES BELOW ABUTMENT'
```

```
     +            ,/,1X,'ENTER KEL=1 FOR ELASTIC SOIL',/,
     +           11X,'ZERO FOR NONLINEAR SOIL')
  919 FORMAT(/1X,'FRONT ROW PILE PROPERTIES',/,
     +            1X,'ENTER 1 FOR CIRCULAR PILE ZERO FOR OTHER PILE',
     +           /1X,'NUMBER=?',\)
  920 FORMAT(/1X,'REAR PILE PROPERTIES'/,
     +            1X,'ENTER 1 FOR CIRCULAR PILE ZERO FOR OTHER PILE',
     +           /1X,'NUMBER=?',\)
  921 FORMAT(/1X,'ABUTMENT PROPERTIES',/,
     +            1X,      'THICKNESS IN FT?',\)
  922 FORMAT(/1X,'TIE ROD PROPERTIES',/,
     +            1X,      'MATERIAL MODULUS IN PSI?',\)
  924 FORMAT(/1X,'ABUTMENT BASE PROPERTIES',/,
     +            1X,      'THICKNESS IN FT?',\)
  926 FORMAT(/1X,'DO YOU NEED OUTPUT OF NODE CONDITION(Y/N)?',\)
  927 FORMAT(/1X,'DO YOU NEED OUTPUT OF MATERIAL PROPERTIES(Y/N)?',\)
  928 FORMAT(/1X,'DO YOU NEED OUTPUT OF ELEMENT INFORMATION(Y/N)?',\)
  929 FORMAT(/1X,'DO YOU NEED OUTPUT OF NODAL DISPLACEMENTS(Y/N)?',\)
  930 FORMAT(/1X,'DO YOU NEED OUTPUT OF ELEMENT STRESS AND STRAIN',
     +            '(Y/N)?',\)
  931 FORMAT(/1X,'DO YOU NEED OUTPUT OF NODAL LOAD(Y/N)?',\)
  932 FORMAT(//3X,'**HORIZONTAL WALL MOVEMENT**',/,
     +            5X,'AT THE TOP         =',F10.5,' FT.',/,
     +            5X,'AT THE TIE-ROD     =',F10.5,' FT.',/,
     +            5X,'AT THE BASE        =',F10.5,' FT.')
  933 FORMAT(  3X,'**NOTE THAT THE CALCULATED DISPLACEMENTS',
     +               ' WILL NOT BE SUMMED IN INCR=2',/)
  934 FORMAT(/1X,'SOIL YOUNGS MODULUS?',\)
  935 FORMAT(/,10X,'MATERIAL PROPERTIES',//,1X,'MAT.',10X,'E',10X,'POIS'
     +            ,10X,'FY')
  936 FORMAT(5X,'**SOLUTION HAS TWO INCREMENTS',/,
     +           7X,'IN INCR 1 ONLY FOUNDATION SOIL STRESSES ARE ',
     +            'CALCULATED')
  937 FORMAT(5X,'**SOLUTION HAS ONE INCREMENT')
  938 FORMAT(/,I4,5X,E10.3,5X,F7.3,5X,F8.2)
  941 FORMAT(/1X,'RELAXATION FACTOR',
     +      /1X,'HIT RETURN IF DEFAULT IS USED',/1X,'RF=?',\)
  943 FORMAT(/1X,'ERROR ALLOWED DURING ITERATION',
     +      /1X,'HIT RETURN IF DEFAULT IS USED',/1X,'ERROR=?',\)
  945 FORMAT(/1X,'ENTER 1 FOR INCREMENTAL ANALYSIS',
     +      /1X,'HIT RETURN FOR NONINCREMENTAL ANALYSIS',
     +      /1X,'NUMBER=?',\)
  947 FORMAT(/1X,'DIST FROM FRONT OF ABUTMENT TO GIRDER SUPPORT IN FT',
     +       /1X,'**GIRDER SUPPORT MUST BE BETWEEN PILES**',
     +       /1X,'DISTANCE=?',\)
  948 FORMAT(/1X,'HORIZ EXTENSION OF BERM IN FRONT OF ABUTMENT IN FT',
     +       /1X,'**MUST BE GREATER THAN ZERO**',/1X,'EXTENSION=?',\)
  949 FORMAT(/1X,'SLOPE OF SOIL BEYOND BERM IN H/V?',\)
  950 FORMAT(/1X,'REAR ROW PILE BATTER TOWARD BACKFILL IN V/H',/,
     +       1X,'HIT RETURN IF SAME AS FRONT ROW PILE',/1X,
     +          'BATTER=?',\)
  951 FORMAT(/1X,     'LENGTH OF FRONT ROW PILES IN FT?',\)
  952 FORMAT(/1X,'LENGTH OF REAR ROW PILES IN FT',/,
```

```
     +       1X,'HIT RETURN IF SAME AS FRONT ROW PILE',/1X,
     +          'LENGTH=?',\)
  953 FORMAT(/1X,       'ENTER ZERO FOR POINT BEARING PILES',/,
     +  1X,  '      1 FOR FRICTION PILES',/1X,'NUMBER=?',\)
  960 FORMAT(/1X,'MAXIMUM NUMBER OF ITERATIONS (BETWEEN 1 AND 9)',
     +      /1X,'HIT RETURN IF DEFAULT IS USED',/1X,'NUMBER=?',\)
  961 FORMAT(/1X,'DIST FROM REAR EDGE OF BASE TO REAR PILES IN FT?',\)
  962 FORMAT(/1X,'SOIL POISSON RATIO?',\)
  963 FORMAT(/1X,'SOIL UNIT WEIGHT?',\)
  964 FORMAT(/1X,'**ERROR**REENTER IN INTEGER(S)',\)
  965 FORMAT(///1X,'GEOMETRIC PARAMETERS',
     + /3X,'WIDTH OF ABUTMENT BASE                    =',F10.3,' FT',
     + /3X,'DISTANCE FROM ABUTMENT TO GIRDER SUPPORT=',F10.3,' FT',
     + /3X,'HORIZ EXTENSION OF BERM IN FRONT OF ABUT=',F10.3,' FT',
     + /3X,'SLOPE OF SOIL BEYOND BERM (H/V)           =',F10.3,
     + /3X,'HORIZ EXTENSION OF SLOPE BEYOND BERM      =',F10.3,' FT',
     + /3X,'DEPTH TO TOP OF DRILLED-IN ANCHOR         =',F10.3,' FT')
  966 FORMAT(/3X,'FRONT ROW PILE BATTER (V/H)                =',F10.3,
     + /3X,'REAR ROW PILE BATTER (V/H)                =',F10.3,
     + /3X,'LENGTH OF FRONT ROW PILES                 =',F10.3,' FT',
     + /3X,'LENGTH OF REAR ROW PILES                  =',F10.3,' FT')
  967 FORMAT(/5X,'**POINT BEARING PILES**')
  968 FORMAT(/5X,'**FRICTION PILES**')
  969 FORMAT(/3X,'DIST FROM FRONT OF BASE TO FRONT PILES  =',F10.3,' FT',
     +    /3X,'DIST FROM REAR  OF BASE TO REAR  PILES  =',F10.3,' FT')
  970 FORMAT(///1X,'MATERIAL NUMBERS AS FOLLOWS',
     + /3X,'1  = BACKFILL SOIL',
     + /3X,'2,3= FOUNDATION SOIL',
     + /3X,'4  = FRONT ROW PILES',
     + /3X,'5  = REAR ROW PILES',
     + /3X,'6  = ABUTMENT STEM',
     + /3X,'7  = TIE ROD',
     + /3X,'8  = DRILLED IN CONCRETE ANCHOR',
     + /3X,'9  = ABUTMENT BASE')
  971 FORMAT(/1X,'HORIZ EXTENSION OF SLOPE IN FT?',\)
  972 FORMAT(/1X,'DEPTH TO TOP OF DRILLED-IN CONCRETE ANCHOR IN FT',
     +       /1X,'HIT RETURN IF ZERO',/1X,'DEPTH=?',\)
  973 FORMAT(I2)
  974 FORMAT(/1X,'REAR PILE PROPERTIES',/1X,
     + 'ENTER Y IF SAME AS FRONT PILE OTHERWISE ENTER N',/1X,
     + '**NO OF PILES CAN BE DIFFERENT**',\)
  975 FORMAT(/1X,'ENTER RATIO OF NO OF PILES (REAR/FRONT)?',\)
  976 FORMAT(/1X,'**IF FINITE ELEMENT GRID IS NEEDED TYPE  PLOT')
C
C     INITIALIZATION
C
      DO 2 I=1,6
    2 PN(I)=0.
      WRITE(7,907)
      A1=HH
      E1=HPP
      E2=XXL
      IOPT=0
```

```
      SSH=Z10
      DDA=Z11
      DDM=Z12
      NUNKN=323
      NBAND=100
      MAXNP=100
C
C     ATMOSPHERIC PRESSURE IN PSF
C
      PA=2116.8
      DO 530 II = 1,NUNKN
      DO 520 JJ=1,NBAND
  520 GK(II,JJ)=0.
      GGP(II)=0.
  530 GP(II)=0.
C
C     DEFAULT BOUNDARY CONDITIONS
C
      DO 527 II=1,MAXNP
      DO 525 KK=1,3
      ID(II,KK)=0
  525 ID(II,KK+3)=1
  527 CONTINUE
C
C     READ DATA FOR NONLINEAR INCREMENTAL ANALYSIS
C
      WRITE(6,960)
      IF(IFILE .EQ. 1)GO TO 111
      CALL EDIT6(ITMAX)
      GO TO 112
  111 READ(5,8,ERR=551)ITMAX
      GO TO 554
  551 WRITE(6,964)
      GO TO 111
  554 CONTINUE
      IF(ITMAX .EQ. 0)ITMAX=9
      IF(IFILE .EQ. 1)WRITE(8,8)ITMAX
  112 WRITE(6,941)
      IF(IFILE .EQ. 1)GO TO 113
      CALL EDIT2(RELAX)
      GO TO 114
  113 READ(5,7)RELAX
      IF(RELAX .EQ. 0.)RELAX=0.8
      IF(IFILE .EQ. 1)WRITE(8,7)RELAX
  114 WRITE(6,943)
      IF(IFILE .EQ. 1)GO TO 115
      CALL EDIT2(ERMAX)
      GO TO 116
  115 READ(5,7)ERMAX
      IF(ERMAX .EQ. 0.)ERMAX=0.05
      IF(IFILE .EQ. 1)WRITE(8,7)ERMAX
  116 WRITE(6,945)
      IF(IFILE .EQ. 1)GO TO 117
```

```
      CALL EDIT6(INR)
      GO TO 118
  117 READ(5,8,ERR=552)INR
      GO TO 555
  552 WRITE(6,964)
      GO TO 117
  555 CONTINUE
      IF(INR .EQ. 0)INR=0
      IF(IFILE .EQ. 1)WRITE(8,8)INR
  118 ALP=0.
      EST=0.
      PHP=0.
      PHR=0.
      WRITE(7,700)RELAX,ERMAX,ITMAX
      IF(INR .EQ. 1)WRITE(7,936)
      IF(INR .NE. 1)WRITE(7,937)
C
C     CONVERT ANGLES INTO RADIANS
C
      PHP=PHP*0.0174533
      PHR=PHR*0.0174533
      IF(ALP .NE. 0.)CALL ROTATE
      NUMNP=0
C
C     INPUT NODAL POINT DATA AND GENERATE NODAL POINTS ON STRAIGHT LINE
C
      WRITE(6,910)
      IF(IFILE .EQ. 1)GO TO 119
      CALL EDIT1(A3)
      GO TO 120
  119 READ(5,*)A3
      IF(IFILE .EQ. 1)WRITE(8,*)A3
  120 WRITE(6,947)
      IF(IFILE .EQ. 1)GO TO 121
      CALL EDIT1(B1)
      GO TO 122
  121 READ(5,*)B1
      IF(IFILE .EQ. 1)WRITE(8,*)B1
  122 WRITE(6,948)
      IF(IFILE .EQ. 1)GO TO 123
      CALL EDIT1(D1)
      GO TO 124
  123 READ(5,*)D1
      IF(IFILE .EQ. 1)WRITE(8,*)D1
  124 WRITE(6,949)
      IF(IFILE .EQ. 1)GO TO 125
      CALL EDIT1(D2)
      GO TO 126
  125 READ(5,*)D2
      IF(IFILE .EQ. 1)WRITE(8,*)D2
  126 IF(D1 .LT. 0.01)D1=0.01
      WRITE(6,971)
      IF(IFILE .EQ. 1)GO TO 165
```

```
      CALL EDIT1(D3)
      GO TO 166
  165 READ(5,*)D3
      IF(IFILE .EQ. 1)WRITE(8,*)D3
  166 WRITE(6,972)
      IF(IFILE .EQ. 1)GO TO 167
      CALL EDIT2(D4)
      GO TO 168
  167 READ(5,7)D4
      IF(IFILE .EQ. 1)WRITE(8,7)D4
  168 WRITE(6,911)
      IF(IFILE .EQ. 1)GO TO 127
      CALL EDIT2(C3)
      GO TO 128
  127 READ(5,7)C3
      IF(IFILE .EQ. 1)WRITE(8,7)C3
  128  WRITE(6,950)
      IF(IFILE .EQ. 1)GO TO 129
      CALL EDIT2(C7)
      GO TO 130
  129 READ(5,7)C7
      IF(C7 .EQ. 0.)C7=C3
      IF(IFILE .EQ. 1)WRITE(8,7)C7
  130  WRITE(6,951)
      IF(IFILE .EQ. 1)GO TO 131
      CALL EDIT2(C9)
      GO TO 132
  131 READ(5,7)C9
      IF(IFILE .EQ. 1)WRITE(8,7)C9
  132 WRITE(6,952)
      IF(IFILE .EQ. 1)GO TO 133
      CALL EDIT2(C10)
      GO TO 134
  133  READ(5,7)C10
      IF(C10 .EQ. 0.)C10=C9
      IF(IFILE .EQ. 1)WRITE(8,7)C10
  134 WRITE(6,953)
      IF(IFILE .EQ. 1)GO TO 135
      CALL EDIT6(IPILE)
      GO TO 136
  135 READ(5,8,ERR=553)IPILE
      GO TO 558
  553 WRITE(6,964)
      GO TO 135
  558 CONTINUE
      IF(IFILE .EQ. 1)WRITE(8,8)IPILE
  136 CONTINUE
      XPILE=1.5
      WRITE(6,912)
      IF(IFILE .EQ. 1)GO TO 137
      CALL EDIT1(C4)
      GO TO 138
  137 READ(5,*)C4
```

```
      IF(IFILE .EQ. 1)WRITE(8,*)C4
  138 WRITE(6,961)
      IF(IFILE .EQ. 1)GO TO 150
      CALL EDIT1(C8)
      GO TO 151
  150 READ(5,*)C8
      IF(IFILE .EQ. 1)WRITE(8,*)C8
  151 CONTINUE
      WRITE(7,965)A3,B1,D1,D2,D3,D4
      WRITE(7,966)C3,C7,C9,C10
      IF(IPILE .EQ. 0)WRITE(7,967)
      IF(IPILE .NE. 0)WRITE(7,968)
      WRITE(7,969)C4,C8
      C3=1./C3
      C7=1./C7
      F2=ZDM-HPP
      CALL AUTO(1,NUMNP,K,NUMEL,ANGLE)
C
C     WRITE XY COORDS ONTO FILE XYDATA FOR FINITE ELEMENT GRID PLOT
C
      OPEN(UNIT=4,FILE='XYDATA',STATUS='UNKNOWN')
      WRITE(4,*)NUMNP
      DO 17 I=1,NUMNP
   17 WRITE(4,*)X(I),Y(I)
C
C     BOUNDARY CONDITIONS
C
      CALL AUTO(2,NUMNP,K,NUMEL,ANGLE)
C
C     CHANGE ID ARRAY TO GLOBAL EQN NUMBERS IN INCR=2 OR LAST INCREMENT
C
      WRITE(6,926)
      IF(IFILE .EQ. 1)GO TO 139
      CALL EDIT7(ICNEQ)
      GO TO 19
  139 READ(5,6)ICNEQ
      IF(IFILE .EQ. 1)WRITE(8,6)ICNEQ
   19 NEQ=0
      INCR=2
      IF(INR .NE. 1)INCR=1
      IF(ICNEQ .EQ. YES)WRITE(7,3)INCR
      DO 30 N=1,NUMNP
      DO 25 I=1,6
C
C     SKIP TRIVIAL EQUATIONS
C
      IF(ID(N,I).GT.0) GO TO 23
      NEQ=NEQ+1
      IEQ(N,I)=NEQ
      GO TO 25
   23 IEQ(N,I)=0
   25 CONTINUE
      IF(ICNEQ .NE. YES)GO TO 30
```

```
      WRITE(7,26) N,X(N),Y(N),(ID(N,I),I=1,6),(IEQ(N,I),I=1,6)
   30 CONTINUE
      WRITE(7,4) NEQ,INCR
      IF(NEQ .GT. NUNKN)WRITE(6,223)
      NNEQ=NEQ
      DO 33 I=1,100
      DO 33 J=1,6
      IIQ(I,J)=IEQ(I,J)
   33 IID(I,J)=ID(I,J)
C
C
C     READ NODAL POINT LOAD AND PUT IN GLOBAL LOAD VECTOR
C
C
      WRITE(6,914)
      IF(IFILE .EQ. 1)GO TO 140
      CALL EDIT1(PN(2))
      GO TO 141
  140 READ(5,*)PN(2)
      IF(IFILE .EQ. 1)WRITE(8,*)PN(2)
  141 NODE=29
      PN(2)=-PN(2)
      WRITE(6,931)
      IF(IFILE .EQ. 1)GO TO 142
      CALL EDIT7(ICONT)
      GO TO 143
  142 READ(5,6)ICONT
      IF(IFILE .EQ. 1)WRITE(8,6)ICONT
  143 IF(ICONT .NE. YES)GO TO 49
      WRITE(7,31)
      WRITE(7,38)NODE,(PN(J),J=1,6)
   49 CONTINUE
      DO 50 I=1,6
      IG=IEQ(NODE,I)
      IF(IG .EQ. 0)GO TO 50
      GGP(IG)=0.
      GGP(IG)=PN(I)
   50 CONTINUE
C
C     READ TABLE OF MATERIAL PROPERTIES
C     K=1-3 SOILS
C       4-9 STRUCTURAL ELEMENTS
C
   51 CONTINUE
      WRITE(6,927)
      IF(IFILE .EQ. 1)GO TO 144
      CALL EDIT7(ICONT)
      GO TO 145
  144 READ(5,6)ICONT
      IF(IFILE .EQ. 1)WRITE(8,6)ICONT
  145 K=0
      IF(ICONT .EQ. YES)WRITE(7,970)
   68 K=K+1
```

```
      IF(K .EQ. 1)KCODE=KCODE1
      IF(K .EQ. 2 .OR. K .EQ. 3)KCODE=KCODE2
      IF(K .EQ. 1)WRITE(6,915)
      IF(K .EQ. 2)WRITE(6,917)
      IF(K .EQ. 4)WRITE(6,919)
      IF(K .EQ. 5)GO TO 581
      GO TO 582
  581 WRITE(6,974)
      IF(IFILE .EQ. 1)GO TO 651
      CALL EDIT7(JPL)
      GO TO 583
  651 READ(5,6)JPL
      IF(IFILE .EQ. 1)WRITE(8,6)JPL
  583 IF(JPL .NE. YES)GO TO 584
      WRITE(6,975)
      IF(IFILE .EQ. 1)GO TO 652
      CALL EDIT1(RP)
      GO TO 586
  652 READ(5,*)RP
      IF(IFILE .EQ. 1)WRITE(8,*)RP
  586 AE(K)=AE(K-1)*RP
      GJX(K)=GJX(K-1)*RP
      EIY(K)=EIY(K-1)*RP
      EIZ(K)=EIZ(K-1)*RP
      E(K)=E(K-1)
      POIS(K)=POIS(K-1)
      FFX(K)=FFX(K-1)
      FFY(K)=FFY(K-1)
      GO TO 585
  584 WRITE(6,920)
  582 IF(K .EQ. 6)WRITE(6,921)
      IF(K .EQ. 7)WRITE(6,922)
      IF(K .EQ. 9)WRITE(6,924)
      IF(K .LE. 3)GO TO 110
C
C     BEAM ELEMENTS
C
      CALL AUTO(3,NUMNP,K,NUMEL,ANGLE)
  585 IF(ICONT .NE. YES)GO TO 55
      IF(K .EQ. 4)WRITE(7,53)
      WRITE(7,62)K,AE(K),GJX(K),EIY(K),EIZ(K),E(K),POIS(K),FFX(K),FFY(K)
   55 CONTINUE
      MNO(K)=ITYP
      IF(K .EQ. 9)GO TO 69
      GO TO 68
  110 IF(K .EQ. 3)GO TO 561
      GO TO 562
  561 KEL=KELL
      KCODE=-1
      GO TO 147
  562 IF(IFILE .EQ. 1)GO TO 146
      CALL EDIT3(KEL)
      GO TO 147
```

```
  146 READ(5,973,ERR=557)KEL
      GO TO 556
  557 WRITE(6,964)
      GO TO 146
  556 CONTINUE
      IF(IFILE .EQ. 1)WRITE(8,973)KEL
  147 CONTINUE
      KELL=KEL
      IF(KCODE .EQ. 0 .OR. KCODE .EQ. -1)GO TO 302
      CALL DUNPRT(KCODE,K)
      CALL SOIL(KCODE,K)
  302 CONTINUE
      IF(KCODE .NE. 0)GO TO 56
      IF(KEL .EQ. 1)GO TO 59
      WRITE(6,916)
C
C     CONTINUUM ELEMENTS - NONLINEAR
C
      IF(IFILE .EQ. 1)GO TO 148
      CALL EDIT9(K)
      GO TO 149
  148 READ(5,*)(DUN(K,I),I=1,9),POIS(K),FFX(K),FFY(K)
      IF(IFILE .EQ. 1)WRITE(8,*)(DUN(K,I),I=1,9),POIS(K),FFX(K),FFY(K)
  149 GO TO 58
   59 WRITE(6,934)
C
C     CONTINUUM ELEMENTS - LINEAR
C
      IF(IFILE .EQ. 1)GO TO 152
      CALL EDIT1(E(K))
      GO TO 153
  152 READ(5,*)E(K)
      IF(IFILE .EQ. 1)WRITE(8,*)E(K)
  153 WRITE(6,962)
      IF(IFILE .EQ. 1)GO TO 154
      CALL EDIT1(POIS(K))
      GO TO 155
  154 READ(5,*)POIS(K)
      IF(IFILE .EQ. 1)WRITE(8,*)POIS(K)
  155 WRITE(6,963)
      IF(IFILE .EQ. 1)GO TO 156
      CALL EDIT1(FFY(K))
      GO TO 157
  156 READ(5,*)FFY(K)
      IF(IFILE .EQ. 1)WRITE(8,*)FFY(K)
  157 CONTINUE
      DUN(K,1)=E(K)/2116.8
      DUN(K,2)=DUN(K,1)*2
      DO 61 I=3,9
   61 DUN(K,I)=0.
      FFX(K)=0.
   58 CONTINUE
   56 CONTINUE
```

```
      IF(KCODE .NE. -1)GO TO 54
      DO 52 I=1,9
   52 DUN(K,I)=DUN(K-1,I)
      E(K)=E(K-1)
      POIS(K)=POIS(K-1)
      FFX(K)=FFX(K-1)
      FFY(K)=FFY(K-1)
   54 CONTINUE
      ITYP=2
      IF(KEL .EQ. 1)ITYP=1
      IF(KCODE .EQ. 0 .OR. KCODE .EQ. -1)GO TO 32
C
C
C     GENERATE SOIL PARAMETERS
C
C
      IF(KEL .NE. 1)GO TO 32
      E(K)=DUN(K,1)*2116.8
      POIS(K)=0.3
   32 CONTINUE
      IF(ICONT .NE. YES)GO TO 67
      IF(KEL .EQ. 1)GO TO 34
      IF(K .EQ. 1)WRITE(7,57)
      WRITE(7,109) K,(DUN(K,I),I=1,9),PA,FFX(K),FFY(K)
      IF(POIS(K) .NE. 0.)WRITE(7,573)K,POIS(K)
      GO TO 67
   34 IF(K .EQ. 1)WRITE(7,935)
      WRITE(7,938)K,E(K),POIS(K),FFY(K)
   67 CONTINUE
      MNO(K)=ITYP
      GO TO 68
C
C     READ ELEMENT DATA AND GENERATE ELEMENTS
C
   69 CONTINUE
      DO 70 IK=1,3
      DO 70 IJ=6,7
   70 DUN(IK,IJ)=DUN(IK,IJ)*0.0174533
      WRITE(6,928)
      IF(IFILE .EQ. 1)GO TO 158
      CALL EDIT7(ICONT)
      GO TO 159
  158 READ(5,6)ICONT
      IF(IFILE .EQ. 1)WRITE(8,6)ICONT
  159 NUMMAT=K
      MB=0
      NE=1
      G1=QQ
      CALL AUTO(4,NUMNP,K,NUMEL,ANGLE)
      IF(ICONT .EQ. YES)WRITE(7,901)
      WRITE(4,*)NUMEL
      DO 401 NE=1,NUMEL
C
```

```
C     WRITE LM ARRAY ON FILE 'XYDATA' FOR FEM GRID PLOT
C
      WRITE(4,*)(LM(NE,K),K=1,4)
      IF(ICONT .EQ. YES)WRITE(7,903)NE,(LM(NE,K),K=1,6),ARE(NE),
     +                              TTX(NE),TTY(NE)
      IF(ARE(NE) .EQ. 0.)ARE(NE)=0.001
      MX=LM(NE,6)
      ITYP=MNO(MX)
      IF(ITYP.NE.1) GO TO 301
C
C     MODULUS AND POISSON RATIO OF ELASTIC SOIL
C
      EX(NE)=E(MX)
      GX(NE)=POIS(MX)
      GO TO 401
C
C IN INCR=1 AND ITERATION ZERO CALCULATE STRESS FROM DEPTH OF SOIL
C
  301 IF(LM(NE,3) .EQ. 0 )GO TO 401
      XC=0.
      YC=0.
      DO 194 J=1,4
      N=LM(NE,J)
      XC=XC+0.25*X(N)
  194 YC=YC+0.25*Y(N)
      HC=0.
      HCPP=0.
      SY=0.
      DO 195 N=1,75
      DO 196 JJ=1,4
      J=JJ
      K=J+1
      IF(K .GT. 4)K=1
      J=IABS(LM(N,J))
      K=IABS(LM(N,K))
      XJ=X(J)
      XK=X(K)
      IF(ABS(XJ-XK) .LE. 0.001)GO TO 196
      IF(XC .GT. XJ .AND. XC .GT. XK)GO TO 196
      IF(XC .LT. XJ .AND. XC .LT. XK)GO TO 196
      HCP=Y(J)+(XC-XJ)/(XK-XJ)*(Y(K)-Y(J))-YC
      IF(HCP .GT. HC)HC=HCP
  196 CONTINUE
      SY=-(HC-HCPP)*FFY(LM(N,6))+SY
      HCPP=HC
  195 CONTINUE
      GU=0.3
      SX=GU*SY/(1.0-GU)
      SDD=SY-SX
      MX=LM(NE,6)
C
C     ESTIMATE NONLINEAR SOIL MODULUS AND POISSON RATIO
C
```

```
      CALL DUNT(SX,SDD,EZ,GU,MX)
      EX(NE)=EZ
      GX(NE)=GU
      IF(POIS(MX) .NE. 0.)GX(NE)=POIS(MX)
  401 CONTINUE
      CLOSE(UNIT=4,STATUS='KEEP')
C
C     OPTIONS FOR OUTPUT
C
C     NODAL DISPLACEMENTS
C
      WRITE(6,929)
      IF(IFILE .EQ. 1)GO TO 160
      CALL EDIT7(ICONTD)
      GO TO 161
  160 READ(5,6)ICONTD
      IF(IFILE .EQ. 1)WRITE(8,6)ICONTD
C
C     ELEMENT STRAINS AND STRESSES
C
  161 WRITE(6,930)
      IF(IFILE .EQ. 1)GO TO 162
      CALL EDIT7(ICONTE)
      GO TO 163
  162 READ(5,6)ICONTE
      IF(IFILE .EQ. 1)WRITE(8,6)ICONTE
C
C     ELEMENT STRAINS AND STRESSES ALONG THE SIDES
C
  163 IF(ICONTE .NE. YES)GO TO 579
      WRITE(6,908)
      IF(IFILE .EQ. 1)GO TO 164
      CALL EDIT7(IOPTE)
      GO TO 579
  164 READ(5,6)IOPTE
      IF(IFILE .EQ. 1)WRITE(8,6)IOPTE
  579 CONTINUE
C
C     CLOSE FILES
C
      IF(IFILE .EQ. 1)CLOSE(UNIT=8,STATUS='KEEP')
      IF(IFILE .NE. 1)CLOSE(UNIT=9,STATUS='KEEP')
      IF(IFILE .NE. 1)CLOSE(UNIT=10,STATUS='KEEP')
C
C     INCREMENT STARTS
C
C     FOR ANALYSIS WITH TWO INCREMENTS (INR=1)
C     IN INCR=1 FOUNDATION SOIL STRESSES ARE CALCULATED
C     WITH NO DISPLACEMENTS
C     IN INCR=2 BACKFILL AND STRUCTURES ARE ADDED
C
      INCR=0
C
```

```
C     INITIALIZE TOTAL ACCUMULATED STRESSES
C
      DO 403 II=1,102
      DO 403 N=1,5
      XXS(II,N)=0.
      YYS(II,N)=0.
      ZZS(II,N)=0.
      XYT(II,N)=0.
      XZT(II,N)=0.
  403 YZT(II,N)=0.
  402 INCR=INCR+1
      IF(INR .NE. 1 .AND. INCR .EQ. 2)GO TO 1000
      IF(INR .EQ. 1 .AND. INCR .EQ. 3)GO TO 1000
      IF(INR .NE. 1)GO TO 417
C
C     NUMBER OF ELEMENTS IS SET
C
      IF(INCR .EQ. 1)NUMEL=55
      IF(INCR .EQ. 2)NUMEL=100
      GO TO 418
  417 NUMEL=100
  418 CONTINUE
C
C     BOUNDARY CONDITIONS CHANGED
C
      IF(INR .NE. 1)GO TO 416
      IF(INCR .EQ. 1)GO TO 404
C
C     RESTORE BOUNDARY CONDITIONS AND IEQ ARRAY IN INCR=2
C
      NEQ=NNEQ
      DO 411 I=1,100
      DO 411 J=1,6
      IEQ(I,J)=IIQ(I,J)
  411 ID(I,J)=IID(I,J)
      GO TO 416
C
C     BOUNDARY CONDITIONS IN INCR=1
C
  404 CONTINUE
      DO 405 I=1,24
      DO 405 J=1,6
  405 ID(I,J)=1
      DO 410 J=4,6
      DO 406 I=27,31
  406 ID(I,J)=1
      DO 407 I=35,47,12
  407 ID(I,J)=1
      DO 408 I=40,76,12
  408 ID(I,J)=1
      DO 409 I=42,78,12
  409 ID(I,J)=1
  410 CONTINUE
```

```
      DO 420 I=76,78,2
      DO 419 J=1,3
  419 ID(I,J)=0
  420 CONTINUE
      DO 421 I=36,84,12
  421 ID(I,2)=0
      DO 422 I=61,73,12
  422 ID(I,2)=0
      DO 423 I=86,95
  423 ID(I,1)=0
C
C     CHANGE ID ARRAY INTO GLOBAL EQUATION NUMBERS IN INCR=1
C
      NEQ=0
      IF(ICNEQ .EQ. YES)WRITE(7,3)INCR
      DO 414 N=1,NUMNP
      DO 412 I=1,6
C
C     SKIP TRIVIAL EQUATIONS
C
      IF(ID(N,I).GT.0) GO TO 413
      NEQ=NEQ+1
      IEQ(N,I)=NEQ
      GO TO 412
  413 IEQ(N,I)=0
  412 CONTINUE
      IF(ICNEQ .NE. YES)GO TO 414
      WRITE(7,26) N,X(N),Y(N),(ID(N,I),I=1,6),(IEQ(N,I),I=1,6)
  414 CONTINUE
      WRITE(7,4) NEQ,INCR
      IF(NEQ .GT. NUNKN)WRITE(6,223)
  416 CONTINUE
C
C     ITERATION STARTS
C
      DO 201 I=1,NUNKN
  201 DGP(I)=0.
      ITNO=0
  202 ITNO=ITNO+1
      ISTOP=0
      DO 531 II=1,NUNKN
      DO 521 JJ=1,NBAND
  521 GK(II,JJ)=0.
  531 GP(II)=GGP(II)
C
C     NO GIRDER LOADING IN INCR=1
C
      IF(INR .NE. 1)GO TO 533
      IF(INCR .EQ. 2)GO TO 533
      DO 532 II=1,NUNKN
  532 GP(II)=0.
  533 CONTINUE
C
```

```
C     ADD ELEMENT STIFFNESS AND LOAD TO GLOBAL ASSEMBLY FOR COMPACTED
C     STORAGE
C
      DO 300 NE=1,NUMEL
      FX=FFX(LM(NE,6))
      FY=FFY(LM(NE,6))
      TX=TTX(NE)
      TY=TTY(NE)
      IRE=0
C
C     ELEMENT STIFFNESS MATRIX AND LOAD VECTOR ARE CALULATED
C     FROM SUBROUTINE
C
      L=4
      IF(LM(NE,3) .EQ. 0)L=2
      GO TO (85,85,95,95),L
   85 CALL BECOL(IRE,NE,NDEL,NDFPN,ISTOP)
      GO TO 96
   95 CALL QUAD(IRE,NE,NDEL,NDFPN,ISTOP)
   96 I=0
      DO 300 NDI=1,NDEL
      NODEI=LM(NE,NDI)
      DO 300 NFI=1,NDFPN
      I=I+1
      IG=IEQ(NODEI,NFI)
      IF(IG.EQ.0) GO TO 300
      J=0
      DO 100 NDJ=1,NDEL
      NODEJ=LM(NE,NDJ)
      DO 100 NFJ=1,NDFPN
      J=J+1
      JG=IEQ(NODEJ,NFJ)
      IF(JG.LT.IG) GO TO 100
      JGC=JG-IG+1
      IF(JGC.GT.MB) MB=JGC
      IF(MB .GT. NBAND)WRITE(6,222)
      GK(IG,JGC)=GK(IG,JGC)+EK(I,J)
  100 CONTINUE
      GP(IG)=GP(IG)+EP(I)
  300 CONTINUE
      IF(ITNO.GT.1) GO TO 501
      WRITE(7,5) MB,INCR
C
C     SOLVE FOR UNKNOWN DISPLACEMENTS(COMPACTED SOLVER)
C
  501 IFLAG=0
      CALL CSOLVE(NEQ,MB,200,100,IFLAG)
C
C      DISPLACEMENTS AT NODES
C
      DO 560 N=1,NUMNP
      DO 560 I=1,6
      NQ=IEQ(N,I)
```

```
      IF(NQ.EQ.0) GO TO 550
      DISN(N,I)=GP(NQ)
      GO TO 560
  550 DISN(N,I)=0.
  560 CONTINUE
C
C     NEED FOR ADDITIONAL ITERATION CHECKED
C     JSTOP=0 FOR  LINEAR SOILS ONLY
C     ISTOP=1 ITERATION TERMINATES
C
      JSTOP=0
      DO 661 I=1,NUMEL
      ITYP=MNO(LM(I,6))
  661 IF(ITYP.NE.1) JSTOP=1
      IF(JSTOP.EQ.0) ISTOP=1
      ERR=0.
      DU1=0.
C
C     ERROR DURING ITERATION - SECOND VECTOR NORM
C
      DO 203 I=1,NEQ
      ERR=(GP(I)-DGP(I))**2+ERR
  203 DU1=DU1+GP(I)**2
      IF(DU1.EQ.0.) DU1=0.0000001
      ERR=SQRT(ERR/DU1)
      IF(JSTOP .EQ. 0)GO TO 939
      IF(ERR.LT.ERMAX) GO TO 207
      IF(ITNO.EQ.ITMAX) GO TO 205
      DO 204 I=1,NEQ
  204 DGP(I)=GP(I)
      ISTOP=0
      GO TO 939
  205 WRITE(7,206) ITNO,ERR
      ISTOP=1
      GO TO 939
  207 WRITE(7,208) ITNO,ERR
      ISTOP=1
  939 CONTINUE
C
C     PRINT DISPLACEMENTS AT NODES
C
      IF(ISTOP .EQ. 0)GO TO 577
      IF(ICONTD .NE. YES)GO TO 578
      WRITE(7,575)
      IF(INR .EQ. 1 .AND. INCR .EQ. 1)WRITE(7,933)
      DO 576 N=1,NUMNP
  576 WRITE(7,580)N,(DISN(N,I),I=1,6)
  578 CONTINUE
C
C     HORIZONTAL ABUTMENT DISPLACEMENT IN INCR=2
C
      IF(INR .EQ. 1 .AND. INCR .EQ. 1)GO TO 577
      WRITE(7,932)DISN(1,1),DISN(13,1),DISN(31,1)
```

```
  577 CONTINUE
C
C     RECOVERY OF ELEMENT STRAINS AND STRESSES (MOMENT AND THRUST)
C
      IF(ISTOP .EQ. 1 .AND. ICONTE .EQ. YES)WRITE(7,350)
      DO 650 NE=1,NUMEL
      IRE=1
C
C     ELEMENT STRAINS AND STRESSES ARE CALCULATED FROM SUBROUTINE
C
      L=4
      IF(LM(NE,3) .EQ. 0)L=2
      GO TO (610,610,630,630),L
  610 IF(ISTOP .EQ. 0)GO TO 650
      CALL BECOL(IRE,NE,NDEL,NDFPN,ISTOP)
      GO TO 650
  630 CALL QUAD(IRE,NE,NDEL,NDFPN,ISTOP)
C
C     CALCULATE REINFORCEMENT STRESS
C
      IF(LM(NE,5).NE.1) GO TO 646
      EPS=0.
      DO 645 J=1,6
  645 EPS=EPS+BA(J,1)*STR(1,J)
      RSTRES=EPS*EST
      IF(ISTOP.EQ.0) GO TO 632
      IF(ICONTE .EQ. YES)WRITE(7,642)RSTRES
      IF(ISTOP.EQ.1) GO TO 219
C
C     CALCULATE SOIL STRESS IN LOCAL COORDINATES
C
  632 CONTINUE
      DO 647 I=1,6
      SG(I)=0.
      DO 647 J=1,6
  647 SG(I)=SG(I)+BAP(J,I)*SIGMA(J)
      SG(1)=SG(1)-RSTRES*ALP
C
C     CALCULATE SOIL STRESS IN GLOBAL COORDINATES
C
      DO 648 I=1,6
      SIGMA(I)=0.
      DO 648 J=1,6
  648 SIGMA(I)=SIGMA(I)+BA(I,J)*SG(J)
  646 CONTINUE
      CALL PRST(SIGMA,SGM1,SGM3,SDEV)
C
C     CALCULATE NEW MODULUS AND POISSON'S RATIO OF CONTINUUM
C
      M=LM(NE,6)
      ITYP=MNO(M)
      IF(ITYP.EQ.1) GO TO 219
      CALL DUNT(SGM3,SDEV,EXX,GXX,M)
```

```
      EX(NE)=EX(NE)+RELAX*(EXX-EX(NE))
      GX(NE)=GXX
      IF(POIS(M).NE.0.) GX(NE)=POIS(M)
  219 CONTINUE
  650 CONTINUE
      IF(ISTOP.EQ.1) GO TO 209
      GO TO 202
  209 CONTINUE
C
C     THRUST, SHEAR, BENDING MOMENT OF ABUTMENT STEM
C
      IF(INR .EQ. 1 .AND. INCR .EQ. 1)GO TO 402
      WRITE(7,653)
C
C     AXIAL THRUST OF ABUTMENT STEM AT TIE-ROD LEVEL
C
      TMAX=EF(2,7)
      IF(ABS(EF(3,7)) .GT. ABS(EF(2,7)))TMAX=EF(3,7)
      WRITE(7,654)TMAX
C
C     SHEAR FORCE AT TIE-ROD LEVEL
C
      TM1=ABS(EF(2,9))
      TM2=ABS(EF(3,3))
      TMAX=TM1
      IF(TM2 .GT. TMAX)TMAX=TM2
      WRITE(7,656)TMAX
C
C     BENDING MOMENT AT TIE-ROD LEVEL
C
      WRITE(7,657)EF(3,5)
C
C     AXIAL THRUST AT THE BASE
C
      WRITE(7,659)EF(4,7)
C
C     SHEAR FORCE AT THE BASE
C
      TM1=ABS(EF(4,9))
      WRITE(7,660)TM1
C
C     BENDING MOMENT AT THE BASE
C
      TM1=-EF(4,11)
      WRITE(7,662)TM1
      GO TO 402
 1000 WRITE(6,976)
      RETURN
      END
C
C
      SUBROUTINE CSOLVE(NEQ,MB,NEQD,MBD,IFLAG)
C
```

```
C
C     THIS SUBROUTINE SOLVES THE SYSTEM  A  X = B  BY GAUSS ELIMINATION
C     IFLAG=0, TRIANGULARIZE A, FORWARD REDUCE B, BACKSUBSTITUTE
C     IFLAG=1, FORWARD REDUCE B, BACKSUBSITITUTE
C
      COMMON /BLK9/GK(323,100),GP(323)
      NQ=NEQ-1
      DO 300 N=1,NQ
      N1=N+1
      NM1=N+MB-1
      DO 200 I=N1,NM1
      IF(I.GT.NEQ) GO TO 300
C
C     FORWARD REDUCTION
C
      IS=I-N+1
      C=GK(N,IS)/GK(N,1)
      GP(I)=GP(I)-GP(N)*C
      IF(IFLAG.EQ.1) GO TO 200
C
C     TRIANGULARIZATION  A
C
      DO 100 J=I,NM1
      J1=J-I+1
      J2=J-N+1
  100 GK(I,J1)=GK(I,J1)-C*GK(N,J2)
  200 CONTINUE
  300 CONTINUE
C
C     BACKSUBSTITUTION
C
      GP(NEQ)=GP(NEQ)/GK(NEQ,1)
      DO 500 K=2,NEQ
      NI=NEQ-K+1
      NI1=NI+1
      NIM1=NI+MB-1
      SUM=0.
      DO 400 J=NI1,NIM1
      IF(J.GT.NEQ) GO TO 500
      JS=J-NI+1
  400 SUM=SUM+GK(NI,JS)*GP(J)
  500 GP(NI)=(GP(NI)-SUM)/GK(NI,1)
      RETURN
      END
C
C
      SUBROUTINE BECOL(IRE,NE,NDEL,NDFPN,ISTOP)
C
C
C     THIS SUBROUTINE CALCULATES THE ELEMENT STIFFNESS MATRIX AND
C     LOAD VECTOR OF BEAM ELEMENT.  ALSO RECOVERS ELEMENT FORCES AND
C     MOMENTS
C
```

```
      COMMON /BLK1/AE(10),EIY(10),E(10),POIS(10),EX(102),GX(102),INR
      COMMON /BLK17/EIZ(10),GJX(10),IPILE,XPILE,INCR,XXS(102,5),
     +              YYS(102,5),IFILE
      COMMON /BLK12/ZZS(102,5),XYT(102,5),XZT(102,5),YZT(102,5)
      COMMON /BLK3/LM(102,6),X(100),Y(100),ARE(102),EF(4,12)
      COMMON /BLK4/FX,FY,IOPT,EP(12),EK(12,12),TX,TY,ICONT,IOPTE,ICONTE
      COMMON /BLK5/DISN(100,6),SIGMA(6)
      COMMON /BLK8/A1,A2,A3,A4,A5,A6,B1,B2,C1,C2,C3,C4,C5,C6,C7,C8,C9,
     +            C10,C11,C12,C13,D1,D2,E1,E2,E3,E4,E5,E6,E7,F1,F2,F3,
     +            F4,F5,G1,SSH,DDA,DDM,D3,D4
      DIMENSION T(12,12),U(12),F(12),TEM(12,12)
      INTEGER PNE
      CHARACTER ICONT*1 ,IOPT*1,ICONTE*1,IOPTE*1
      CHARACTER*1 YES/'Y'/
C
C     FORMAT STATEMENTS
C
   30 FORMAT(//45X,'BENDING ELEMENT FORCES AND MOMENTS',//,
     +   2X,'EL.NO.',5X,'1-P1',6X,'1-P2',6X,'1-P3',6X,'1-M1',6X,
     +   '1-M2',6X,'1-M3',6X,'2-P1',6X,'2-P2',6X,'2-P3',6X,'2-M1',6X,
     +   '2-M2',6X,'2-M3')
   50 FORMAT(I7,12E10.3)
C
C     NO. OF NODES PER ELEMENT AND DEGREE OF FREEDOM SET
C
      NDEL=2
      NDFPN=6
C
C
      DO 4 J=1,12
      DO 2 K=1,12
    2 EK(J,K)=0.
    4 EP(J)=0.
      N1=LM(NE,1)
      N2=LM(NE,2)
      MT=LM(NE,6)
      ANG=ARE(NE) *0.0174533
      XI=X(N1)
      XJ=X(N2)
      YI=Y(N1)
      YJ=Y(N2)
      XL=XJ-XI
      YL=YJ-YI
      XS=XL**2
      YS=YL**2
      SXY=XS+YS
      SL=SQRT(SXY)/COS(ANG)
      CO=XL/SL
      SI=YL/SL
C
C     ELEMENT STIFFNESS MATRIX (LOCAL COORDINATES)
C
      EML=EIY(MT)/SL
```

```
      EMM=EIZ(MT)/SL
      EMN=GJX(MT)/SL
      S=AE(MT)/SL
C
C     IF THE TOP ELEVATION OF THE DRILLED-IN CONCRETE ANCHOR IS NOT
C     AT THE GROUND SURFACE, SET PROPERTIES OF ELEMENT 84 VERY SMALL
C
      IF(D4 .GT. 0. .AND. NE .EQ. 84)GO TO 11
      GO TO 12
   11 EML=EML/100000000.
      EMM=EMM/100000000.
      EMN=EMN/100000000.
      S=S/100000000.
   12 CONTINUE
      A11=S
      A22=12.*EMM/SL**2
      A26=6.*EMM/SL
      A33=12.*EML/SL**2
      A35=-6.*EML/SL
      A44=EMN
      A55=4.*EML
      A511=2.*EML
      A66=4.*EMM
      A612=2.*EMM
      EK(1,1)=A11
      EK(1,7)=-A11
      EK(2,2)=A22
      EK(2,6)=A26
      EK(2,8)=-A22
      EK(2,12)=A26
      EK(3,3)=A33
      EK(3,5)=A35
      EK(3,9)=-A33
      EK(3,11)=A35
      EK(4,4)=A44
      EK(4,10)=-A44
      EK(5,5)=A55
      EK(5,9)=-A35
      EK(5,11)=A511
      EK(6,6)=A66
      EK(6,8)=-A26
      EK(6,12)=A612
      EK(7,7)=A11
      EK(8,8)=A22
      EK(8,12)=-A26
      EK(9,9)=A33
      EK(9,11)=-A35
      EK(10,10)=A44
      EK(11,11)=A55
      EK(12,12)=A66
      DO 20 I=2,12
      DO 10 J=1,11
      IF(J.GE.I) GO TO 20
```

```
      EK(I,J)=EK(J,I)
   10 CONTINUE
   20 CONTINUE
C
C     ROTATION MATRIX
C
      DO 5 I=1,12
      DO 5 J=1,12
    5 T(I,J)=0.
      CX=XL/SL
      CY=YL/SL
      CZ=SIN(ANG)
      CON=SQRT(CX**2+CY**2)
      T21=-CX*CZ/CON
      T22=-CY*CZ/CON
      T31=CY/CON
      T32=-CX/CON
      T(1,1)=CX
      T(1,2)=CY
      T(1,3)=CZ
      T(2,1)=T21
      T(2,2)=T22
      T(2,3)=CON
      T(3,1)=T31
      T(3,2)=T32
      DO 6 K=3,9,3
      DO 6 I=1,3
      DO 6 J=1,3
    6 T(I+K,J+K)=T(I,J)
      DO 7 I=1,12
      DO 7 J=1,12
      TEM(I,J)=0.
      DO 7 K=1,12
    7 TEM(I,J)=TEM(I,J)+EK(I,K)*T(K,J)
      IF(IRE .NE. 0)GO TO 21
C
C     ELEMENT STIFFNESS MATRIX (GLOBAL COORDINATES)
C
      DO 8 I=1,12
      DO 8 J=I,12
      EK(I,J)=0.
      DO 8 K=1,12
      Q=T(K,I)*TEM(K,J)
    8 EK(I,J)=EK(I,J)+Q
      DO 9 I=2,12
      DO 9 J=1,I-1
    9 EK(I,J)=EK(J,I)
      GO TO 60
C
C     RECOVERY OF ELEMENT FORCES AND MOMENTS
C
   21 CONTINUE
      DO 70 I=1,6
```

```
      U(I)=DISN(N1,I)
   70 U(I+6)=DISN(N2,I)
      DO 80 I=1,12
      F(I)=0.
      DO 80 K=1,12
   80 F(I)=F(I)+TEM(I,K)*U(K)
      PNE=NE-1
      IF(ICONTE .NE. YES)GO TO 45
      IF(LM(NE,3).EQ.LM(PNE,3)) GO TO 40
      WRITE(7,30)
   40 WRITE(7,50) NE, (F(I),I=1,12)
   45 CONTINUE
C
C     STORE MAXIMUM AXIAL FORCE, SHEAR, MOMENT OF THE ABUTMENT STEM
C
      IF(NE .GT. 79)GO TO 60
      DO 65 I=1,12
   65 EF(NE-75,I)=F(I)
   60 RETURN
      END
C
C
      SUBROUTINE QUAD(IRE,NE,NDEL,NDFPN,ISTOP)
C
C
C     THIS SOUBROUTINE CALCULATES ELEMENT STIFFNESS MATRIX AND
C     LOAD VECTOR, AND THE RECOVERY OF ELEMENT STRAINS AND STRESSES
C     OF 4-NODAL QUADRILATERAL ELEMENT
C
      COMMON /BLK1/AE(10),EIY(10),E(10),POIS(10),EX(102),GX(102),INR
      COMMON /BLK17/EIZ(10),GJX(10),IPILE,XPILE,INCR,XXS(102,5),
     +              YYS(102,5),IFILE
      COMMON /BLK12/ZZS(102,5),XYT(102,5),XZT(102,5),YZT(102,5)
      COMMON /BLK2/XQ(4),YQ(4),B(6,12),XN(4),ALP,EST,PHP,PHR,BA(6,6),
     +             BAP(6,6)
      COMMON /BLK3/LM(102,6),X(100),Y(100),ARE(102),EF(4,12)
      COMMON /BLK4/FX,FY,IOPT,EP(12),EK(12,12),TX,TY,ICONT,IOPTE,ICONTE
      COMMON /BLK5/DISN(100,6),SIGMA(6)
      COMMON /BLK6/SMX(5),SMY(5),SMZ(5),TXY(5),TXZ(5),TYZ(5),STR(5,6)
      DIMENSION U(4),V(4),SA(4),TA(4),SR(5),TR(5)
      DIMENSION RE(5),C(6,6),CB(6,12),W(4),TEM(6,6)
      CHARACTER ICONT*1 ,IOPT*1,IOPTE*1,ICONTE*1
      CHARACTER*1 YES/'Y'/
      DATA RE/3HCEN,3HI-J,3HJ-K,3HK-L,3HL-I/
C
C     FORMAT STATEMENTS
C
  550 FORMAT(I7,3X,A3,1X,11E10.3)
  551 FORMAT(10X,A3,1X,11E10.3)
  610 FORMAT(8X,'**ELEMENT',I4,1X,'IS REINFORCED')
C
C     NO. OF NODES PER ELEMENT AND DEGREE OF FREEDOM SET
C
```

```
      NDEL=4
      NDFPN=3
C
C
      MT=LM(NE,6)
      EM=EX(NE)
      POI=GX(NE)
C
C     STRESS-STRAIN RELATIONSHIP
C
      DO 5 I=1,6
      DO 5 J=1,6
      TEM(I,J)=0.
    5 C(I,J)=0.
      A1=1./EM
      A2=-POI*A1
      A3=2.*(1.+POI)/EM
      IF(LM(NE,5).NE.1) GO TO 10
C
C     REINFORCED SOIL ELEMENT
C
      CALL PROP(EM,POI,NE,A3,C)
C
C     ROTATE TO GLOBAL COORDINATES
C
      DO 60 I=1,6
      DO 60 J=1,6
      DO 60 K=1,6
   60 TEM(I,J)=TEM(I,J)+BA(I,K)*C(K,J)
      DO 70 I=1,6
      DO 70 J=1,6
      C(I,J)=0.
      DO 70 K=1,6
   70 C(I,J)=C(I,J)+TEM(I,K)*BA(J,K)
      GO TO 20
C
C     UNREINFORCED SOIL ELEMENT
C
   10 C(1,1)=A1
      C(2,2)=A1
      C(1,2)=A2
      C(2,1)=A2
      C(1,3)=A2
      C(2,3)=A2
      C(3,1)=A2
      C(3,2)=A2
      C(3,3)=A1
      C(4,4)=A3
      C(5,5)=A3
      C(6,6)=A3
      CALL STIFIV(C,6,6)
C
C     PSEUDO SOIL PROPERTIES FOR THIN ELEMENTS BEHIND THE ABUTMENT
```

```
C
      IF(NE .EQ. 56)GO TO 6
      IF(NE .EQ. 61)GO TO 6
      IF(NE .EQ. 66)GO TO 6
      IF(NE .EQ. 71)GO TO 6
      GO TO 7
    6 CONTINUE
      C(1,2)=0.01
      C(1,3)=0.01
      C(2,3)=0.01
      C(2,1)=0.01
      C(3,1)=0.01
      C(3,2)=0.01
      C(2,2)=0.01
      C(3,3)=0.01
      C(4,4)=0.01
      C(5,5)=0.01
      C(6,6)=0.01
    7 CONTINUE
   20 CONTINUE
      DO 30 J=1,4
      NP=LM(NE,J)
      XQ(J)=X(NP)
   30 YQ(J)=Y(NP)
      IF(IRE.NE.0) GO TO 210
C
C     ELEMENT STIFFNESS MATRIX
C
      YL=YQ(4)-YQ(3)
      XL=XQ(4)-XQ(3)
      SQL=XL**2+YL**2
      SL=SQRT(SQL)
      SA(1)=1.
      SA(2)=-1.
      SA(3)=-1.
      SA(4)=1.
      TA(1)=1.
      TA(2)=1.
      TA(3)=-1.
      TA(4)=-1.
      DO 50 J=1,12
      DO 40 K=1,12
   40 EK(J,K)=0.
   50 EP(J)=0.
      DO 199 N=1,4
      SAM=3.
      SN=SA(N)/SQRT(SAM)
      TN=TA(N)/SQRT(SAM)
      CALL FORMB(SN,TN,DJ,XS,YS)
      DO 100 I=1,6
      DO 100 J=1,12
      CB(I,J)=0.
      DO 100 K=1,6
```

```
  100 CB(I,J)=CB(I,J)+C(I,K)*B(K,J)
      DO 110 I=1,12
      DO 110 J=I,12
      DO 110 K=1,6
      Q=B(K,I)*CB(K,J)
      EK(I,J)=EK(I,J)+Q/DJ
  110 IF(N .EQ. 4) EK(J,I)=EK(I,J)
C
C     ELEMENT LOAD VECTOR BY BODY FORCE
C
C     FOUNDATION SOIL HAS NO BODY FORCE IN INCR=2
C
      IF(INR .NE. 1)GO TO 202
      IF(INCR .EQ. 2 .AND. NE .LE. 55)GO TO 199
  202 CONTINUE
      DO 200 M=1,4
      L1=3*M-2
      L2=3*M-1
      L3=3*M
      EP(L1)=EP(L1)+XN(M)*FX*DJ
      EP(L2)=EP(L2)+XN(M)*FY*DJ
  200 CONTINUE
  199 CONTINUE
      IF(INR .EQ. 1 .AND. INCR .EQ. 1)GO TO 600
C
C     ELEMENT LOAD VECTOR BY TRACTION
C
      EP(7)=EP(7)+TX*SL/2.
      EP(8)=EP(8)+TY*SL/2.
      EP(10)=EP(10)+TX*SL/2.
      EP(11)=EP(11)+TY*SL/2.
      GO TO 600
  210 CONTINUE
C
C     RECOVERY OF ELEMENT STRAINS AND STRESSES
C
      DO 220 J=1,4
      NP=LM(NE,J)
      U(J)=DISN(NP,1)
      V(J)=DISN(NP,2)
  220 W(J)=DISN(NP,3)
      SR(1)=0.
      SR(2)=0.
      SR(3)=1.
      SR(4)=0.
      SR(5)=-1.
      TR(1)=0.
      TR(2)=-1.
      TR(3)=0.
      TR(4)=1.
      TR(5)=0.
      N=1
  250 SI=SR(N)
```

```
      TI=TR(N)
      SMX(N)=0.
      SMY(N)=0.
      SMZ(N)=0.
      TXY(N)=0.
      TXZ(N)=0.
      TYZ(N)=0.
      CALL FORMB(SI,TI,DJ,XS,YS)
      DO 260 L=1,6
  260 STR(N,L)=0.
      DO 300 M=1,4
      M1=3*M-2
      M2=3*M-1
      M3=3*M
C
C     ELEMENT STRAINS
C
      DO 300 L=1,6
  300 STR(N,L)=STR(N,L)+(B(L,M1)*U(M)+B(L,M2)*V(M)+B(L,M3)*W(M))/DJ
C
C     ELEMENT STRESSES
C
      DO 310 J=1,6
      SMX(N)=SMX(N)+C(1,J)*STR(N,J)
      SMY(N)=SMY(N)+C(2,J)*STR(N,J)
      SMZ(N)=SMZ(N)+C(3,J)*STR(N,J)
      TXY(N)=TXY(N)+C(4,J)*STR(N,J)
      TXZ(N)=TXZ(N)+C(5,J)*STR(N,J)
  310 TYZ(N)=TYZ(N)+C(6,J)*STR(N,J)
C
C     ACCUMULATED STRESSES IN INCR=1
C
      IF(INR .EQ. 1 .AND. INCR .EQ. 2)GO TO 320
      XXS(NE,N)=SMX(N)
      YYS(NE,N)=SMY(N)
      ZZS(NE,N)=SMZ(N)
      XYT(NE,N)=TXY(N)
      XZT(NE,N)=TXZ(N)
      YZT(NE,N)=TYZ(N)
      GO TO 330
C
C     ACCUMULATED STRESSES IN INCR=2
C
  320 SMX(N)=SMX(N)+XXS(NE,N)
      SMY(N)=SMY(N)+YYS(NE,N)
      SMZ(N)=SMZ(N)+ZZS(NE,N)
      TXY(N)=TXY(N)+XYT(NE,N)
      TXZ(N)=TXZ(N)+XZT(NE,N)
      TYZ(N)=TYZ(N)+YZT(NE,N)
  330 CONTINUE
      IF(ISTOP .EQ. 0)GO TO 400
      IF(N .EQ. 1)GO TO 500
      GO TO 510
```

```
  500 IF(ICONTE .EQ. YES)WRITE(7,550)NE,RE(N),(STR(N,I),I=1,2),
     +   (STR(N,I),I=4,6),SMX(N),SMY(N),SMZ(N),TXY(N),TXZ(N),TYZ(N)
      GO TO 555
  510 CONTINUE
      IF(ICONTE .EQ. YES .AND. IOPTE .EQ. YES)
     +                  WRITE(7,551)RE(N),(STR(N,I),I=1,2),
     + (STR(N,I),I=4,6),SMX(N),SMY(N),SMZ(N),TXY(N),TXZ(N),TYZ(N)
  555 CONTINUE
      N=N+1
      IF(N.LE.5) GO TO 250
  400 CONTINUE
C
C     STRESSES AT THE CENTER OF ELEMENT IN MID-INCREMENT
C
      SIGMA(1)=(SMX(1)+XXS(NE,1))/2.
      SIGMA(2)=(SMY(1)+YYS(NE,1))/2.
      SIGMA(3)=(SMZ(1)+ZZS(NE,1))/2.
      SIGMA(4)=(TXY(1)+XYT(NE,1))/2.
      SIGMA(5)=(TXZ(1)+XZT(NE,1))/2.
      SIGMA(6)=(TYZ(1)+YZT(NE,1))/2.
      IF(LM(NE,5).EQ.0) RETURN
      IF(ISTOP.EQ.0) RETURN
      IF(ICONTE .EQ. YES)WRITE(7,610)NE
  600 RETURN
      END
C
C
      SUBROUTINE FORMB(S,T,DJ,XS,YS)
C
C
C     THIS SUBROUTINE CALCULATES 'B' MATRIX OF 4-NODAL QUADRILATERAL
C     ELEMENT
C
      COMMON /BLK2/XQ(4),YQ(4),B(6,12),XN(4),ALP,EST,PHP,PHR,BA(6,6),
     +            BAP(6,6)
      T1=0.25*(1.-T)
      T2=0.25*(1.+T)
      S1=0.25*(1.-S)
      S2=0.25*(1.+S)
C
C     ELEMENT SHAPE FUNCTIONS
C
      XN(1)=T1*(1.-S)
      XN(2)=T1*(1.+S)
      XN(3)=T2*(1.+S)
      XN(4)=T2*(1.-S)
      XS=-T1*XQ(1)+T1*XQ(2)+T2*XQ(3)-T2*XQ(4)
      XT=-S1*XQ(1)-S2*XQ(2)+S2*XQ(3)+S1*XQ(4)
      YS=-T1*YQ(1)+T1*YQ(2)+T2*YQ(3)-T2*YQ(4)
      YT=-S1*YQ(1)-S2*YQ(2)+S2*YQ(3)+S1*YQ(4)
      DJ=XS*YT-XT*YS
      XS1=S1*XS
      XS2=S2*XS
```

```
      YS1=S1*YS
      YS2=S2*YS
      XT1=T1*XT
      XT2=T2*XT
      YT1=T1*YT
      YT2=T2*YT
C
C     DERIVATIVES OF SHAPE FUNCTION
C
      F1=YS1-YT1
      F2=YT1+YS2
      F3=YT2-YS2
      F4=-YT2-YS1
      G1=XT1-XS1
      G2=-XS2-XT1
      G3=XS2-XT2
      G4=XS1+XT2
      DO 10 I=1,6
      DO 10 J=1,12
   10 B(I,J)=0.
      B(1,1)=F1
      B(1,4)=F2
      B(1,7)=F3
      B(1,10)=F4
      B(2,2)=G1
      B(2,5)=G2
      B(2,8)=G3
      B(2,11)=G4
      B(4,1)=G1
      B(4,2)=F1
      B(4,4)=G2
      B(4,5)=F2
      B(4,7)=G3
      B(4,8)=F3
      B(4,10)=G4
      B(4,11)=F4
      B(5,3)=F1
      B(5,6)=F2
      B(5,9)=F3
      B(5,12)=F4
      B(6,3)=G1
      B(6,6)=G2
      B(6,9)=G3
      B(6,12)=G4
      RETURN
      END
C
C
      SUBROUTINE DUNT(S3,SSD,E,GU,M)
C
C
C     THIS SUBROUTINE CALCULATES SOIL TANGENT MODULUS AND
C     POISON'S RATIO BY MODIFIED DUNCAN'S MODEL
```

```
C
      COMMON /BLK7/DUN(10,9),PA
      IF(S3 .LE. 0.)GO TO 30
      IF(S3 .LE. (0.1*PA))S3=0.1*PA
      Z=S3/PA
      PH=DUN(M,6)-DUN(M,7)*ALOG10(Z)
      IF(PH .LE. 0.)PH=0.01
      EI=0.001*DUN(M,1)*PA*0.1**DUN(M,3)
      BT=1.-0.5*(DUN(M,4)*(1.-SIN(PH))*SSD)/
     +    (DUN(M,5)*COS(PH)+S3*SIN(PH))
      IF(BT.LT.0.001) BT=0.001
      BTS=BT
      XKS=DUN(M,1)
      GAMA=XKS*PA*(S3/PA)**DUN(M,3)
      E=BTS*BTS*GAMA
      IF(E .LT. EI)E=EI
      B=DUN(M,8)*PA*(S3/PA)**DUN(M,9)
      IF(B .LE. 0.)GO TO 10
      GU=0.5-E/(6.*B)
      GO TO 20
C
C     SET POISSON RATIO = 0.3 IF BULK MODULUS IS NOT GIVEN
C
   10 GU=0.3
   20 CONTINUE
C
C     SET LIMITS
C
      IF(B.GT.(E*17.)) GU=0.495
      IF(B.LT.(E/3.)) GU=0.001
      RETURN
   30 GU=0.495
      E=0.001*DUN(M,1)*PA*0.1**DUN(M,3)
      RETURN
      END
C
C
      SUBROUTINE PROP(EM,POI,NE,G,C)
C
C
C     THIS SUBROUTINE CALCULATES THE PROPERTIES OF REINFORCED CONTINUUM
C
      COMMON /BLK2/XQ(4),YQ(4),B(6,12),XN(4),ALP,EST,PHP,PHR,BA(6,6),
     +             BAP(6,6)
      DIMENSION C(6,6)
      ALPHA=ALP*EST/EM
      CON=1./EM/(1.+ALPHA)
      DO 5 I=1,6
      DO 5 J=1,6
    5 C(I,J)=0.
      C(1,1)=1.
      C(1,2)=-POI
      C(1,3)=-POI
```

```
      C(2,1)=-POI
      C(2,2)=1.+ALPHA*(1.-POI**2)
      C(2,3)=-POI*(1.+ALPHA*(1.+POI))
      C(3,1)=-POI
      C(3,2)=C(2,3)
      C(3,3)=C(2,2)
      C(4,4)=G
      C(5,5)=G
      C(6,6)=G
      DO 10 I=1,3
      DO 10 J=1,3
   10 C(I,J)=C(I,J)*CON
      CALL STIFIV(C,6,6)
      RETURN
      END
C
C
      SUBROUTINE STIFIV(A,MS,N)
C
C
C     THIC SUBROUTINE INVERT THE MATRIX
C     MS=SIZE OF THE DIMENSION STATEMENT ON THE MATRIX TO BE INVERTED
C     N=SIZE OF THE MATRIX TO BE INVERTED
C
      DIMENSION A(MS,MS),INDEX(12)
      DO 3 I=1,N
    3 INDEX(I)=0
    4 AMAX=-1.
      DO 7 I=1,N
      IF(INDEX(I))7,5,7
    5 TEMP=ABS(A(I,I))
      IF(TEMP-AMAX)7,7,6
    6 ICOL=I
      AMAX=TEMP
    7 CONTINUE
      IF(AMAX)20,21,8
    8 INDEX(ICOL)=1
      PIVOT=A(ICOL,ICOL)
      A(ICOL,ICOL)=1.
      PIVOT=1./PIVOT
      DO 9 J=1,N
    9 A(ICOL,J)=A(ICOL,J)*PIVOT
      DO 12 I=1,N
      IF(I-ICOL) 10,12,10
   10 TEMP=A(I,ICOL)
      A(I,ICOL)=0.
      DO 11 J=1,N
   11 A(I,J)=A(I,J)-A(ICOL,J)*TEMP
   12 CONTINUE
      GO TO 4
   21 WRITE(6,100)
  100 FORMAT(1X,'SINGULAR MATRIX',//)
   20 RETURN
```

```
      END
C
C
      SUBROUTINE PRST(SIGMA,SGM1,SGM3,SDEV)
C
C
C     THIS SUBROUTINE CALCULATES THREE REAL ROOTS OF A RATIONAL
C     CUBIC EQUATION.  THEY ARE THREE PRINCIPAL STRESSES
C
      DIMENSION SIGMA(6)
      DATA ZC,ZD/-0.5,0.866025404/
      S1=SIGMA(1)
      S2=SIGMA(2)
      S3=SIGMA(3)
      S4=SIGMA(4)
      S5=SIGMA(5)
      S6=SIGMA(6)
      B1=-(S1+S2+S3)
      B2=S2*S3+S1*S3+S2*S1-S4**2-S5**2-S6**2
      B3=S1*S6**2+S2*S5**2+S3*S4**2-S1*S2*S3-2.*S4*S5*S6
      T3=1./3.
      Q=2.*(B1*T3)**3-B1*B2*T3+B3
      Q2=Q*ZC
      P=B2-B1*B1*T3
      R=Q2**2+(P*T3)**3
      R2=SQRT(ABS(R))
      A=Q2+R2
      B=Q2-R2
      TERM=-B1*T3
      THETA=ATAN2(R2,Q2)
      RO=(SQRT(Q2*Q2+R2*R2))**T3
      ZA=RO*COS(THETA*T3)
      ZB=RO*SIN(THETA*T3)
C
C     THREE PRINCIPAL STRESSES ( NEGATIVE OF ROOTS )
C
      B1=-(2.0*ZA+TERM)
      B2=-(2.0*(ZA*ZC-ZB*ZD)+TERM)
      B3=-(2.0*(ZA*ZC+ZB*ZD)+TERM)
      SGM1=B1
      IF(B2 .GT. SGM1)SGM1=B2
      IF(B3 .GT. SGM1)SGM1=B3
      SGM3=B1
      IF(B2 .LT. SGM3)SGM3=B2
      IF(B3 .LT. SGM3)SGM3=B3
      SDEV=SGM1-SGM3
      RETURN
      END
C
C
      SUBROUTINE AUTO(IX,NUMNP,K,NUMEL,ANGLE)
C
C
```

```
C     THIS SUBROUTINE SETS UP DATA FOR AUTOMATIC GENERATION
C
      COMMON /BLK1/AE(10),EIY(10),E(10),POIS(10),EX(102),GX(102),INR
      COMMON /BLK17/EIZ(10),GJX(10),IPILE,XPILE,INCR,XXS(102,5),
     +              YYS(102,5),IFILE
      COMMON /BLK12/ZZS(102,5),XYT(102,5),XZT(102,5),YZT(102,5)
      COMMON /BLK3/LM(102,6),X(100),Y(100),ARE(102),EF(4,12)
      COMMON /BLK8/A1,A2,A3,A4,A5,A6,B1,B2,C1,C2,C3,C4,C5,C6,C7,C8,C9,
     +            C10,C11,C12,C13,D1,D2,E1,E2,E3,E4,E5,E6,E7,F1,F2,F3,
     +            F4,F5,G1,SSH,DDA,DDM,D3,D4
      COMMON/BLK10/ID(100,6),PN(6),FFX(10),FFY(10),TTX(102),TTY(102)
      COMMON/BLK13/CD
      CHARACTER   CD*1
C
C     FORMAT STATEMENTS
C
  500 FORMAT(//3X,'**END OF TIE ROD MUST BE BEYOND TIP OF REAR PILE',//)
  501 FORMAT(I1)
  510 FORMAT(//3X,'**FRONT ROW PILES TOO LONG OR TOO MUCH BATTERED',//)
  520 FORMAT(//3X,'**GIRDER SUPPORT MUST BE BETWEEN PILES',//)
  530 FORMAT(//3X,'**FRONT ROW PILE TOO SHORT**',
     +        /3X,'MUST BE AT LEAST',F10.3,' FT',//)
  550 FORMAT(//3X,'**TOP OF CONCRETE ANCHOR MUST BE ABOVE TIE ROD',//)
  560 FORMAT(//3X,'**TIP OF CONCRETE ANCHOR MUST BE BELOW ABUTMENT BASE',
     +       //)
  570 FORMAT(//3X,'**TIP OF CONCRETE ANCHOR MUST BE ABOVE MID-DEPTH OF',
     +           ' REAR PILE',//)
  580 FORMAT(/1X,'PILE PROPERTIES',/,
     +            1X,      'NO OF PILES PER GROUP?',\)
  581 FORMAT(    /1X,'CROSS SECTIONAL AREA IN SQ IN?',\)
  582 FORMAT(    /1X,'MAJOR AXIS MOMENT OF INERTIA IN IN**4?',\)
  583 FORMAT(    /1X,'MINOR AXIS MOMENT OF INERTIA IN IN**4?',\)
  584 FORMAT(    /1X,'PILE MATERIAL MODULUS IN PSI?',\)
  585 FORMAT(    /1X,'PILE GROUP SPACING IN FT?',\)
  590 FORMAT(/1X,'CIRCULAR PILE PROPERTIES',/,
     +            1X,      'NO OF PILES PER GROUP?',\)
  591 FORMAT(    /1X,'OUTSIDE DIAMETER OF PILE IN FT?',\)
  592 FORMAT(    /1X,'INSIDE DIAMETER OF PILE IN FT?',\)
  593 FORMAT(    /1X,'PILE MATERIAL MODULUS IN PSI?',\)
  594 FORMAT(    /1X,'PILE GROUP SPACING IN FT?',\)
  595 FORMAT(/1X,'**ERROR**REENTER IN INTEGER(S)',\)
  596 FORMAT(//1X,'FRONT ROW PILE PROPERTIES')
  597 FORMAT(//1X,'REAR ROW PILE PROPERTIES')
  598 FORMAT(/2X,'CIRCULAR PILES',
     + /3X,'NO OF PILES PER GROUP   =',I10,
     + /3X,'OUTSIDE DIAMETER        =',F10.3,' FT',
     + /3X,'INSIDE DIAMETER         =',F10.3,' FT',
     + /3X,'PILE MATERIAL MODULUS   =',E10.3,' PSI',
     + /3X,'PILE GROUP SPACING      =',F10.3,' FT')
  599 FORMAT(/2X,'GENERAL SHAPE PILE',
     + /3X,'NO OF PILES PER GROUP       =',I10,
     + /3X,'CROSS SECTIONAL AREA        =',F10.3,' IN**2',
     + /3X,'MAJOR AXIS MOMENT OF INERTIA=',F10.3,' IN**4',
```

```
     + /3X,'MINOR AXIS MOMENT OF INERTIA=',F10.3,' IN**4',
     + /3X,'PILE MATERIAL MODULUS       =',E10.3,' PSI',
     + /3X,'PILE GROUP SPACING          =',F10.3,' FT')
  600 FORMAT(//1X,'ABUTMENT STEM THICKNESS  =',F10.3,' FT')
  601 FORMAT(/1X,'ABUTMENT BASE THICKNESS  =',F10.3,' FT')
  602 FORMAT(/1X,'TIE-ROD MATERIAL MODULUS =',E10.3,' PSI')
C
C     NODAL COORDINATES
C
      GO TO(10,60,90,120),IX
   10 NUMNP=96
      DO 12 I=1,NUMNP
      X(I)=0.
   12 Y(I)=0.
      DO 15 K=1,6
      DO 15 I=1,6
      KK=(K-1)*6+I
      X(KK)=(I-1)*0.25*E2
      IF(I .EQ. 6)X(KK)=1.5*E2
      IF(K .EQ. 1)Y(KK)=A1
      IF(K .EQ. 2)Y(KK)=A1-E1/2.
      IF(K .EQ. 3)Y(KK)=A1-E1
      IF(K .EQ. 4)Y(KK)=(A1-E1)/2.
      IF(I .EQ. 2)X(KK)=0.05*E2
      IF(I .EQ. 3)X(KK)=E2/3.
      IF(I .EQ. 4)X(KK)=2.*E2/3.
C
C     TOP ELEVATION OF DRILLED-IN CONCRETE ANCHOR
C
      IF(D4 .GT. 0.)Y(11)=Y(5)-D4
   15 CONTINUE
      X(25)=-(A3+D1)
      X(26)=-(A3+D1/2.)
      X(27)=-A3
      X(28)=-(A3-C4)
      X(29)=-(A3-B1)
      X(30)=-C8
      X(90)=X(30)+C10*C7/SQRT(1.+C7**2)*XPILE
      Y(90)=-C10/SQRT(1.+C7**2)*XPILE
      X(88)=X(28)-C9*C3/SQRT(1.+C3**2)*XPILE
      Y(88)=-C9/SQRT(1.+C3**2)*XPILE
      X(64)=(X(28)+X(88))/2.
      Y(64)=Y(88)/2.
C
C     TIP OF PILE COORDINATES
C
      X(76)=X(28)-C9*C3/SQRT(1.+C3**2)
      Y(76)=-C9/SQRT(1.+C3**2)
      X(40)=X(28)+(X(64)-X(28))/3.
      Y(40)=Y(64)/3.
      X(52)=X(28)+(X(64)-X(28))*2./3.
      Y(52)=Y(64)*2./3.
      X(61)=4.*X(25)
```

```
      TEM=2.*X(88)
      IF(TEM .LT. X(61))X(61)=TEM
      Y(61)=(X(61)-X(25))/D2
      X(85)=X(61)
      Y(85)=Y(88)
      X(73)=X(61)
      Y(73)=(Y(61)+Y(85))/2.
      X(37)=X(25)+(X(61)-X(25))/3.
      Y(37)=Y(61)/3.
      X(49)=X(25)+(X(61)-X(25))*2./3.
      Y(49)=Y(61)*2./3.
C
C     PROVISION FOR SLOPING BERM IN FRONT OF THE ABUTMENT
C
      IF(X(61) .GE. (X(25)-D3))GO TO 20
      X(49)=X(25)-D3
      Y(49)=(X(49)-X(25))/D2
      X(37)=(X(25)+X(49))/2.
      Y(37)=(Y(25)+Y(49))/2.
      Y(61)=Y(49)
      IF(Y(64) .LT. Y(61))GO TO 20
      IF(Y(61) .GT. Y(76))GO TO 18
      PL=-Y(61)/SQRT(1.+C3**2)
      WRITE(6,530)PL
      STOP
   18 CONTINUE
      Y(64)=(Y(61)+Y(76))/2.
   20 CONTINUE
      DO 17 I=38,74,12
      X(I)=X(I+2)+(X(I-1)-X(I+2))*2./3.
   17 Y(I)=Y(I+2)+(Y(I-1)-Y(I+2))*2./3.
      X(86)=X(88)+(X(85)-X(88))/3.*2.
      Y(86)=Y(88)
      DO 19 I=39,75,12
      X(I)=X(I+1)+(X(I-2)-X(I+1))/3.
   19 Y(I)=Y(I+1)+(Y(I-2)-Y(I+1))/3.
      X(87)=X(88)+(X(85)-X(88))/3.
      Y(87)=Y(88)
      X(89)=(X(88)+X(90))/2.
      Y(89)=(Y(88)+Y(90))/2.
      X(65)=(X(29)+X(89))/2.
      Y(65)=Y(89)/2.
      X(41)=X(29)+(X(65)-X(29))/3.
      Y(41)=Y(65)/3.
      X(53)=X(29)+(X(65)-X(29))*2./3.
      Y(53)=Y(65)*2./3.
      DO 21 I=1,3
      Y(12*I+30)=Y(90)*I/6.
   21 Y(12*I+31)=Y(90)*I/6.
C
C     TIP OF PILE COORDINATES
C
      Y(78)=-C10/SQRT(1.+C7**2)
```

```
      Y(79)=Y(78)
      X(66)=(X(30)+X(90))/2.
      X(78)=X(30)+C10*C7/SQRT(1.+C7**2)
      X(77)=(X(76)+X(78))/2.
      Y(77)=(Y(76)+Y(78))/2.
      X(42)=X(30)+(X(66)-X(30))/3.
      X(54)=X(30)+(X(66)-X(30))*2./3.
      X(96)=E2*1.5
      X(91)=X(90)+(X(96)-X(90))/6.
      X(67)=X(91)/2.
      X(43)=X(67)/3.
      X(55)=X(43)*2.
      X(79)=(X(67)+X(91))/2.
      Y(91)=Y(90)
      Y(96)=Y(90)
      DO 23 I=92,95
      X(I)=X(90)+(X(96)-X(90))*(I-90)/6.
   23 Y(I)=Y(90)
      DO 25 I=44,48
      X(I)=X(I-12)+(X(I+48)-X(I-12))/6.
   25 Y(I)=Y(90)/6.
      X(47)=X(35)
      Y(47)=Y(17)-F2
      DO 27 I=56,60
      X(I)=X(I-24)+(X(I+36)-X(I-24))/3.
   27 Y(I)=Y(90)/3.
      DO 29 I=68,72
      X(I)=(X(I-36)+X(I+24))/2.
   29 Y(I)=Y(90)/2.
      Y(59)=(Y(47)+Y(71))/2.
      DO 31 I=80,84
      X(I)=(X(I-12)+X(I+12))/2.
   31 Y(I)=Y(90)*2./3.
C
C     CHECK ANY VIOLATIONS
C
      IF(X(95) .GT. X(90))GO TO 35
      WRITE(6,500)
      STOP
   35 IF(X(88) .GT. X(85))GO TO 37
      WRITE(6,510)
      STOP
   37 IF(X(30) .GT. X(29))GO TO 39
      WRITE(6,520)
      STOP
   39 IF(X(29) .GT. X(28))GO TO 41
      WRITE(6,520)
      STOP
   41 IF(Y(61) .GT. Y(88))GO TO 43
      PL=-Y(88)/1.5/SQRT(1.+C3**2)
      WRITE(6,530)PL
      STOP
   43 IF(Y(11) .GT. Y(17))GO TO 47
```

```
      WRITE(6,550)
      STOP
   47 IF(Y(35) .GT. Y(47))GO TO 49
      WRITE(6,560)
      STOP
   49 IF(Y(47) .GT. Y(71))GO TO 51
      WRITE(6,570)
      STOP
   51 RETURN
C
C     SET BOUNDARY CONDITIONS
C
   60 DO 65 J=1,3
      DO 62 I=6,24,6
   62 ID(I,J)=1
      DO 64 I=36,84,12
   64 ID(I,J)=1
      ID(61,J)=1
   65 ID(73,J)=1
      DO 66 I=1,3
      ID(85,I)=1
   66 ID(96,I)=1
      DO 68 I=86,95
      ID(I,1)=1
      ID(I,2)=1
   68 ID(I,3)=1
      DO 85 J=4,6
      DO 70 I=27,31
   70 ID(I,J)=0
      DO 72 I=1,19,6
   72 ID(I,J)=0
      DO 74 I=14,17
   74 ID(I,J)=0
      DO 76 I=40,76,12
   76 ID(I,J)=0
      DO 78 I=42,78,12
   78 ID(I,J)=0
      DO 80 I=11,47,12
   80 ID(I,J)=0
      ID(5,J)=0
   85 CONTINUE
C
C     FIX PILE TIP FOR POINT BEARING PILES
C
      IF(IPILE .EQ. 1)RETURN
      DO 87 I=1,6
      DO 86 J=76,78,2
   86 ID(J,I)=1
   87 CONTINUE
      RETURN
C
C     MATERIAL PROPERTIES
C
```

```
C      K=4    FRONT PILE
C      K=5    REAR PILE
C      K=6    ABUTMENT STEM
C      K=7    TIE ROD
C      K=8    DRILLED-IN CONCRETE ANCHOR
C      K=9    ABUTMENT BASE
C
   90 GO TO(95,95,95,100,100,110,105,107,110),K
C
C      TIE ROD PROPERTIES
C
  105 IF(IFILE .EQ. 1)GO TO 300
      CALL EDIT1(C11)
      GO TO 301
  300 READ(5,*)C11
      IF(IFILE .EQ. 1)WRITE(8,*)C11
  301 CONTINUE
      C2=DDA/12.
      C13=SSH
      C14=0.
      IC1=1
      WRITE(7,602)C11
      C11=C11*144.
      GO TO 106
C
C      DRILLED-IN CONCRETE ANCHOR
C
  107 C2=DDM
      C11=5.76E+08
      C13=SSH
      C14=0.
      IC1=1
      GO TO 106
C
C      FRONT AND REAR PILES
C
C      GENERAL SHAPED PILE INCLUDING H SECTION
C
  100 IF(K .EQ. 4)WRITE(7,596)
      IF(K .EQ. 5)WRITE(7,597)
      IF(IFILE .EQ. 1)GO TO 302
      CALL EDIT6(ITP)
      GO TO 303
  302 READ(5,501,ERR=111)ITP
      GO TO 112
  111 WRITE(6,595)
      GO TO 302
  112 CONTINUE
      IF(IFILE .EQ. 1)WRITE(8,501)ITP
  303 IF(ITP .EQ. 1)GO TO 108
      WRITE(6,580)
      IF(IFILE .EQ. 1)GO TO 304
      CALL EDIT6(IC1)
```

```
      GO TO 305
  304 READ(5,501,ERR=113)IC1
      GO TO 114
  113 WRITE(6,595)
      GO TO 304
  114 CONTINUE
      IF(IFILE .EQ. 1)WRITE(8,501)IC1
  305 WRITE(6,581)
      IF(IFILE .EQ. 1)GO TO 306
      CALL EDIT1(XA)
      GO TO 307
  306 READ(5,*)XA
      IF(IFILE .EQ. 1)WRITE(8,*)XA
  307 WRITE(6,582)
      IF(IFILE .EQ. 1)GO TO 308
      CALL EDIT1(YI)
      GO TO 309
  308 READ(5,*)YI
      IF(IFILE .EQ. 1)WRITE(8,*)YI
  309 WRITE(6,583)
      IF(IFILE .EQ. 1)GO TO 310
      CALL EDIT1(ZI)
      GO TO 311
  310 READ(5,*)ZI
      IF(IFILE .EQ. 1)WRITE(8,*)ZI
  311 WRITE(6,584)
      IF(IFILE .EQ. 1)GO TO 312
      CALL EDIT1(C11)
      GO TO 313
  312 READ(5,*)C11
      IF(IFILE .EQ. 1)WRITE(8,*)C11
  313 WRITE(6,585)
      IF(IFILE .EQ. 1)GO TO 314
      CALL EDIT1(C13)
      GO TO 315
  314 READ(5,*)C13
      IF(IFILE .EQ. 1)WRITE(8,*)C13
  315 CONTINUE
      WRITE(7,599)IC1,XA,YI,ZI,C11,C13
      C11=C11*144.
      XA=XA/144.*IC1
      YI=YI/20736.*IC1
      ZI=ZI/20736.*IC1
      GO TO 109
C
C     CIRCULAR PILE
C
  108 WRITE(6,590)
      IF(IFILE .EQ. 1)GO TO 316
      CALL EDIT6(IC1)
      GO TO 317
  316 READ(5,501,ERR=115)IC1
      GO TO 116
```

```
  115 WRITE(6,595)
      GO TO 316
  116 CONTINUE
      IF(IFILE .EQ. 1)WRITE(8,501)IC1
  317 WRITE(6,591)
      IF(IFILE .EQ. 1)GO TO 318
      CALL EDIT1(C2)
      GO TO 319
  318 READ(5,*)C2
      IF(IFILE .EQ. 1)WRITE(8,*)C2
  319 WRITE(6,592)
      IF(IFILE .EQ. 1)GO TO 320
      CALL EDIT1(C14)
      GO TO 321
  320 READ(5,*)C14
      IF(IFILE .EQ. 1)WRITE(8,*)C14
  321 WRITE(6,593)
      IF(IFILE .EQ. 1)GO TO 322
      CALL EDIT1(C11)
      GO TO 323
  322 READ(5,*)C11
      IF(IFILE .EQ. 1)WRITE(8,*)C11
  323 WRITE(6,594)
      IF(IFILE .EQ. 1)GO TO 324
      CALL EDIT1(C13)
      GO TO 325
  324 READ(5,*)C13
      IF(IFILE .EQ. 1)WRITE(8,*)C13
  325 CONTINUE
      WRITE(7,598)IC1,C2,C14,C11,C13
      C11=C11*144.
  106 CONTINUE
      XA=(C2**2-C14**2)*3.141592/4.*IC1
      YI=(C2**4-C14**4)*3.141592/64.*IC1
      ZI=YI
  109 AE(K)=XA*C11/C13
      GJX(K)=(YI+ZI)*C11/2.5/C13
      EIY(K)=YI*C11/C13
      EIZ(K)=ZI*C11/C13
      E(K)=C11
      POIS(K)=0.25
      FFX(K)=0.
      FFY(K)=0.
C
C     TIE ROD HAS NO BENDING AND TORSION RESISTENCE
C
      IF(K .NE. 7)RETURN
      GJX(K)=0.00000001*GJX(K)
      EIY(K)=0.00000001*EIY(K)
      EIZ(K)=0.00000001*EIZ(K)
   95 RETURN
C
C     ABUTMENT STEM AND BASE
```

```
C
  110 IF(IFILE .EQ. 1)GO TO 326
      CALL EDIT1(A2)
      GO TO 327
  326 READ(5,*)A2
      IF(IFILE .EQ. 1)WRITE(8,*)A2
C
C     MODULUS OF CONCRETE
C
  327 E(K)=5.76E+08
      POIS(K)=0.25
      FFX(K)=0.
      FFY(K)=0.
      AE(K)=A2*E(K)
      YI=A2**3/12.
      ZI=A2/12.
      GJX(K)=(YI+ZI)*E(K)/2.5
      EIY(K)=YI*E(K)
      EIZ(K)=ZI*E(K)
      IF(K .EQ. 6)WRITE(7,600)A2
      IF(K .EQ. 9)WRITE(7,601)A2
      RETURN
C
C     ELEMENT GENERATION
C
  120 NUMEL=100
      DO 126 I=1,NUMEL
      DO 125 J=1,6
  125 LM(I,J)=0
      ARE(I)=0.
      TTX(I)=0.
  126 TTY(I)=0.
C
C     ELEMENTS 1 THRU 55
C
      DO 130 J=1,5
      DO 130 I=1,11
      K=(J-1)*11+I
      L=23*(J-1)-K
      LM(K,1)=84-L
      LM(K,2)=85-L
      LM(K,3)=73-L
      LM(K,4)=72-L
      LM(K,6)=3
  130 IF(J .GT. 2)LM(K,6)=2
C
C     ELEMENTS 56 THRU 60
C
      DO 180 I=56,60
      LM(I,1)=I-25
      LM(I,2)=I-24
      LM(I,3)=I-36
      LM(I,4)=I-37
```

```
  180 LM(I,6)=2
      LM(56,6)=1
C
C     ELEMENTS 61 THRU 75
C
      DO 190 J=1,3
      DO 190 I=1,5
      K=(J-1)*5+60+I
      L=11*(J-1)
      LM(K,1)=K-42-L
      LM(K,2)=K-41-L
      LM(K,3)=K-47-L
      LM(K,4)=K-48-L
      LM(K,6)=1
  190 IF(J .EQ. 3)TTY(K)=-G1
      LM(63,6)=2
      LM(64,6)=2
      LM(65,6)=2
      LM(69,6)=2
      LM(70,6)=2
      LM(75,6)=2
C
C     ELEMENTS 76 THRU 79
C
      DO 240 I=1,3
      LM(75+I,1)=1+6*(I-1)
      LM(75+I,2)=7+6*(I-1)
  240 LM(75+I,6)=6
      LM(79,1)=19
      LM(79,2)=31
      LM(79,6)=6
C
C     ELEMENTS 80 THRU 83
C
      DO 250 I=80,83
      ARE(I)=ANGLE
      LM(I,1)=I-67
      LM(I,2)=I-66
  250 LM(I,6)=7
C
C     ELEMENTS 84 THRU 86
C
      DO 260 I=1,3
      LM(83+I,1)=5+6*(I-1)
      LM(83+I,2)=11+6*(I-1)
  260 LM(83+I,6)=8
C
C     ELEMENTS 87 THRU 88
C
      DO 270 I=1,2
      LM(86+I,1)=23+12*(I-1)
      LM(86+I,2)=35+12*(I-1)
  270 LM(86+I,6)=8
```

```
C
C     ELEMENTS 89 THRU 92
C
      DO 280 I=89,92
      LM(I,1)=I-62
      LM(I,2)=I-61
  280 LM(I,6)=9
C
C     ELEMENTS 93 THRU 99
C
      DO 220 I=1,4
      LM((91+I*2),1)=28+12*(I-1)
      LM((91+I*2),2)=40+12*(I-1)
  220 LM((91+I*2),6)=4
C
C     ELEMENTS 94 THRU 100
C
      DO 230 I=1,4
      LM((92+I*2),1)=30+12*(I-1)
      LM((92+I*2),2)=42+12*(I-1)
  230 LM((92+I*2),6)=5
      RETURN
      END
C
C
      SUBROUTINE ROTATE
C
C     THIS SUBROUTINE FORMS BA AND B'A' MATRICES NECESSARY FOR
C     ROTATION OF REINFORCED EARTH ELEMENTS
C
      COMMON /BLK2/XQ(4),YQ(4),B(6,12),XN(4),ALP,EST,PHP,PHR,BA(6,6),
     +             BAP(6,6)
      DIMENSION A(6,6),B1(6,6),AP(6,6),BP(6,6)
C
C     INITIALIZATION
C
      DO 10 I=1,6
      DO 10 J=1,6
      A(I,J)=0.
      B1(I,J)=0.
      AP(I,J)=0.
      BP(I,J)=0.
      BA(I,J)=0.
   10 BAP(I,J)=0.
C
C     A MATRIX
C
      AC=COS(PHP)
      AS=SIN(PHP)
      AC2=AC**2
      AS2=AS**2
      A(1,1)=AC2
      A(1,3)=AS2
```

```
      A(1,5)=2.*AC*AS
      A(2,2)=1.
      A(3,1)=AS2
      A(3,3)=AC2
      A(3,5)=-A(1,5)
      A(4,4)=AC
      A(4,6)=AS
      A(5,1)=-AC*AS
      A(5,3)=AC*AS
      A(5,5)=AC2-AS2
      A(6,4)=-AS
      A(6,6)=AC
C
C     B MATRIX
C
      BC=COS(PHR)
      BS=SIN(PHR)
      BC2=BC**2
      BS2=BS**2
      B1(1,1)=BC2
      B1(1,2)=BS2
      B1(1,4)=-2.*BC*BS
      B1(2,1)=BS2
      B1(2,2)=BC2
      B1(2,4)=2.*BC*BS
      B1(3,3)=1.
      B1(4,1)=BC*BS
      B1(4,2)=-B1(4,1)
      B1(4,4)=BC2-BS2
      B1(5,5)=BC
      B1(5,6)=-BS
      B1(6,5)=BS
      B1(6,6)=BC
C
C     A' AND B' MATRICES
C
      DO 20 I=1,6
      DO 20 J=1,6
      AP(I,J)=A(I,J)
   20 BP(I,J)=B1(I,J)
      AP(1,5)=0.5*AP(1,5)
      AP(3,5)=0.5*AP(3,5)
      AP(5,1)=2.*AP(5,1)
      AP(5,3)=2.*AP(5,3)
      BP(1,4)=0.5*BP(1,4)
      BP(2,4)=0.5*BP(2,4)
      BP(4,1)=2.*BP(4,1)
      BP(4,2)=2.*BP(4,2)
C
C     BA AND B'A' MATRICES
C
      DO 30 I=1,6
      DO 30 J=1,6
```

```
      DO 30 K=1,6
      BA(I,J)=BA(I,J)+B1(I,K)*A(K,J)
   30 BAP(I,J)=BAP(I,J)+BP(I,K)*AP(K,J)
      RETURN
      END
C
C
      SUBROUTINE DUNPRT(KCODE,K)
C
C
C     THIS SUBROUTINE ESTABLISHES MODIFIED DUNCAN'S SOIL PROPERTIES
C     FOR NONLINEAR FINITE ELEMENT ANALYSIS
C
C     KCODE = 1 CRUSHED LIMESTONE BACKFILL
C             2 CRUSHED QUARTZITE BACKFILL
C             3 REDDISH BROWN SILTY CLAY,                 MEADE   Z= 8-10 FT
C             4 REDDISH BROWN SILTY CLAY,                 MEADE   Z=13-15
C             5 REDDISH BROWN SILTY CLAY,                 MEADE   Z=18-20
C             6 GRAY SILTY CLAY WITH PEBBLES,             PENNING Z=8.5-10.5
C             7 GRAY SHALE,                               PENNING Z=14-16
C             8 LIGHT BROWN SILTY CLAY,                   PENNING Z= 8- 9
C             9 LIGHT BROWN SILTY CLAY,                   PENNING Z=13-15
C            10 BROWN SILTY CLAY WITH PEBBLES,            PENNING Z=18-20
C            11 BROWN SILTY CLAY WITH SAND & PEBBLES,     MINNEHA Z=13-14
C            12 BROWN SILTY CLAY WITH SAND,               MINNEHA Z=16-17
C            13 GRAY BROWN SHALE,                         JONES   Z=32-56
C            14 GRAY SILTY CLAY,                          LAWRENC Z=7.5- 9
C            15 GRAY SILTY CLAY,                          LAWRENC Z=14-15
C            16 GRAY SILTY CLAY,                          LAWRENC Z=19-20
C            17 BROWN SILTY CLAY,                         BRULE   Z= 3-16
C            18 BROWN SILTY CLAY,                         BRULE   Z=32-42
C            19 BROWN SILTY CLAY,                         BRULE   Z=51-55
C            20 GRAY SHALE,                               BUFFALO Z=38-41.5
C
C     GENERAL COMPACTED SOIL PROPERTIES
C
C            21 GW,GP,SW,SP SOIL WITH STANDARD RC=105 OF AASHTO
C            22                                    100
C            23                                     95
C            24                                     90
C            25 SM SOIL WITH STANDARD RC=100 OF AASHTO
C            26                                95
C            27                                90
C            28                                85
C            29 SM,SC SOIL WITH STANDARD RC=100 OF AASHTO
C            30                                   95
C            31                                   90
C            32                                   85
C            33 CL SOIL WITH STANDARD RC=100 OF AASHTO
C            34                                95
C            35                                90
C            36                                85
C
```

```
      COMMON /BLK1/AE(10),EIY(10),E(10),POIS(10),EX(102),GX(102),INR
      COMMON /BLK17/EIZ(10),GJX(10),IPILE,XPILE,INCR,XXS(102,5),
     +              YYS(102,5),IFILE
      COMMON /BLK12/ZZS(102,5),XYT(102,5),XZT(102,5),YZT(102,5)
      COMMON /BLK7/DUN(10,9),PA
      COMMON/BLK10/ID(100,6),PN(6),FFX(10),FFY(10),TTX(102),TTY(102)
      DO 5 I=1,9
    5 DUN(K,I)=0.
      IF(KCODE .GT. 9)GO TO 100
      GO TO(10,20,30,40,50,60,70,80,90),KCODE
C
C
C     KCODE = 1
C
C
   10 DUN(K,1)=700.
      DUN(K,2)=1400.
      DUN(K,3)=0.85
      DUN(K,4)=0.74
      DUN(K,5)=0.
      DUN(K,6)=47.
      DUN(K,7)=12.
      DUN(K,8)=208.
      DUN(K,9)=0.2
      FFX(K)=0.
      FFY(K)=-103.5
      POIS(K)=0.0
      RETURN
C
C
C     KCODE = 2
C
C
   20 DUN(K,1)=525.
      DUN(K,2)=1050.
      DUN(K,3)=0.95
      DUN(K,4)=0.76
      DUN(K,5)=0.
      DUN(K,6)=47.
      DUN(K,7)=12.
      DUN(K,8)=150.
      DUN(K,9)=0.2
      FFX(K)=0.
      FFY(K)=-103.5
      POIS(K)=0.0
      RETURN
C
C
C     KCODE = 3
C
C
   30 DUN(K,1)=140.
      DUN(K,2)=280.
```

```
      DUN(K,3)=0.34
      DUN(K,4)=0.59
      DUN(K,5)=2400.
      DUN(K,6)=17.
      DUN(K,7)=0.
      DUN(K,8)=130.
      DUN(K,9)=0.2
      FFX(K)=0.
      FFY(K)=-137.5
      POIS(K)=0.0
      RETURN
C
C
C     KCODE = 4
C
C
   40 DUN(K,1)=150.
      DUN(K,2)=300.
      DUN(K,3)=0.23
      DUN(K,4)=0.43
      DUN(K,5)=1700.
      DUN(K,6)=30.
      DUN(K,7)=0.
      DUN(K,8)=140.
      DUN(K,9)=0.2
      FFX(K)=0.
      FFY(K)=-137.5
      POIS(K)=0.0
      RETURN
C
C
C     KCODE = 5
C
C
   50 DUN(K,1)=215.
      DUN(K,2)=430.
      DUN(K,3)=0.41
      DUN(K,4)=0.6
      DUN(K,5)=1100.
      DUN(K,6)=35.
      DUN(K,7)=0.
      DUN(K,8)=205.
      DUN(K,9)=0.2
      FFX(K)=0.
      FFY(K)=-136.0
      POIS(K)=0.0
      RETURN
C
C
C     KCODE = 6
C
C
   60 DUN(K,1)=290.
```

```
      DUN(K,2)=580.
      DUN(K,3)=0.32
      DUN(K,4)=0.65
      DUN(K,5)=0.
      DUN(K,6)=50.
      DUN(K,7)=0.
      DUN(K,8)=280.
      DUN(K,9)=0.2
      FFX(K)=0.
      FFY(K)=-125.3
      POIS(K)=0.0
      RETURN
C
C
C     KCODE = 7
C
C
   70 DUN(K,1)=140.
      DUN(K,2)=280.
      DUN(K,3)=0.43
      DUN(K,4)=0.7
      DUN(K,5)=1550.
      DUN(K,6)=15.
      DUN(K,7)=0.
      DUN(K,8)=130.
      DUN(K,9)=0.2
      FFX(K)=0.
      FFY(K)=-124.8
      POIS(K)=0.0
      RETURN
C
C
C     KCODE = 8
C
C
   80 DUN(K,1)=90.
      DUN(K,2)=180.
      DUN(K,3)=0.26
      DUN(K,4)=0.63
      DUN(K,5)=1850.
      DUN(K,6)=25.
      DUN(K,7)=0.
      DUN(K,8)=80.
      DUN(K,9)=0.2
      FFX(K)=0.
      FFY(K)=-125.
      POIS(K)=0.0
      RETURN
C
C
C     KCODE = 9
C
C
```

```
   90 DUN(K,1)=235.
      DUN(K,2)=470.
      DUN(K,3)=0.1
      DUN(K,4)=0.79
      DUN(K,5)=1650.
      DUN(K,6)=30.
      DUN(K,7)=0.
      DUN(K,8)=225.
      DUN(K,9)=0.2
      FFX(K)=0.
      FFY(K)=-126.6
      POIS(K)=0.0
      RETURN
C
C
  100 IF(KCODE .GT. 18)GO TO 200
      KK=MOD(KCODE,9)
      IF(KK .EQ. 0)KK=9
      GO TO(110,120,130,140,150,160,170,180,190),KK
C
C
C     KCODE = 10
C
C
  110 DUN(K,1)=220.
      DUN(K,2)=440.
      DUN(K,3)=0.36
      DUN(K,4)=0.82
      DUN(K,5)=0.
      DUN(K,6)=36.
      DUN(K,7)=0.
      DUN(K,8)=210.
      DUN(K,9)=0.2
      FFX(K)=0.
      FFY(K)=-125.5
      POIS(K)=0.0
      RETURN
C
C
C     KCODE = 11
C
C
  120 DUN(K,1)=125.
      DUN(K,2)=250.
      DUN(K,3)=0.26
      DUN(K,4)=0.76
      DUN(K,5)=0.
      DUN(K,6)=32.
      DUN(K,7)=0.
      DUN(K,8)=115.
      DUN(K,9)=0.2
      FFX(K)=0.
      FFY(K)=-134.3
```

```
      POIS(K)=0.0
      RETURN
C
C
C     KCODE = 12
C
C
  130 DUN(K,1)=140.
      DUN(K,2)=280.
      DUN(K,3)=0.53
      DUN(K,4)=0.67
      DUN(K,5)=0.
      DUN(K,6)=42.
      DUN(K,7)=0.
      DUN(K,8)=130.
      DUN(K,9)=0.2
      FFX(K)=0.
      FFY(K)=-127.8
      POIS(K)=0.0
      RETURN
C
C
C     KCODE = 13
C
C
  140 DUN(K,1)=470.
      DUN(K,2)=940.
      DUN(K,3)=0.21
      DUN(K,4)=0.37
      DUN(K,5)=0.
      DUN(K,6)=40.
      DUN(K,7)=0.
      DUN(K,8)=460.
      DUN(K,9)=0.2
      FFX(K)=0.
      FFY(K)=-121.5
      POIS(K)=0.0
      RETURN
C
C
C     KCODE = 14
C
C
  150 DUN(K,1)=170.
      DUN(K,2)=340.
      DUN(K,3)=0.38
      DUN(K,4)=0.53
      DUN(K,5)=1200.
      DUN(K,6)=24.
      DUN(K,7)=0.
      DUN(K,8)=160.
      DUN(K,9)=0.2
      FFX(K)=0.
```

```
      FFY(K)=-124.
      POIS(K)=0.0
      RETURN
C
C
C     KCODE = 15
C
C
  160 DUN(K,1)=245.
      DUN(K,2)=490.
      DUN(K,3)=0.25
      DUN(K,4)=0.65
      DUN(K,5)=1500.
      DUN(K,6)=35.
      DUN(K,7)=0.
      DUN(K,8)=235.
      DUN(K,9)=0.2
      FFX(K)=0.
      FFY(K)=-125.6
      POIS(K)=0.0
      RETURN
C
C
C     KCODE = 16
C
C
  170 DUN(K,1)=105.
      DUN(K,2)=210.
      DUN(K,3)=0.36
      DUN(K,4)=0.57
      DUN(K,5)=0.
      DUN(K,6)=40.
      DUN(K,7)=0.
      DUN(K,8)=95.
      DUN(K,9)=0.2
      FFX(K)=0.
      FFY(K)=-125.
      POIS(K)=0.0
      RETURN
C
C
C     KCODE = 17
C
C
  180 DUN(K,1)=195.
      DUN(K,2)=390.
      DUN(K,3)=0.2
      DUN(K,4)=0.8
      DUN(K,5)=1300.
      DUN(K,6)=40.
      DUN(K,7)=0.
      DUN(K,8)=185.
      DUN(K,9)=0.2
```

```
      FFX(K)=0.
      FFY(K)=-128.7
      POIS(K)=0.0
      RETURN
C
C
C     KCODE = 18
C
C
  190 DUN(K,1)=255.
      DUN(K,2)=510.
      DUN(K,3)=0.34
      DUN(K,4)=0.7
      DUN(K,5)=2950.
      DUN(K,6)=10.
      DUN(K,7)=0.
      DUN(K,8)=245.
      DUN(K,9)=0.2
      FFX(K)=0.
      FFY(K)=-122.
      POIS(K)=0.0
      RETURN
  200 IF(KCODE .GT. 27)GO TO 300
      KK=MOD(KCODE,9)
      IF(KK .EQ. 0)KK=9
      GO TO(210,220,230,240,250,260,270,280,290),KK
C
C
C     KCODE = 19
C
C
  210 DUN(K,1)=305.
      DUN(K,2)=610.
      DUN(K,3)=0.42
      DUN(K,4)=.65
      DUN(K,5)=2100.
      DUN(K,6)=32.
      DUN(K,7)=0.
      DUN(K,8)=295.
      DUN(K,9)=0.2
      FFX(K)=0.
      FFY(K)=-122.
      POIS(K)=0.0
      RETURN
C
C
C     KCODE = 20
C
C
  220 DUN(K,1)=115.
      DUN(K,2)=230.
      DUN(K,3)=0.92
      DUN(K,4)=0.2
```

```
      DUN(K,5)=0.
      DUN(K,6)=50.
      DUN(K,7)=0.
      DUN(K,8)=105.
      DUN(K,9)=0.2
      FFX(K)=0.
      FFY(K)=-116.
      POIS(K)=0.0
      RETURN
C
C
C     KCODE = 21
C
C
  230 DUN(K,1)=600.
      DUN(K,2)=1200.
      DUN(K,3)=0.4
      DUN(K,4)=0.7
      DUN(K,5)=0.
      DUN(K,6)=42.
      DUN(K,7)=9.
      DUN(K,8)=175.
      DUN(K,9)=0.2
      FFY(K)=-150.
      POIS(K)=0.
      RETURN
C
C
C     KCODE = 22
C
C
  240 DUN(K,1)=450.
      DUN(K,2)=900.
      DUN(K,3)=0.4
      DUN(K,4)=0.7
      DUN(K,5)=0.
      DUN(K,6)=39.
      DUN(K,7)=7.
      DUN(K,8)=125.
      DUN(K,9)=0.2
      FFX(K)=0.
      FFY(K)=-145.
      POIS(K)=0.
      RETURN
C
C
C     KCODE = 23
C
C
  250 DUN(K,1)=300.
      DUN(K,2)=600.
      DUN(K,3)=0.4
      DUN(K,4)=0.7
```

```
      DUN(K,5)=0.
      DUN(K,6)=36.
      DUN(K,7)=5.
      DUN(K,8)=75.
      DUN(K,9)=0.2
      FFX(K)=0.
      FFY(K)=-140.
      POIS(K)=0.
      RETURN
C
C
C     KCODE = 24
C
C
  260 DUN(K,1)=200.
      DUN(K,2)=400.
      DUN(K,3)=0.4
      DUN(K,4)=0.7
      DUN(K,5)=0.
      DUN(K,6)=33.
      DUN(K,7)=3.
      DUN(K,8)=50.
      DUN(K,9)=0.2
      FFX(K)=0.
      FFY(K)=-135.
      POIS(K)=0.
      RETURN
C
C
C     KCODE = 25
C
C
  270 DUN(K,1)=600.
      DUN(K,2)=1200.
      DUN(K,3)=0.25
      DUN(K,4)=0.7
      DUN(K,5)=0.
      DUN(K,6)=36.
      DUN(K,7)=8.
      DUN(K,8)=450.
      DUN(K,9)=0.
      FFX(K)=0.
      FFY(K)=-135.
      POIS(K)=0.
      RETURN
C
C
C     KCODE = 26
C
C
  280 DUN(K,1)=450.
      DUN(K,2)=900.
      DUN(K,3)=0.25
```

```
      DUN(K,4)=0.7
      DUN(K,5)=0.
      DUN(K,6)=34.
      DUN(K,7)=6.
      DUN(K,8)=350.
      DUN(K,9)=0.
      FFX(K)=0.
      FFY(K)=-130.
      POIS(K)=0.
      RETURN
C
C
C     KCODE = 27
C
C
  290 DUN(K,1)=300.
      DUN(K,2)=600.
      DUN(K,3)=0.25
      DUN(K,4)=0.7
      DUN(K,5)=0.
      DUN(K,6)=32.
      DUN(K,7)=4.
      DUN(K,8)=250.
      DUN(K,9)=0.
      FFX(K)=0.
      FFY(K)=-125.
      POIS(K)=0.
      RETURN
  300 IF(KCODE .GT. 36)GO TO 400
      KK=MOD(KCODE,9)
      IF(KK .EQ. 0)KK=9
      GO TO(310,320,330,340,350,360,370,380,390),KK
C
C
C     KCODE = 28
C
C
  310 DUN(K,1)=150.
      DUN(K,2)=300.
      DUN(K,3)=0.25
      DUN(K,4)=.7
      DUN(K,5)=0.
      DUN(K,6)=30.
      DUN(K,7)=2.
      DUN(K,8)=150.
      DUN(K,9)=0.
      FFX(K)=0.
      FFY(K)=-120.
      POIS(K)=0.
      RETURN
C
C
C     KCODE = 29
```

```
C
C
  320 DUN(K,1)=400.
      DUN(K,2)=800.
      DUN(K,3)=0.6
      DUN(K,4)=0.7
      DUN(K,5)=500.
      DUN(K,6)=33.
      DUN(K,7)=0.
      DUN(K,8)=200.
      DUN(K,9)=0.5
      FFX(K)=0.
      FFY(K)=-135.
      POIS(K)=0.
      RETURN
C
C
C     KCODE = 30
C
C
  330 DUN(K,1)=200.
      DUN(K,2)=400.
      DUN(K,3)=0.6
      DUN(K,4)=0.7
      DUN(K,5)=400.
      DUN(K,6)=33.
      DUN(K,7)=0.
      DUN(K,8)=100.
      DUN(K,9)=0.5
      FFX(K)=0.
      FFY(K)=-130.
      POIS(K)=0.
      RETURN
C
C
C     KCODE = 31
C
C
  340 DUN(K,1)=150.
      DUN(K,2)=300.
      DUN(K,3)=0.6
      DUN(K,4)=0.7
      DUN(K,5)=300.
      DUN(K,6)=33.
      DUN(K,7)=0.
      DUN(K,8)=75.
      DUN(K,9)=0.5
      FFX(K)=0.
      FFY(K)=-125.
      POIS(K)=0.
      RETURN
C
C
```

```
C      KCODE = 32
C
C
  350 DUN(K,1)=100.
      DUN(K,2)=200.
      DUN(K,3)=0.6
      DUN(K,4)=0.7
      DUN(K,5)=200.
      DUN(K,6)=33.
      DUN(K,7)=0.
      DUN(K,8)=50.
      DUN(K,9)=0.5
      FFX(K)=0.
      FFY(K)=-120.
      POIS(K)=0.
      RETURN
C
C
C      KCODE = 33
C
C
  360 DUN(K,1)=150.
      DUN(K,2)=300.
      DUN(K,3)=0.45
      DUN(K,4)=0.7
      DUN(K,5)=400.
      DUN(K,6)=30.
      DUN(K,7)=0.
      DUN(K,8)=140.
      DUN(K,9)=0.2
      FFX(K)=0.
      FFY(K)=-135.
      POIS(K)=0.
      RETURN
C
C
C      KCODE = 34
C
C
  370 DUN(K,1)=120.
      DUN(K,2)=240.
      DUN(K,3)=0.45
      DUN(K,4)=0.7
      DUN(K,5)=300.
      DUN(K,6)=30.
      DUN(K,7)=0.
      DUN(K,8)=110.
      DUN(K,9)=0.2
      FFX(K)=0.
      FFY(K)=-130.
      POIS(K)=0.
      RETURN
C
```

```
C
C     KCODE = 35
C
C
  380 DUN(K,1)=90.
      DUN(K,2)=180.
      DUN(K,3)=0.45
      DUN(K,4)=0.7
      DUN(K,5)=200.
      DUN(K,6)=30.
      DUN(K,7)=0.
      DUN(K,8)=80.
      DUN(K,9)=0.2
      FFX(K)=0.
      FFY(K)=-125.
      POIS(K)=0.
      RETURN
C
C
C     KCODE = 36
C
C
  390 DUN(K,1)=60.
      DUN(K,2)=120.
      DUN(K,3)=0.45
      DUN(K,4)=0.7
      DUN(K,5)=100.
      DUN(K,6)=30.
      DUN(K,7)=0.
      DUN(K,8)=50.
      DUN(K,9)=0.2
      FFX(K)=0.
      FFY(K)=-120.
      POIS(K)=0.
  400 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE SOIL(KCODE,K)
C
C
C     THIS SUBROUTINE PRINTS THE SOIL CHARACTERISTIC BASED ON THE
C     REFERENCE NUMBER
C
C
      COMMON /BLK7/DUN(10,9),PA
      COMMON/BLK10/ID(100,6),PN(6),FFX(10),FFY(10),TTX(102),TTY(102)
C
C
   10 FORMAT(/1X,'THE SOIL YOU DESIGNATED IS')
   20 FORMAT(1X,'CRUSHED LIMESTONE BACKFILL')
   30 FORMAT(1X,'CRUSHED QUARTZITE BACKFILL')
```

```
   40 FORMAT(1X,'REDDISH BROWN SILTY CLAY MEADE CO Z=8-10 FT')
   50 FORMAT(1X,'REDDISH BROWN SILTY CLAY MEADE CO Z=13-15 FT')
   60 FORMAT(1X,'REDDISH BROWN SILTY CLAY MEADE CO Z=18-20 FT')
   70 FORMAT(1X,'GRAY SILTY CLAY WITH PEBBLES PENN CO Z=8.5-10.5 FT')
   80 FORMAT(1X,'GRAY SHALE PENN CO Z=14-16 FT')
   90 FORMAT(1X,'LIGHT BROWN SILTY CLAY PENN CO Z=8-9 FT')
  100 FORMAT(1X,'LIGHT BROWN SILTY CLAY PENN CO Z=13-15 FT')
  110 FORMAT(1X,'BROWN SILTY CLAY WITH PEBBLES PENN CO Z=18-20 FT')
  120 FORMAT(1X,'BROWN SILTY CLAY WITH PEBBLES MINN CO Z=13-14 FT')
  130 FORMAT(1X,'BROWN SILTY CLAY WITH SAND MINN CO Z=16-17 FT')
  140 FORMAT(1X,'GRAY BROWN SHALE JONES CO Z=32-56 FT')
  150 FORMAT(1X,'GRAY SILTY CLAY LAWRENCE CO Z=7.5-9 FT')
  160 FORMAT(1X,'GRAY SILTY CLAY LAWRENCE CO Z=14-15 FT')
  170 FORMAT(1X,'GRAY SILTY CLAY LAWRENCE CO Z=19-20 FT')
  180 FORMAT(1X,'BROWN SILTY CLAY BRULE CO Z=3-16 FT')
  190 FORMAT(1X,'BROWN SILTY CLAY BRULE CO Z=32-42 FT')
  200 FORMAT(1X,'BROWN SILTY CLAY BRULE CO Z=51-55 FT')
  210 FORMAT(1X,'GRAY SHALE BUFFALO CO Z=38-41.5 FT')
  220 FORMAT(1X,'GW GP SW SP SOIL WITH STD RC=105%')
  230 FORMAT(1X,'GW GP SW SP SOIL WITH STD RC=100%')
  240 FORMAT(1X,'GW GP SW SP SOIL WITH STD RC= 95%')
  250 FORMAT(1X,'GW GP SW SP SOIL WITH STD RC= 90%')
  260 FORMAT(1X,'SM SOIL WITH STD RC=100%')
  270 FORMAT(1X,'SM SOIL WITH STD RC= 95%')
  280 FORMAT(1X,'SM SOIL WITH STD RC= 90%')
  290 FORMAT(1X,'SM SOIL WITH STD RC= 85%')
  300 FORMAT(1X,'SM SC SOIL WITH STD RC=100%')
  310 FORMAT(1X,'SM SC SOIL WITH STD RC= 95%')
  320 FORMAT(1X,'SM SC SOIL WITH STD RC= 90%')
  330 FORMAT(1X,'SM SC SOIL WITH STD RC= 85%')
  340 FORMAT(1X,'CL SOIL WITH STD RC=100%')
  350 FORMAT(1X,'CL SOIL WITH STD RC= 95%')
  360 FORMAT(1X,'CL SOIL WITH STD RC= 90%')
  370 FORMAT(1X,'CL SOIL WITH STD RC= 85%')
  380 FORMAT( 1X,'COHESION=',F10.3,' PSF',1X, 'FRICTION ANGLE=',F10.3,
     +  ' DEG',1X,'UNIT WEIGHT=',F10.3,' PCF')
  390 FORMAT( 1X,'SAME AS PREVIOUS SOIL')
C
C
      IF(KCODE .EQ. 0)RETURN
      WRITE(6,10)
      IF(K .NE. -1)GO TO 400
      WRITE(6,390)
      RETURN
  400 CONTINUE
      IF(KCODE .GT. 9)GO TO 500
      GO TO (410,420,430,440,450,460,470,480,490),KCODE
  410 WRITE(6,20)
      GO TO 800
  420 WRITE(6,30)
      GO TO 800
  430 WRITE(6,40)
      GO TO 800
```

```
  440 WRITE(6,50)
      GO TO 800
  450 WRITE(6,60)
      GO TO 800
  460 WRITE(6,70)
      GO TO 800
  470 WRITE(6,80)
      GO TO 800
  480 WRITE(6,90)
      GO TO 800
  490 WRITE(6,100)
      GO TO 800
  500 IF(KCODE .GT. 18)GO TO 600
      KK=MOD(KCODE,9)
      IF(KK .EQ. 0)KK=9
      GO TO (510,520,530,540,550,560,570,580,590),KK
  510 WRITE(6,110)
      GO TO 800
  520 WRITE(6,120)
      GO TO 800
  530 WRITE(6,130)
      GO TO 800
  540 WRITE(6,140)
      GO TO 800
  550 WRITE(6,150)
      GO TO 800
  560 WRITE(6,160)
      GO TO 800
  570 WRITE(6,170)
      GO TO 800
  580 WRITE(6,180)
      GO TO 800
  590 WRITE(6,190)
      GO TO 800
  600 IF(KCODE .GT. 27)GO TO 700
      KK=MOD(KCODE,9)
      IF(KK .EQ. 0)KK=9
      GO TO (610,620,630,640,650,660,670,680,690),KK
  610 WRITE(6,200)
      GO TO 800
  620 WRITE(6,210)
      GO TO 800
  630 WRITE(6,220)
      GO TO 800
  640 WRITE(6,230)
      GO TO 800
  650 WRITE(6,240)
      GO TO 800
  660 WRITE(6,250)
      GO TO 800
  670 WRITE(6,260)
      GO TO 800
  680 WRITE(6,270)
```

```
      GO TO 800
  690 WRITE(6,280)
      GO TO 800
  700 IF(KCODE .GT. 36)GO TO 900
      KK=MOD(KCODE,9)
      IF(KK .EQ. 0)KK=9
      GO TO (710,720,730,740,750,760,770,780,790),KK
  710 WRITE(6,290)
      GO TO 800
  720 WRITE(6,300)
      GO TO 800
  730 WRITE(6,310)
      GO TO 800
  740 WRITE(6,320)
      GO TO 800
  750 WRITE(6,330)
      GO TO 800
  760 WRITE(6,340)
      GO TO 800
  770 WRITE(6,350)
      GO TO 800
  780 WRITE(6,360)
      GO TO 800
  790 WRITE(6,370)
  800 FY=-FFY(K)
      WRITE(6,380)DUN(K,5),DUN(K,6),FY
  900 RETURN
      END
C
C
      SUBROUTINE EDIT1(A)
C
C
C     THIS SUBROUTINE EDITS OR COPIES INPUT DATA FILE
C     FOR FREE FORMATTED REAL NUMBER
C
C
      COMMON/BLK13/CD
      CHARACTER*1 YES/'Y'/
      CHARACTER   CD*1
   10 FORMAT(/1X,'TO EDIT',5X,'CURRENT VALUE =',1X,F15.3,
     +/13X,'ENTER NEW VALUE  TO EDIT OR RETURN TO RETAIN')
   20 FORMAT(F20.9)
      READ(9,*)A
      IF(CD .NE. YES)RETURN
      WRITE(6,10)A
      READ(5,20)AX
      IF(AX .EQ. 0.)AX=A
      WRITE(10,*)AX
      A=AX
      RETURN
      END
C
```

```
C
      SUBROUTINE EDIT2(A)
C
C
C     THIS SUBROUTINE EDITS OR COPIES INPUT DATA FILE
C     FOR FIXED FORMATTED REAL NUMBER
C
C
      COMMON/BLK13/CD
      CHARACTER*1 YES/'Y'/
      CHARACTER   CD*1
   10 FORMAT(/1X,'TO EDIT',5X,'CURRENT VALUE =',1X,F15.3,
     +/13X,'ENTER NEW VALUE  TO EDIT OR RETURN TO RETAIN')
   20 FORMAT(F20.9)
      READ(9,20)A
      IF(CD .NE. YES)RETURN
      WRITE(6,10)A
      READ(5,20)AX
      IF(AX .EQ. 0.)AX=A
      WRITE(10,20)AX
      A=AX
      RETURN
      END
C
C
      SUBROUTINE EDIT3(I)
C
C
C     THIS SUBROUTINE EDITS OR COPIES INPUT DATA FILE
C     FOR FIXED FORMATTED INTEGER NUMBER (I2)
C
C
      COMMON/BLK13/CD
      CHARACTER*1 YES/'Y'/
      CHARACTER   CD*1
   10 FORMAT(/1X,'TO EDIT',5X,'CURRENT VALUE =',1X,I10,
     +/13X,'ENTER NEW VALUE  TO EDIT OR RETURN TO RETAIN')
   20 FORMAT(I2)
   30 FORMAT(/1X,'**ERROR**REENTER IN INTEGER(S)',\)
      READ(9,20)I
      IF(CD .NE. YES)RETURN
      WRITE(6,10)I
   25 READ(5,20,ERR=40)IX
      GO TO 50
   40 WRITE(6,30)
      GO TO 25
   50 CONTINUE
      IF(IX .EQ. 0 )IX=I
      WRITE(10,20)IX
      I=IX
      RETURN
      END
C
```

```
C
      SUBROUTINE EDIT6(I)
C
C
C     THIS SUBROUTINE EDITS OR COPIES INPUT DATA FILE
C     FOR FIXED FORMATTED INTEGER NUMBER (I1)
C
C
      COMMON/BLK13/CD
      CHARACTER*1 YES/'Y'/
      CHARACTER   CD*1
      CHARACTER*1 C
      CHARACTER*1 B/'Y'/
   10 FORMAT(/1X,'TO EDIT',5X,'CURRENT VALUE =',1X,I10,
     +/13X,'LIKE TO MAKE CHANGE (Y/N)?')
   15 FORMAT(1X,'ENTER NEW VALUE')
   16 FORMAT(A1)
   20 FORMAT(I1)
   30 FORMAT(/1X,'**ERROR**REENTER IN INTEGER(S)',\)
      READ(9,20)I
      IF(CD .NE. YES)RETURN
      WRITE(6,10)I
      READ(5,16)C
      IF(C .NE. B)GO TO 60
      WRITE(6,15)
   25 READ(5,20,ERR=40)IX
      GO TO 50
   40 WRITE(6,30)
      GO TO 25
   50 CONTINUE
      WRITE(10,20)IX
      I=IX
      RETURN
   60 WRITE(10,20)I
      RETURN
      END
C
C
      SUBROUTINE EDIT7(I)
C
C
C     THIS SUBROUTINE EDITS OR COPIES INPUT DATA FILE
C     FOR CHARACTER (A1)
C
C
      COMMON/BLK13/CD
      CHARACTER*1 YES/'Y'/
      CHARACTER   CD*1
      CHARACTER I*1,IX*1
      CHARACTER*1 K/' '/
   10 FORMAT(/1X,'TO EDIT',5X,'CURRENT VALUE =',1X,A1,
     +/13X,'ENTER NEW VALUE  TO EDIT OR RETURN TO RETAIN')
   20 FORMAT(A1)
```

```
   40 FORMAT(/1X,'**ERROR**REENTER IN CHARACTER',\)
      READ(9,20)I
      IF(CD .NE. YES)RETURN
      WRITE(6,10)I
   45 READ(5,20,ERR=50)IX
      GO TO 60
   50 WRITE(6,40)
      GO TO 45
   60 CONTINUE
      IF(IX .EQ. K)IX=I
      WRITE(10,20)IX
      I=IX
      RETURN
      END
C
C
      SUBROUTINE EDIT8(I,J)
C
C
C     THIS SUBROUTINE EDITS OR COPIES INPUT DATA FILE
C     FOR FREE FORMATTED TWO INTEGER NUMBERS
C
C
      COMMON/BLK13/CD
      CHARACTER*1 YES/'Y'/
      CHARACTER   CD*1
      CHARACTER*1 C
      CHARACTER*1 B/'Y'/
   10 FORMAT(/1X,'TO EDIT',5X,'CURRENT VALUES=',1X,2I10,
     +/13X,'LIKE TO MAKE CHANGES(Y/N)?')
   15 FORMAT(/1X,'ENTER BOTH NUMBERS')
   20 FORMAT(A1)
   40 FORMAT(/1X,'**ERROR**REENTER IN INTEGER(S)',\)
      READ(9,*)I,J
      IF(CD .NE. YES)RETURN
      WRITE(6,10)I,J
      READ(5,20)C
      IF(C .NE. B)GO TO 30
      WRITE(6,15)
   45 READ(5,*,ERR=50)IX,JX
      GO TO 60
   50 WRITE(6,40)
      GO TO 45
   60 CONTINUE
      WRITE(10,*)IX,JX
      I=IX
      J=JX
      RETURN
   30 WRITE(10,*)I,J
      RETURN
      END
C
C
```

```
      SUBROUTINE EDIT9(K)
C
C
C     THIS SUBROUTINE EDITS OR COPIES INPUT DATA FILE
C     FOR FREE FORMATTED REAL NUMBERS(12)
C
C
      COMMON /BLK1/AE(10),EIY(10),E(10),POIS(10),EX(102),GX(102),INR
      COMMON /BLK7/DUN(10,9),PA
      COMMON /BLK10/ID(100,6),PN(6),FFX(10),FFY(10),TTX(102),TTY(102)
      COMMON/BLK13/CD
      DIMENSION DUNX(10,9),POISX(10),FFXX(10),FFYX(10)
      CHARACTER*1 YES/'Y'/
      CHARACTER   CD*1
      CHARACTER*1 C
      CHARACTER*1 B/'Y'/
   10 FORMAT(/1X,'TO EDIT',5X,'CURRENT VALUES=',1X,4(F7.2,1X),
     +/13X,8(F7.2,1X),
     +/13X,'LIKE TO MAKE CHANGES (Y/N)?')
   25 FORMAT(/1X,'ENTER ALL 12 VALUES')
   20 FORMAT(A1)
      READ(9,*)(DUN(K,I),I=1,9),POIS(K),FFX(K),FFY(K)
      IF(CD .NE. YES)RETURN
      WRITE(6,10)(DUN(K,I),I=1,9),POIS(K),FFX(K),FFY(K)
      READ(5,20)C
      IF(C .NE. B)GO TO 50
      WRITE(6,25)
      READ(5,*)(DUNX(K,I),I=1,9),POISX(K),FFXX(K),FFYX(K)
   30 WRITE(10,*)(DUNX(K,I),I=1,9),POISX(K),FFXX(K),FFYX(K)
      DO 40 I=1,9
   40 DUN(K,I)=DUNX(K,I)
      POIS(K)=POISX(K)
      FFX(K)=FFXX(K)
      FFY(K)=FFYX(K)
      RETURN
   50 WRITE(10,*)(DUN(K,I),I=1,9),POIS(K),FFX(K),FFY(K)
      RETURN
      END
```

LISTING OF PROGRAM PLOT and PLOT2

A complete listing of the finite element grid plotting program, PLOT and PLOT2, are given in this section. The program is written in FORTRAN 5 for IBM compatible personal computers. Compilation of the program was done with the Microsoft Fortran Compiler 5.0. The program must reside in the same subdirectory with the programs BULK or BART, since the necessary data file is generated by those programs. The sizes of the source program and executable program are approximately:

.FOR files : 6,700 bytes

.EXE file : 59,000 bytes

The commands necessary to compile and link the .FOR file using the MIcrosoft Fortran Compiler are

```
FL /FPi /c PLOT.FOR
LINK
PLOT
(return)
(return)
(path of GRAPHICS.LIB)
(return)
```

This command compiles the .FOR files with or without math coprocessor.

```
C
C     THIS PROGRAM DRAWS THE FINITE ELEMENT GRID ASSOCIATED WITH THE
C     COMPUTER PROGRAM BART OR BULK.  THE NECESSARY INPUT DATA ARE
C     WRITTEN BY THE PROGRAM BART OR BULK ONTO FILE 'XYDATA' AND READ
C     BY THIS PROGRAM.  IT IS PREFERRED TO USE A HARD DISC FOR ALL
C     FILES.  THIS PROGRAM REQUIRES THE FOLLOWING.
C     MONITOR MUST HAVE A GRAPHICS CAPABILITY WITH
C     CGA, OCGA, EGA, OEGA, VGA, OVGA, MCGA, OR HGC
C     PRINTER MUST BE IN GRAPHICS MODE
C     'PRINT SCREEN' COMMAND WILL PRODUCE A HARD COPY
C
      INCLUDE 'FGRAPH.FI'
      INCLUDE 'FGRAPH.FD'
      INTEGER*2 DUMMY,MAXX,MAXY,NEWX,NEWY
      INTEGER*2 XP(96),YP(96),XX,YY
      REAL*4 X(96),Y(96)
      DIMENSION LM(100,4)
      RECORD/XYCOORD/XY
      RECORD/VIDEOCONFIG/MYSCREEN
      COMMON MAXX,MAXY
      EXTERNAL NEWX,NEWY
C
C     READ DATA FROM FILE 'XYDATA'
C
      OPEN(UNIT=4,FILE='XYDATA',STATUS='OLD')
C
C     NPT    = NO. OF NODES
C     NELEM  = NO. OF ELEMENTS
C     LM(I,J)= J TH NODE NUMBER OF ELEMENT I
C          J = 1,2 FOR BENDING ELEMENTS
C
      READ(4,*)NPT
      DO 70 I=1,NPT
   70 READ(4,*)X(I),Y(I)
      READ(4,*)NELEM
      DO 75 I=1,NELEM
   75 READ(4,*)(LM(I,J),J=1,4)
C
C     FIND THE LEFTMOST AND UPPERMOST COORDINATES
C
      X85=X(1)
      Y1=Y(1)
      DO 210 I=2,NPT
      IF(Y(I) .GT. Y1)Y1=Y(I)
  210 IF(X(I) .LT. X85)X85=X(I)
C
C     CHANGE TO MSFORTRAN GRAPHICS COORDINATES
C
      DO 80 I=1,NPT
      X(I)=X(I)-X85
   80 Y(I)=-Y(I)+Y1
C
C     FIND THE MAXIMUM VALUES OF X & Y COORDINATES
```

```
C
      XP6=X(1)
      YP85=Y(1)
      DO 220 I=2,NPT
      IF(X(I) .GT. XP6)XP6=X(I)
  220 IF(Y(I) .GT. YP85)YP85=Y(I)
C
C     FIND CURRENT GRAPHICS MODE
C
      CALL GETVIDEOCONFIG(MYSCREEN)
      SELECT CASE(MYSCREEN.ADAPTER)
      CASE($CGA)
      DUMMY=SETVIDEOMODE($HRESBW)
      CASE($OCGA)
      DUMMY=SETVIDEOMODE($ORESCOLOR)
      CASE($EGA,$OEGA)
      IF(MYSCREEN.MONITOR .EQ. $MONO)THEN
      DUMMY=SETVIDEOMODE($ERESNOCOLOR)
      ELSE
      DUMMY=SETVIDEOMODE($ERESCOLOR)
      END IF
      CASE($VGA,$OVGA,$MCGA)
      DUMMY=SETVIDEOMODE($VRES2COLOR)
      CASE($HGC)
      DUMMY=SETVIDEOMODE($HERCMONO)
      CASE DEFAULT
      DUMMY=0
      END SELECT
      IF(DUMMY .NE. 0)GO TO 5
      CALL SETTEXTPOSITION(1,1,XY)
      WRITE(*,'('' ERROR:CAN NOT SET GRAPHICS MODE'',\)')
      STOP
    5 CONTINUE
C
C     DETERMINE THE MAXIMUM AND MINIMUM DIMENSIONS
C
      CALL GETVIDEOCONFIG(MYSCREEN)
      IX=MYSCREEN.NUMTEXTCOLS
      IY=MYSCREEN.NUMTEXTROWS
      MAXX=MYSCREEN.NUMXPIXELS-1
      MAXY=MYSCREEN.NUMYPIXELS-1
C
C     NORMALIZE THE COORDINATES
C
      DO 10 K=1,NPT
      XP(K)=NEWX(X(K),XP6)
   10 YP(K)=NEWY(Y(K),YP85)
C
C     DRAW LINES FOR FINITE ELEMENT GRID
C     #FFFF IS FOR A STRAIGHT LINE, I.E., CONTINUUM ELEMENTS
C     #8888 IS FOR A DOT LINE, I.E., BENDING ELEMENTS
C
      DO 20 I=1,NELEM
```

```
      KK=4
      IF(LM(I,3) .EQ. 0)KK=1
      DO 22 K=1,KK
      M1=LM(I,K)
      IF(K .EQ. 4)M2=LM(I,1)
      IF(K .NE. 4)M2=LM(I,K+1)
      IJ=0
C
C     CHECK WHETHER THIS SIDE IS CONNECTED WITH OTHER ELEMENT
C
      DO 30 J=I+1,NELEM
      LL=4
      IF(LM(J,3) .EQ. 0)LL=1
      DO 32 L=1,LL
      L1=LM(J,L)
      IF(L .EQ. 4)L2=LM(J,1)
      IF(L .NE. 4)L2=LM(J,L+1)
      IF(M1 .EQ. L1 .AND. M2 .EQ. L2)IJ=1
      IF(M1 .EQ. L2 .AND. M2 .EQ. L1)IJ=1
   32 CONTINUE
   30 CONTINUE
      IF(IJ .EQ. 1)GO TO 22
C
C     DRAW LINE
C
      IF(LM(I,3) .EQ. 0)CALL SETLINESTYLE(#8888)
      IF(LM(I,3) .NE. 0)CALL SETLINESTYLE(#FFFF)
      CALL MOVETO(XP(M1),YP(M1),XY)
      DUMMY=LINETO(XP(M2),YP(M2))
   22 CONTINUE
   20 CONTINUE
C
      CALL SETTEXTPOSITION(1,1,XY)
      WRITE(*,'('' DOTS FOR BENDING ELEMENTS'',\)')
C
C     LABEL AS MANY NODES AS INPUTTED
C
      CALL SETTEXTPOSITION(2,1,XY)
      WRITE(*,'(''NODE NUMBERS(-1 FOR ALL)'',\)')
  120 CALL SETTEXTPOSITION(3,1,XY)
      WRITE(*,'(''RETURN TO QUIT'',\)')
      CALL SETTEXTPOSITION(4,1,XY)
      WRITE(*,'(''NUMBER?'')')
      READ(*,100)I
  100 FORMAT(I3)
      IF(I .EQ. 0)GO TO 110
      IF(I .EQ. -1)GO TO 122
      IF(I .LE. 0 .OR. I .GT. NPT)GO TO 120
      XX=XP(I)/REAL(MAXX+1)*IX
      YY=YP(I)/REAL(MAXY+1)*IY
      CALL SETTEXTPOSITION(YY+1,XX,XY)
      WRITE(*,'(I3,\)')I
      GO TO 120
```

```
  122 DO 124 II=1,NPT
      XX=XP(II)/REAL(MAXX+1)*IX
      YY=YP(II)/REAL(MAXY+1)*IY
      CALL SETTEXTPOSITION(YY+1,XX,XY)
  124 WRITE(*,'(I3,\)')II
  110 CONTINUE
      CALL SETTEXTPOSITION(5,1,XY)
      WRITE(*,'('' '',\)')
C
C     LABEL AS MANY ELEMENTS AS INPUTTED
C
      CALL SETTEXTPOSITION(2,1,XY)
      WRITE(*,'(''ELEMENT NOS.(-1 FOR ALL)'',\)')
  130 CALL SETTEXTPOSITION(3,1,XY)
      WRITE(*,'(''RETURN TO QUIT'',\)')
      CALL SETTEXTPOSITION(4,1,XY)
      WRITE(*,'(''NUMBER?'')')
      READ(*,100)I
      IF(I .EQ. 0)GO TO 140
      IF(I .EQ. -1)GO TO 132
      IF(I .LE. 0 .OR. I .GT. NELEM)GO TO 130
      JX=4
      IF(LM(I,3) .EQ. 0)JX=2
      XA=0.
      YA=0.
      DO 160 J=1,JX
      XA=XA+XP(LM(I,J))/JX
  160 YA=YA+YP(LM(I,J))/JX
      XX=XA/REAL(MAXX+1)*IX
      YY=YA/REAL(MAXY+1)*IY
      CALL SETTEXTPOSITION(YY+1,XX,XY)
      IF(I .LT. 100)WRITE(*,'(I3,\)')I
      IF(I .GE. 100)WRITE(*,'(I4,\)')I
      GO TO 130
  132 DO 134 II=1,NELEM
      JX=4
      IF(LM(II,3) .EQ. 0)JX=2
      XA=0.
      YA=0.
      DO 162 J=1,JX
      XA=XA+XP(LM(II,J))/JX
  162 YA=YA+YP(LM(II,J))/JX
      XX=XA/REAL(MAXX+1)*IX
      YY=YA/REAL(MAXY+1)*IY
      CALL SETTEXTPOSITION(YY+1,XX,XY)
      IF(II .LT. 100)WRITE(*,'(I3,\)')II
      IF(II .GE. 100)WRITE(*,'(I4,\)')II
  134 CONTINUE
  140 CONTINUE
C
      CALL SETTEXTPOSITION(2,1,XY)
      WRITE(*,'('' PRINT SCREEN FOR HARD COPY '',\)')
      CALL SETTEXTPOSITION(3,1,XY)
```

```
      WRITE(*,'(''                    '',\)')
      CALL SETTEXTPOSITION(4,1,XY)
      WRITE(*,'(''          '',\)')
      CALL SETTEXTPOSITION(5,1,XY)
      WRITE(*,'(''      '',\)')
      READ(*,*)   !PRESS ENTER TO EXIT
      DUMMY=SETVIDEOMODE($DEFAULTMODE)
      STOP
      END
C
C     FUNCTION TO RESCALE X COORDINATES
C
      INTEGER*2 FUNCTION NEWX(XCOORD,XP6)
      INTEGER*2 MAXX,MAXY
      REAL*4 TEMPX
      COMMON MAXX,MAXY
      TEMPX=MAXX/XP6
      TEMPX=XCOORD*TEMPX*0.9
      NEWX=TEMPX+0.05*MAXX
      RETURN
      END
C
C     FUNCTION TO RESCALE Y COORDINATES
C
      INTEGER*2 FUNCTION NEWY(YCOORD,YP85)
      INTEGER*2 MAXX,MAXY
      REAL*4 TEMPY
      COMMON MAXX,MAXY
      TEMPY=MAXY/YP85
      TEMPY=YCOORD*TEMPY*0.9
      NEWY=TEMPY+0.05*MAXY
      RETURN
      END
```

```
C
C     THIS PROGRAM DRAWS THE FINITE ELEMENT GRID ASSOCIATED WITH THE
C     COMPUTER PROGRAM BART OR BULK.  THE NECESSARY INPUT DATA ARE
C     WRITTEN BY THE PROGRAM BART OR BULK ONTO FILE 'XYDATA' AND READ
C     BY THIS PROGRAM.  IT IS PREFERRED TO USE A HARD DISC FOR ALL
C     FILES.  THIS PROGRAM REQUIRES THE FOLLOWING.
C     MONITOR MUST HAVE A GRAPHICS CAPABILITY WITH
C     CGA, OCGA, EGA, OEGA, VGA, OVGA, MCGA, OR HGC
C     PRINTER MUST BE IN GRAPHICS MODE
C     'PRINT SCREEN' COMMAND WILL PRODUCE A HARD COPY
C
      INCLUDE 'FGRAPH.FI'
      INCLUDE 'FGRAPH.FD'
      INTEGER*2 DUMMY,MAXX,MAXY,NEWX,NEWY
      INTEGER*2 XP(96),YP(96),XX,YY
      REAL*4 X(96),Y(96)
      DIMENSION LM(100,4)
      RECORD/XYCOORD/XY
      RECORD/VIDEOCONFIG/MYSCREEN
      COMMON MAXX,MAXY
      EXTERNAL NEWX,NEWY
C
C     READ DATA FROM FILE 'XYDATA'
C
      OPEN(UNIT=4,FILE='XYDATA',STATUS='OLD')
C
C     NPT    = NO. OF NODES
C     NELEM  = NO. OF ELEMENTS
C     LM(I,J)= J TH NODE NUMBER OF ELEMENT I
C          J = 1,2 FOR BENDING ELEMENTS
C
      READ(4,*)NPT
      DO 70 I=1,NPT
   70 READ(4,*)X(I),Y(I)
      READ(4,*)NELEM
      DO 75 I=1,NELEM
   75 READ(4,*)(LM(I,J),J=1,4)
C
C     FIND THE LEFTMOST AND UPPERMOST COORDINATES
C
      X85=X(1)
      Y1=Y(1)
      DO 210 I=2,NPT
      IF(Y(I) .GT. Y1)Y1=Y(I)
  210 IF(X(I) .LT. X85)X85=X(I)
C
C     CHANGE TO MSFORTRAN GRAPHICS COORDINATES
C
      DO 80 I=1,NPT
      X(I)=X(I)-X85
   80 Y(I)=-Y(I)+Y1
C
C     FIND THE MAXIMUM VALUES OF X & Y COORDINATES
```

```
C
      XP6=X(1)
      YP85=Y(1)
      DO 220 I=2,NPT
      IF(X(I) .GT. XP6)XP6=X(I)
  220 IF(Y(I) .GT. YP85)YP85=Y(I)
C
C     FIND CURRENT GRAPHICS MODE
C
      CALL GETVIDEOCONFIG(MYSCREEN)
      SELECT CASE(MYSCREEN.ADAPTER)
      CASE($CGA)
      DUMMY=SETVIDEOMODE($HRESBW)
      CASE($OCGA)
      DUMMY=SETVIDEOMODE($ORESCOLOR)
      CASE($EGA,$OEGA)
      IF(MYSCREEN.MONITOR .EQ. $MONO)THEN
      DUMMY=SETVIDEOMODE($ERESNOCOLOR)
      ELSE
      DUMMY=SETVIDEOMODE($ERESCOLOR)
      END IF
      CASE($VGA,$OVGA,$MCGA)
      DUMMY=SETVIDEOMODE($ERESCOLOR)
      CASE($HGC)
      DUMMY=SETVIDEOMODE($HERCMONO)
      CASE DEFAULT
      DUMMY=0
      END SELECT
      IF(DUMMY .NE. 0)GO TO 5
      CALL SETTEXTPOSITION(1,1,XY)
      WRITE(*,'('' ERROR:CAN NOT SET GRAPHICS MODE'',\)')
      STOP
    5 CONTINUE
C
C     DETERMINE THE MAXIMUM AND MINIMUM DIMENSIONS
C
      CALL GETVIDEOCONFIG(MYSCREEN)
      IX=MYSCREEN.NUMTEXTCOLS
      IY=MYSCREEN.NUMTEXTROWS
      MAXX=MYSCREEN.NUMXPIXELS-1
      MAXY=MYSCREEN.NUMYPIXELS-1
C
C     NORMALIZE THE COORDINATES
C
      DO 10 K=1,NPT
      XP(K)=NEWX(X(K),XP6)
   10 YP(K)=NEWY(Y(K),YP85)
C
C     DRAW LINES FOR FINITE ELEMENT GRID
C     #FFFF IS FOR A STRAIGHT LINE, I.E., CONTINUUM ELEMENTS
C     #8888 IS FOR A DOT LINE, I.E., BENDING ELEMENTS
C
      DO 20 I=1,NELEM
```

```
      KK=4
      IF(LM(I,3) .EQ. 0)KK=1
      DO 22 K=1,KK
      M1=LM(I,K)
      IF(K .EQ. 4)M2=LM(I,1)
      IF(K .NE. 4)M2=LM(I,K+1)
      IJ=0
C
C     CHECK WHETHER THIS SIDE IS CONNECTED WITH OTHER ELEMENT
C
      DO 30 J=I+1,NELEM
      LL=4
      IF(LM(J,3) .EQ. 0)LL=1
      DO 32 L=1,LL
      L1=LM(J,L)
      IF(L .EQ. 4)L2=LM(J,1)
      IF(L .NE. 4)L2=LM(J,L+1)
      IF(M1 .EQ. L1 .AND. M2 .EQ. L2)IJ=1
      IF(M1 .EQ. L2 .AND. M2 .EQ. L1)IJ=1
   32 CONTINUE
   30 CONTINUE
      IF(IJ .EQ. 1)GO TO 22
C
C     DRAW LINE
C
      IF(LM(I,3) .EQ. 0)CALL SETLINESTYLE(#8888)
      IF(LM(I,3) .NE. 0)CALL SETLINESTYLE(#FFFF)
      CALL MOVETO(XP(M1),YP(M1),XY)
      DUMMY=LINETO(XP(M2),YP(M2))
   22 CONTINUE
   20 CONTINUE
C
      CALL SETTEXTPOSITION(1,1,XY)
      WRITE(*,'('' DOTS FOR BENDING ELEMENTS'',\)')
C
C     LABEL AS MANY NODES AS INPUTTED
C
      CALL SETTEXTPOSITION(2,1,XY)
      WRITE(*,'(''NODE NUMBERS(-1 FOR ALL)'',\)')
  120 CALL SETTEXTPOSITION(3,1,XY)
      WRITE(*,'(''RETURN TO QUIT'',\)')
      CALL SETTEXTPOSITION(4,1,XY)
      WRITE(*,'(''NUMBER?'')')
      READ(*,100)I
  100 FORMAT(I3)
      IF(I .EQ. 0)GO TO 110
      IF(I .EQ. -1)GO TO 122
      IF(I .LE. 0 .OR. I .GT. NPT)GO TO 120
      XX=XP(I)/REAL(MAXX+1)*IX
      YY=YP(I)/REAL(MAXY+1)*IY
      CALL SETTEXTPOSITION(YY+1,XX,XY)
      WRITE(*,'(I3,\)')I
      GO TO 120
```

```
  122 DO 124 II=1,NPT
      XX=XP(II)/REAL(MAXX+1)*IX
      YY=YP(II)/REAL(MAXY+1)*IY
      CALL SETTEXTPOSITION(YY+1,XX,XY)
  124 WRITE(*,'(I3,\)')II
  110 CONTINUE
      CALL SETTEXTPOSITION(5,1,XY)
      WRITE(*,'('' '',\)')
C
C     LABEL AS MANY ELEMENTS AS INPUTTED
C
      CALL SETTEXTPOSITION(2,1,XY)
      WRITE(*,'(''ELEMENT NOS.(-1 FOR ALL)'',\)')
  130 CALL SETTEXTPOSITION(3,1,XY)
      WRITE(*,'(''RETURN TO QUIT'',\)')
      CALL SETTEXTPOSITION(4,1,XY)
      WRITE(*,'(''NUMBER?'')')
      READ(*,100)I
      IF(I .EQ. 0)GO TO 140
      IF(I .EQ. -1)GO TO 132
      IF(I .LE. 0 .OR. I .GT. NELEM)GO TO 130
      JX=4
      IF(LM(I,3) .EQ. 0)JX=2
      XA=0.
      YA=0.
      DO 160 J=1,JX
      XA=XA+XP(LM(I,J))/JX
  160 YA=YA+YP(LM(I,J))/JX
      XX=XA/REAL(MAXX+1)*IX
      YY=YA/REAL(MAXY+1)*IY
      CALL SETTEXTPOSITION(YY+1,XX,XY)
      IF(I .LT. 100)WRITE(*,'(I3,\)')I
      IF(I .GE. 100)WRITE(*,'(I4,\)')I
      GO TO 130
  132 DO 134 II=1,NELEM
      JX=4
      IF(LM(II,3) .EQ. 0)JX=2
      XA=0.
      YA=0.
      DO 162 J=1,JX
      XA=XA+XP(LM(II,J))/JX
  162 YA=YA+YP(LM(II,J))/JX
      XX=XA/REAL(MAXX+1)*IX
      YY=YA/REAL(MAXY+1)*IY
      CALL SETTEXTPOSITION(YY+1,XX,XY)
      IF(II .LT. 100)WRITE(*,'(I3,\)')II
      IF(II .GE. 100)WRITE(*,'(I4,\)')II
  134 CONTINUE
  140 CONTINUE
C
      CALL SETTEXTPOSITION(2,1,XY)
      WRITE(*,'('' PRINT SCREEN FOR HARD COPY '',\)')
      CALL SETTEXTPOSITION(3,1,XY)
```

```
      WRITE(*,'('' '',\)')
      CALL SETTEXTPOSITION(4,1,XY)
      WRITE(*,'('' '',\)')
      CALL SETTEXTPOSITION(5,1,XY)
      WRITE(*,'('' '',\)')
      READ(*,*)   !PRESS ENTER TO EXIT
      DUMMY=SETVIDEOMODE($DEFAULTMODE)
      STOP
      END
C
C     FUNCTION TO RESCALE X COORDINATES
C
      INTEGER*2 FUNCTION NEWX(XCOORD,XP6)
      INTEGER*2 MAXX,MAXY
      REAL*4 TEMPX
      COMMON MAXX,MAXY
      TEMPX=MAXX/XP6
      TEMPX=XCOORD*TEMPX*0.9
      NEWX=TEMPX+0.05*MAXX
      RETURN
      END
C
C     FUNCTION TO RESCALE Y COORDINATES
C
      INTEGER*2 FUNCTION NEWY(YCOORD,YP85)
      INTEGER*2 MAXX,MAXY
      REAL*4 TEMPY
      COMMON MAXX,MAXY
      TEMPY=MAXY/YP85
      TEMPY=YCOORD*TEMPY*0.9
      NEWY=TEMPY+0.05*MAXY
      RETURN
      END
```