Appendix C
User's Guide for EPOLIQAN Software

C.1. Introduction to the EPOLIQAN Program.

A computer program called EPOLIQAN, for *EPOLLS LIQuefaction ANalysis*, was written to analyze soil borings and compute parameters for the EPOLLS database. To document the procedures used in computing the EPOLLS geotechnical parameters, a user's guide and the source code for EPOLIQAN are given in this appendix. Written in FORTRAN, this code evaluates the liquefaction potential based on Standard Penetration Test (SPT) blowcounts and computes the EPOLLS geotechnical parameters. The empirical model used in EPOLIQAN to evaluate liquefaction resistance based on SPT blowcounts is described in Chapter 7. Details on the EPOLLS geotechnical parameters computed by the code are given in Chapter 6. Note that the EPOLIQAN code only computes the EPOLLS geotechnical parameters for individual soil borings. In compiling the EPOLLS database, the average (*Avg*-) and range (*Rng*-) values were computed manually using the EPOLIQAN output for every soil boring at a case study site.

For this study, EPOLIQAN was used on a personal computer running the MS-DOS 5.0 operating system. Using a FORTRAN-77 compatible compiler, the EPOLIQAN source code can be compiled to run under any computer operating system. However, the program assumes a DOS-compatible system for the naming of input and output files, so it may be necessary to alter the source code (in the subroutine "OPENFILES") to follow the file-naming conventions of another operating system. Once an executable file has been created, the code is run by following the instructions in Figure C.1.

The input and output data files for EPOLIQAN are simple text files (ASCII format) which can be created or read using any suitable software editor. Specifications and general notes for the EPOLIQAN input data file are given in Section C.2. An example input data file is shown in Figure C.3 with the corresponding example output file shown in Figure C.4 of Section C.3. The output file from EPOLIQAN contains all of the EPOLLS parameters for each soil boring in the input data file.

To analyze a soil boring log using EPOLIQAN:

(1) Create an executable version of the EPOLIQAN source code using a FORTRAN-77 compatible compiler.

(2) Following the input data specifications, create an input data file (in ASCII format). Give this file a name with a ".DAT" filetype. For example, call the file "SITE-1.DAT".

(3) Execute the EPOLIQAN program.

(4) At the prompt, enter the file name of the input file. The ".DAT" filetype is assumed and should not be specified. For example:

```
..... Enter filename for boring logs
.....      ( .DAT filetype assumed ):
SITE-1 <return>
```

(5) The program will then analyze the soil boring data and write the results to a new ASCII file automatically given the name "SITE-1.LIQ". This output can then be read or printed as desired.

Figure C.1. Instructions for running the EPOLIQAN software.

C.2. Input Data File for EPOLIQAN.

Specifications for the input data file read by EPOLIQAN are given in Figure C.2. The required input variables are shown in boldface in this figure and defined on the pages that follow. General notes on the EPOLIQAN input data file include:

- All of the input data are read free-format by EPOLIQAN. Therefore, a value must be given, in the specified order, for every variable that is read by the code.

- Several lines of column headings are used to enhance the readability of the input data file. These lines (indicated with * in Figure C.2) are not actually read by the program, but are skipped; therefore, these column heading lines *must* be included in the input file.

- Depths are positive values measured from the ground surface.

- Hydrostatic pore water pressures are assumed and computed from the water table depth. When the ground surface is submerged (for example, when a boring is conducted in the bed of a stream), specify the ground water table depth (**DWT**) as a negative value measured from the ground surface up to the water surface.

- The EPOLIQAN code works in units of g (acceleration), meters (lengths), and kN/m^3 (density). The input data can be given in any other set of units and the program will multiply by the specified unit conversion factors **CFG**, **CFM**, and **CFK**. Note that the unit conversion factors are applied to *all* input data. Hence, use a consistent set of units for all data in one boring record; for example, if **DEPTHN** is given in feet, also give **DWT**, **SDEPTP**, **SPLTOP**, and **SPLBOT** in feet.

- Soil strata properties, SPT blowcounts, and specified liquefied sublayers for each boring are given on individual lines. Repeat these lines as many times as required to enter all of this data for each boring log. To denote the end of these three sections of data, type the delimiter "BOTTOM" or "bottom" on a single line.

- The soil strata, measured SPT blowcounts, and specified liquefied sublayers can be given in any order in the input file and the program will automatically sort them into order of increasing depth.

- To include more than one soil boring in one input file, simply repeat the input data set. That is, after the last line for "Boring #1", give the same required data for "Boring #2". The program will process each set of data until all of the borings in the given input file have been analyzed.

Liquefied soil sublayers can be optionally specified in the EPOLIQAN input data set. If this option is not desired, do not include a line for **SPLTOP** and **SPLBOT** (but do include the column heading line and "bottom"). When liquefied sublayers are specified, the program will calculate two sets of the EPOLLS parameters: one set based on the sublayers predicted to liquefy from the SPT profile, and a second set based on the specified sublayers (see additional comments in Section C.3). Use this option when engineering judgment supersedes the liquefied sublayers predicted by the code. For example, a preliminary analysis made with EPOLIQAN may show no liquefaction in a thin sublayer within a larger, liquefied deposit due to a single high SPT blowcount. If this thin sublayer is expected to actually liquefy, it can be included in the computed average EPOLLS parameters for the liquefied thickness by using the option for specified liquefied sublayers.

Boring Number		**BORGID**
Site Parameters, Unit Conversions	*	EQ Mw Amax Depth WT CFg CFm CFkN/m3 **EQM AMAX DWT CFG CFM CFK**
Properties of Soil Strata	*	Depth Top Unit Wt D50 Cu %Fines %Clay Pot. Liq? **SDEPTP SGAMMA SD50 SCU SPF SPC SPLIQ** *(repeat this line for each soil strata in profile)*
	+	**bottom**
Corrections for SPT	*	EnergyRatio Borehole Sampler **ER CB CS**
SPT Blowcounts and Sample Properties	*	Depth Nspt D50 Cu %Fines %Clay **DEPTHN SPTN D50 CU PF PC** *(repeat this line for each measured blowcount in boring log)*
	+	**bottom**
Specified Liquefied Sublayers	*	LiqTop LiqBottom **SPLTOP SPLBOT** *(repeat this line for each specified layer, skip if none specified)*
	+	**bottom**
Requested Output	*	SPT locations? Continuous Profile? EPOLLS Parameters? **OUTSPT OUTCPF OUTEPL**

* Column heading lines that are not read, but must be included in the input file.
\+ Must give the delimiter "**bottom**" or "**BOTTOM**" to indicate end of data in this section.

Figure C.2. Specifications for the EPOLIQAN input data file.

Definition of Parameters:

Boring Number

BORGID Identification for boring log, maximum length of 72 characters.

Site Parameters, Unit Conversions

EQM Moment magnitude of earthquake.
AMAX Peak horizontal acceleration at the ground surface that would occur in the absence of excess pore pressures or liquefaction generated by the earthquake.
DWT Depth from the surface to the ground water table.
CFG Multiplication factor for converting **AMAX** to units of g.
Example: If acceleration given in m/sec^2, specify **CFG** = 0.102 $(g)/(m/sec^2)$.
CFM Multiplication factor for converting depths (**DWT**, **SDEPTP** and **DEPTH**) to units of m.
Example: If depths are entered in feet, specify **CFM** = 0.3048 meters/foot.
CFK Multiplication factor for converting **SGAMMA** values to units of kN/m^3.
Example: If unit weights are entered in lb/ft^3, specify
CFK = 0.1571 $(kN/m^3)/(lb/ft^3)$.

Properties of Soil Strata

SDEPTP Depth from ground surface to top of soil stratum.
SGAMMA Total, wet unit weight of soil stratum.
SD50 * Mean grain size (D_{50}) representative of soil stratum (*in units of* ***mm*** *only)*.
SCU * Uniformity coefficient ($C_u=D_{60}/D_{10}$) representative of soil stratum.
SPF * Fines content (percent by weight finer than 0.075 mm) representative of soil stratum.
SPC * Clay content (percent by weight finer than 0.005 mm) representative of soil stratum.
SPLIQ "T" if the soil stratum is potentially liquefiable;
"F" if the soil stratum is not susceptible to liquefaction (perhaps due to high clay content).

** If value is unknown, enter a negative number.*

Corrections for SPT

ER SPT hammer energy ratio (%) = actual energy delivered to top of drill rod stem divided by the theoretical maximum energy delivered by a SPT hammer (4200 lb-in).
CB Correction factor for borehole diameter greater than 115 mm (suggested values given in Section 7.3).
CS Correction factor for non-standard SPT sampler (give **CS** = 1.2 for standard sampler used without liner inside).

SPT Blowcounts and Sample Properties

DEPTHN Depth from ground surface to midpoint of measured **SPTN**.
SPTN Measured Standard Penetration Test (SPT) blowcount (blows per foot).
D50 * Mean grain size (D_{50}) of soil sample from **DEPTHN** (*in units of **mm** only)*.
CU * Uniformity coefficient (C_u=D_{60}/D_{10}) of soil sample from **DEPTHN**.
PF * Fines content (percent by weight finer than 0.075 mm) of soil sample from **DEPTHN**.
PC * Clay content (percent by weight finer than 0.005 mm) of soil sample from **DEPTHN**.

** If value is unknown, enter a negative number and the values specified above for the appropriate soil stratum will be assumed for this* **DEPTHN.**

Specified Liquefied Sublayers

SPLTOP Depth to top of sublayer specified by user to liquefy.
SPLBOT Depth to bottom of sublayer specified by user to liquefy.

Note: Leave this line blank (but include the column heading line and **bottom***) if do not want to specify liquefied sublayers.*

Requested Output

OUTSPT "T" produces output for each SPT location; "F" suppresses this output.
OUTCPF "T" produces output for the continuous profile interpolated by EPOLIQAN at depth increments of 0.1 m; "F" suppresses this output.
OUTEPL "T" produces output for all EPOLLS geotechnical parameters; "F" suppresses this output.

```
---> BORING #1
  EQ Mw    Amax     Depth WT   CFg     CFm     CFkN/m3
   7.9     0.21       6.0      1.0    .3048    0.1571
 Depth Top   Unit Wt    D50    Cu   % fines   % clay   Pot.Liq?
   0.0       120.0      1.2   20.0     8.0      2.0      True
  30.0       125.0     0.03   10.0    82.0     12.0      True
  44.5       120.0      1.2   20.0     8.0      2.0      T
  13.0       125.0      6.0   50.0     5.0      0.0      true
bottom
EnergyRatio  Borehole  Sampler
   60          1.00       1.0
  Depth      Nspt         D50      Cu   % fines   % clay
   1.0       15           0.2     3.6      8       2.
   4.0       12           1.2     50.      6       1.
   7.0        6           0.65    10.      6       1.
   9.0        9           -1      -1      -1      -1
  13.0       31           4.0     73.      5       2.
  20.0       14           6.0     57.      2       0.
  30.0       10           0.035    5.     98       7.
  34.5       14           0.025   10.     99      17.
  40.5        9           0.045   15.     65      13.
  42.0       10           0.025   -1      72      23.
  44.5       17           0.8      2.5     7       3.
  49.5       22           0.28    14.     20       4.
BOTTOM
LiqTop     LiqBottom
 6.0         47.0
bottom
SPT Locations?  Continuous Profile?  EPOLLS Parameters?
    T               F                    T
---> BORING #2
  EQ Mw    Amax     Depth WT   CFg     CFm     CFkN/m3
   7.9     0.21       1.2      1.0     1.0      1.0
 Depth Top   Unit Wt    D50    Cu   % fines   % clay   Pot.Liq?
   0.0       18.9       1.2   20.0     8.0      2.0      T
   7.3       19.6       0.03  10.0    82.0     12.0      t
  12.0       19.6       0.2    7.3    20.0      2.0      t
BOTTOM
EnergyRatio  Borehole  Sampler
   60          1.00      1.20
 Depth      Nspt         D50      Cu   % fines   % clay
  13.6       54           0.19     3.6    12        1.
  12.0       45           0.08    24.     48       10.
  10.4       27           0.03     7.6    95       10.
   8.8       23           0.03     13.    89       15.
   7.3       12           -1.     -1.     -1       -1
   5.6       11           1.9     15.      3        0.
   2.7       13           2.1     19.      4        0.
   4.6        9           0.6      8.4     7        2.
   1.2       15           0.5     11.      8        4.
bottom
LiqTop     LiqBottom
BOTTOM
SPT Locations?  Continuous Profile?  EPOLLS Parameters?
    True            False                True
```

Figure C.3. Example input data file for EPOLIQAN.

C.3. Output Data from EPOLIQAN.

Shown in Figure C.4 is the output file produced by EPOLIQAN from the example input in Figure C.3. The input data is written into the output file along with the evaluation of liquefaction potential at each SPT location. The continuous profile, produced by interpolation at depth increments of 0.1 m, can also be written to the output file, although this output is not shown in Figure C.4. The geotechnical parameters defined for EPOLLS are given in the last section of the output file. Unless otherwise indicated, all of the output from EPOLIQAN is presented in units of g, m, kN/m^3, and kPa (D_{50} is always given in mm).

The code always predicts the thickness of liquefied soil based on the given SPT data and the interpolated, continuous profile. As indicated earlier, the user has the option to also specify that certain sublayers within the profile will liquefy. Specifically, this allows the user to override the liquefied soil sublayers predicted by EPOLIQAN and used to compute the EPOLLS parameters. When this option is used, the EPOLLS parameters are calculated twice:

(1) Based on the given data where liquefaction is predicted. For example, *N160_Liq* is computed as the average of all $(N_1)_{60}$ values where the factor of safety is less than one.

(2) Based on the given data within the specified liquefied sublayers. Now, *N160_Liq* is computed as the average of all $(N_1)_{60}$ values determined within the specified thickness.

Both sets of EPOLLS parameters are written in the output file, and it is left to the user to select which values are more appropriate for the EPOLLS model. See the discussion in Section 6.7 and the list in Table 6.1 for the EPOLLS parameters that are affected by the predicted versus specified liquefied sublayers.

Finally, the code computes two measures of liquefied thickness (*Thick_Liq*), as discussed in Section 6.7, for both the predicted and specified liquefied sublayers. These measures are:

(1) "Sum liquefied thickness" is the summation of the thicknesses of all sublayers either predicted or specified to liquefy.

(2) "Gross liquefied thickness" is the net thickness from the shallowest to the deepest occurrence of liquefaction as either predicted or specified (equal to *Z_BotLiq* - *Z_TopLiq*).

Again, it is left to the user to select which measure best represents the thickness of liquefied soil in a lateral spread for the EPOLLS model (see the discussion in Section 6.7).

```
-- Output File Name: EXAMPLE.LIQ
((((((((((((((((((((((((((  EPOLIQAN  ))))))))))))))))))))))))))
(                                                              )
(                    EPOLIQAN Version 2.0                      )
(               EPOLLS Liquefaction Analysis Code              )
(                                                              )
((((((((((((((((((((((((((  EPOLIQAN  ))))))))))))))))))))))))))

---> BORING #1


     (values in meters, kN, and kPa unless otherwise stated)
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
    EQ Mw   Amax(g)    Depth WT      CFg       CFm     CFkN/m3
     7.90  .210E+00      1.83      1.0000     .3048     .1571

...Site soil strata and properties...................................
  Depth Top  Unit Wt   D50(mm)      Cu     % fines   % clay  Pot.Liq?
       .00     18.85      1.20     20.00      8.00     2.00      T
      3.96     19.64      6.00     50.00      5.00      .00      T
      9.14     19.64       .03     10.00     82.00    12.00      T
     13.56     18.85      1.20     20.00      8.00     2.00      T
   < Bottom >
...Measured SPT values matched with soil properties.................
....SPT hammer Energy Ratio (%)         =  60.0
....Correction for borehole diameter    =   1.00
....Correction for non-standard sampler =   1.00
    Depth      Nspt    D50(mm)      Cu     % fines   % clay  Pot.Liq?
      .30      15.0       .20      3.60      8.00     2.00      T
     1.22      12.0      1.20     50.00      6.00     1.00      T
     2.13       6.0       .65     10.00      6.00     1.00      T
     2.74       9.0      1.20     20.00      8.00     2.00      T
     3.96      31.0      4.00     73.00      5.00     2.00      T
     6.10      14.0      6.00     57.00      2.00      .00      T
     9.14      10.0       .04      5.00     98.00     7.00      T
    10.52      14.0       .03     10.00     99.00    17.00      T
    12.34       9.0       .05     15.00     65.00    13.00      T
    12.80      10.0       .03     10.00     72.00    23.00      T
    13.56      17.0       .80      2.50      7.00     3.00      T
    15.09      22.0       .28     14.00     20.00     4.00      T
  < Bottom >

---Results of analysis for recorded Nspt values---------------------
    Depth     N1,60    Sig'Vo      CRR        CSR     FSliq      Liq??
      .30      22.5       5.7    -1.000      .136    -1.000       -n-
     1.22      18.0      23.0    -1.000      .135    -1.000       -n-
     2.13       7.4      37.2      .079      .145      .542     *LIQ*
     2.74      10.4      42.7      .107      .162      .658     *LIQ*
     3.96      33.6      53.8    -1.000      .184    -1.000       -n-
     6.10      14.3      74.8      .134      .203      .663     *LIQ*
     9.14       9.8     104.7      .157      .210      .750     *LIQ*
    10.52      13.0     118.2      .189      .209      .906     *LIQ*
    12.34       7.8     136.2      .138      .205      .675     *LIQ*
    12.80       8.5     140.7      .145      .203      .713     *LIQ*
    13.56      14.1     148.2      .136      .200      .679     *LIQ*
    15.09      17.4     161.9      .200      .194     1.029       -n-
  < Bottom >
---Liquefied sublayers-------PREDICTED by CODE----SPECIFIED by USER--
         Liquefied sublayer:     Top    Bottom        Top    Bottom
                          1     1.90     4.00        1.83    14.33
                          2     5.80    15.00

===Computed EPOLLS geotechnical parameters==========================

                             PREDICTED by CODE    SPECIFIED by USER
                             =================    =================
         Liquefied sublayer:      Thick_Liq            Thick_Liq
                          1          2.1                 12.5
                          2          9.2

   Sum liquefied thickness =        11.3                 12.5
 Gross liquefied thickness =        13.1                 12.5

                 Z_GrWater =         1.83                 1.83
                 Z_TopLiq  =         1.9                  1.8
                 Z_BotLiq  =        15.0                 14.3
                 Index_Liq =          .201                 .201
                 FS_Liq    =          .698                 .509
                 N160_Liq  =        10.66                13.21
                 FS_Min    =          .542                 .542
                 N160_MnFS =         7.42                 7.42
                 Z_MnFS    =         2.13                 2.13
                 N160_Min  =         7.42                 7.42
                 Z_MnN160  =         2.13                 2.13
                 D50_Liq   =         1.10                 1.42
                 Cu_Liq    =        16.19                22.50
                 Clay_Liq  =         8.25                 7.56
                 Fine_Liq  =        44.63                40.22
                 N160_Cap  =        20.25                20.25
                 Fine_Cap  =         7.00                 7.00
                 Fine_Base =        20.00                20.00
======================================================================
```

```
---> BORING #2


     (values in meters, kN, and kPa unless otherwise stated)
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
    EQ Mw   Amax(g)    Depth WT      CFg       CFm     CFkN/m3
     7.90  .210E+00      1.20      1.0000    1.0000    1.0000

...Site soil strata and properties...................................
  Depth Top  Unit Wt   D50(mm)      Cu     % fines   % clay  Pot.Liq?
       .00     18.90      1.20     20.00      8.00     2.00      T
      7.30     19.60       .03     10.00     82.00    12.00      T
     12.00     19.60       .20      7.30     20.00     2.00      T
   < Bottom >
...Measured SPT values matched with soil properties.................
....SPT hammer Energy Ratio (%)         =  60.0
....Correction for borehole diameter    =   1.00
....Correction for non-standard sampler =   1.20
    Depth      Nspt    D50(mm)      Cu     % fines   % clay  Pot.Liq?
     1.20      15.0       .50     11.00      8.00     4.00      T
     2.70      13.0      2.10     19.00      4.00      .00      T
     4.60       9.0       .60      8.40      7.00     2.00      T
     5.60      11.0      1.90     15.00      3.00      .00      T
     7.30      12.0       .03     10.00     82.00    12.00      T
     8.80      23.0       .03     13.00     89.00    15.00      T
    10.40      27.0       .03      7.60     95.00    10.00      T
    12.00      45.0       .08     24.00     48.00    10.00      T
    13.60      54.0       .19      3.60     12.00     1.00      T
  < Bottom >

---Results of analysis for recorded Nspt values---------------------
    Depth     N1,60    Sig'Vo      CRR        CSR     FSliq      Liq??
     1.20      27.0      22.7    -1.000      .135    -1.000       -n-
     2.70      19.5      36.3      .185      .188      .981     *LIQ*
     4.60      12.1      53.6      .119      .214      .557     *LIQ*
     5.60      14.4      62.7      .135      .221      .612     *LIQ*
     7.30      15.2      78.1      .216      .226      .955     *LIQ*
     8.80      28.6      92.8    -1.000      .226    -1.000       -n-
    10.40      31.3     108.5    -1.000      .223    -1.000       -n-
    12.00      48.8     124.2    -1.000      .218    -1.000       -n-
    13.60      55.2     139.8    -1.000      .211    -1.000       -n-
  < Bottom >
---Liquefied sublayers-------PREDICTED by CODE----SPECIFIED by USER--
         Liquefied sublayer:     Top    Bottom        Top    Bottom
                          1     2.60     7.40

===Computed EPOLLS geotechnical parameters==========================

                             PREDICTED by CODE    SPECIFIED by USER
                             =================    =================
         Liquefied sublayer:      Thick_Liq            Thick_Liq
                          1          4.8

   Sum liquefied thickness =         4.8                   .0
 Gross liquefied thickness =         4.8                   .0

                 Z_GrWater =         1.20
                 Z_TopLiq  =         2.6
                 Z_BotLiq  =         7.4
                 Index_Liq =          .126
                 FS_Liq    =          .776
                 N160_Liq  =        15.33
                 FS_Min    =          .557
                 N160_MnFS =        12.13
                 Z_MnFS    =         4.60
                 N160_Min  =        12.13
                 Z_MnN160  =         4.60
                 D50_Liq   =         1.16
                 Cu_Liq    =        13.10
                 Clay_Liq  =         3.50
                 Fine_Liq  =        24.00
                 N160_Cap  =        27.00
                 Fine_Cap  =         8.00
                 Fine_Base =        61.00
======================================================================
((((((((((((((((((((((((((  EPOLIQAN  ))))))))))))))))))))))))))
(.................. Execution complete - Normal exit ................)
((((((((((((((((((((((((((  EPOLIQAN  ))))))))))))))))))))))))))
```

Figure C.4. Example output data from EPOLIQAN.

C.4. Source Code Listing for EPOLIQAN.

```
C#######################################################################
C##                                                                   ##
C##   EPOLIQAN     Version 2.0                                        ##
C##     Program for computing geotechnical parameters for             ##
C##     the EPOLLS lateral spreading model.                           ##
C##         -  reads multiple SPT boring logs from single input file ##
C##         -  performs liquefaction analysis using method            ##
C##            recommended by consensus at 1996 NCEER Workshop        ##
C##         -  outputs the EPOLLS parameters for each boring          ##
C##                                                                   ##
C##   Written by:                                                     ##
C##          Alan F. Rauch                                            ##
C##          Dept. of Civil Engineering, Virginia Tech                ##
C##          Blacksburg, Virginia                                     ##
C##                                                                   ##
C##   Version 1.0 -- April, 1996                                      ##
C##   Version 2.0 -- September, 1996                                  ##
C##                                                                   ##
C##   NOTE:                                                           ##
C##     The user of this computer code assumes all responsibility    ##
C##     for verifying the accuracy and applicability of the           ##
C##     calculated results. Neither Virginia Tech nor Alan F. Rauch  ##
C##     make or assume liability for any warranties, expressed or    ##
C##     implied, concerning the accuracy, reliability, usefulness    ##
C##     or adaptability of this computer program.                     ##
C##                                                                   ##
C#######################################################################
      PROGRAM EPOLIQAN
      IMPLICIT REAL*4 (A-H,O-Z), INTEGER (I-N)
      LOGICAL FLGEND, SPLIQ, PLIQ, FLIQS, OUTSPT, OUTCPF, OUTEPL
      CHARACTER LIQBAN*5

      PARAMETER (MAXDEP = 100, MAXSPT = 500, MAXINT=1000)
      DIMENSION SDEPTP(MAXDEP), SGAMMA(MAXDEP), SD50(MAXDEP),
     +          SCU(MAXDEP), SPF(MAXDEP), SPC(MAXDEP), SPLIQ(MAXDEP)
      DIMENSION DEPTHN(MAXSPT), SPTN(MAXSPT), D50(MAXSPT), CU(MAXSPT),
     +          PF(MAXSPT), PC(MAXSPT), PLIQ(MAXSPT)
      DIMENSION SPTN160(MAXSPT), CSN160(MAXSPT), CRR(MAXSPT),
     +          FSLIQ(MAXSPT)
      DIMENSION CRRINT(MAXINT), CSRINT(MAXINT)
      DIMENSION BNDLIQS(MAXDEP,2), THKLIQ(2,MAXDEP), SPLTOP(MAXDEP),
     +          SPLBOT(MAXDEP), STHKLIQ(2)
      DIMENSION ZTOPLIQ(2), ZBOTLIQ(2), AVFSLIQ(2), SPTLIQ(2),
     +          D50LIQ(2), CULIQ(2), CLAYLIQ(2), FINELIQ(2),
     +          SPTCAP(2), FINECAP(2), FINEBOT(2)

C--Set calculation parameters:
C-- DINC = depth increment used to interpolate the continuous profile
      PARAMETER (DINC = 0.1)

C--Set max limits for memory allocation
      COMMON /MEMORY/ MDEP, MSPT, MINT
      MDEP = MAXDEP
      MSPT = MAXSPT
      MINT = MAXINT

      NBOR = 0

C--Display program banner to screen
      CALL BANNER(0,1)
C--Open input and output files
      CALL OPENFILES
C--Print program banner in output file
      CALL BANNER(2,1)

CRD---Begin processing of each boring in the input file
      FLGEND = .TRUE.
  10  NBOR = NBOR + 1
CRD---Read title info for this boring
      CALL GETTITLE(FLGEND, EQM, AMAX, DWT, CFM, CFK)
      IF (FLGEND) THEN
      WRITE(0,'(//,1X,A31,I2)') '..... Reading data from Boring ',NBOR

CRD---Read data on soil deposits at this boring
      CALL GETSOILS(NUMDEP, SDEPTP, SGAMMA, SD50, SCU, SPF, SPC,
     +              CFM, CFK, SPLIQ)

CRD---Read SPT data from this boring
      CALL GETNSPTS(NUMSPT, DEPTHN, SPTN, D50, CU, PF, PC, PLIQ, CFM,
     +          ER, CB, CS, NUMDEP, SDEPTP, SD50, SCU, SPF, SPC, SPLIQ)

CRD---Read specified liquefied layers
      CALL GETSPLIQ(NUMSLIQ, SPLTOP, SPLBOT, CFM)

CRD---Read output options
      CALL GETOUTOP(OUTSPT, OUTCPF, OUTEPL)

CAN---Begin analysis of this boring
      WRITE(0,*) '..... Analyzing ....'

CAN---Perform analysis at Nspt depths----------------------------------
      IF (OUTSPT) THEN
      WRITE(2,*) ' '
      WRITE(2,*) '---Results of analysis for recorded Nspt values-------
     +---------------'
      WRITE(2,*) '    Depth     N1,60    Sig''Vo       CRR        CSR
     + FSliq     Liq??'
      ENDIF

      DO 100 N=1, NUMSPT
CAN---Compute total and effective vertical stresses
        CALL XVSTRS(DEPTHN(N), SIGVT, SIGVE, DWT,NUMDEP,SDEPTP,SGAMMA)

CAN---Compute and store a corrected N160 value
        CALL XSPTN160(SPTN160(N), SPTN(N), DEPTHN(N), SIGVE, ER, CB, CS)

CAN---Compute clean-sand equivalent N1,60 value based on percent fines
        CALL XCSN160(CSN160(N), SPTN160(N), PF(N))

CAN---Compute CRR for clean-sand equivalent N160
        CALL XCRR(CRR(N), CSN160(N), EQM, PLIQ(N), DEPTHN(N), DWT)

CAN---Compute CSR induced by earthquake
        CALL XCSR(CSR, DEPTHN(N), AMAX, SIGVT, SIGVE)

CAN---Compute factor of safety against liquefaction
        CALL XFSLIQ(FSLIQ(N), CRR(N), CSR)

CAN---Set output marker for liqueified soil
        LIQBAN = ' -n- '
        IF( FSLIQ(N) .GE. 0.0 .AND. FSLIQ(N) .LE. 1.0) LIQBAN = '*LIQ*'

CAN---Output analysis for the depth of this recorded Nspt
        IF(OUTSPT) WRITE(2,200) DEPTHN(N), SPTN160(N), SIGVE,
     +                          CRR(N), CSR, FSLIQ(N), LIQBAN
 100  CONTINUE
      IF(OUTSPT) WRITE(2,*) '  < Bottom >'
 200  FORMAT(5X,F5.2,5X,F5.1,5X,F5.1,3X,F7.3,5X,F5.3,3X,F7.3,5X,A5)

CIN---Compute full profile of CRR & CSR at DINC intervals  -------
      IF (OUTCPF) THEN
      WRITE(2,*) ' '
      WRITE(2,*) '---Interpolated, continuous profile-------------------
     +---------------'
      WRITE(2,*) '    Depth     CRR        CSR       FSliq     Liq??'
      WRITE(2,400) 0.0, -1.0, 0.0, -1.0, ' -n- '
      ENDIF

      NUMINT = NINT( DEPTHN(NUMSPT)/DINC ) + 1
      DEPTH = 0.0
      FLIQS = .FALSE.
      NUMPLIQ = 0
      ISD = 2
      DO 300 K=1, NUMINT

CIN---Compute interpolation depth, round off to even DINC increments
        DEPTH = DEPTH + DINC
        DEPTH = NINT(DEPTH/DINC) * DINC

CIN---Interpolate CRR computed previously at locations of SPTs
        CALL INTCRR(CRRINT(K), DEPTH, NUMDEP, SDEPTP, SPLIQ, DWT,
     +               NUMSPT, DEPTHN, CSN160, CRR)

CIN---Compute total and effective vertical stresses
        CALL XVSTRS(DEPTH, SIGVT, SIGVE, DWT, NUMDEP, SDEPTP, SGAMMA)

CIN---Compute CSR at corresponding depths
        CALL XCSR(CSRINT(K), DEPTH, AMAX, SIGVT, SIGVE)

CIN---Compute factor of safety
        CALL XFSLIQ(FS, CRRINT(K), CSRINT(K))

CIN---Keep track of intervals of liquefied soil
        IF (.NOT. FLIQS .AND. (FS .GE. 0.0 .AND. FS .LE. 1.0)) THEN
CIN-----found top of liquefied soil
            NUMPLIQ = NUMPLIQ + 1
            BNDLIQS(NUMPLIQ,1) = DEPTH
            FLIQS = .TRUE.
        ELSEIF (FLIQS .AND. (FS .LT. 0.0 .OR. FS .GT. 1.0)) THEN
CIN-----found bottom of liquefied soil
            BNDLIQS(NUMPLIQ,2) = DEPTH
            FLIQS = .FALSE.
        ENDIF

        IF (OUTCPF) THEN
CIN---Set output marker for liquefied soil
          LIQBAN = ' -n- '
          IF( FS .GE. 0.0 .AND. FS .LE. 1.0) LIQBAN = '*LIQ*'

CIN---Write out marker for deposit interface
          IF((ISD.LE.NUMDEP) .AND. (DEPTH .GE. SDEPTP(ISD))) THEN
           WRITE(2,*)'    ::::::::::::::::::::::::::::::::::::::::::::'
           ISD = ISD + 1
          ENDIF

CIN---Write out interpolated profile
          WRITE(2,400) DEPTH, CRRINT(K), CSRINT(K), FS, LIQBAN
        ENDIF

 300  CONTINUE
      IF(FLIQS) BNDLIQS(NUMPLIQ,2) = DEPTH
      IF(OUTCPF) WRITE(2,*) '  < Bottom >'
 400  FORMAT(5X,F5.2,3X,F7.3,5X,F5.3,3X,F7.3,5X,A5)

CIN---Write out predicted and specified liquefied layers
      IF(OUTCPF .OR. OUTEPL) THEN
      NLAY = MIN0(NUMPLIQ,NUMSLIQ)
      WRITE(2,*) '---Liquefied sublayers-------PREDICTED by CODE----SPEC
     +IFIED by USER--'
      WRITE(2,*) '         Liquefied sublayer:    Top    Bottom        T
     +op    Bottom'
      DO 500 K=1,NLAY
 500  WRITE(2,508) K,BNDLIQS(K,1),BNDLIQS(K,2),SPLTOP(K),SPLBOT(K)
      IF (NUMPLIQ .GT. NLAY) THEN
        DO 504 L=NLAY+1, NUMPLIQ
 504    WRITE(2,508) L,BNDLIQS(L,1),BNDLIQS(L,2)
      ELSEIF (NUMSLIQ .GT. NLAY) THEN
        DO 506 L=NLAY+1, NUMSLIQ
 506    WRITE(2,510) L, SPLTOP(L), SPLBOT(L)
      ENDIF
 508  FORMAT(26X,I2,3X,F6.2,3X,F6.2,6X,F6.2,3X,F6.2)
 510  FORMAT(26X,I2,24X,F6.2,3X,F6.2)
      ENDIF

CEP---Compute EPOLLS parameters ---------------------------------------
      IF (OUTEPL) THEN
      WRITE(2,*) ' '
      WRITE(2,*) '===Computed EPOLLS geotechnical parameters============
     +==============='
      WRITE(2,*) ' '
      WRITE(2,*) '                                PREDICTED by CODE    SPEC
     +IFIED by USER  '
      WRITE(2,*) '                                =================    ====
     +============  '

CEP---Compute liquefied thickness parameters
      IF(NUMPLIQ .GT. 0) THEN
         DO 550 K=1, NUMPLIQ
 550     THKLIQ(1,K) = BNDLIQS(K,2) - BNDLIQS(K,1)
```

```
        ZTOPLIQ(1) = BNDLIQS(1,1)
        ZBOTLIQ(1) = BNDLIQS(NUMPLIQ,2)
      ELSE
        ZTOPLIQ(1) = 0.0
        ZBOTLIQ(1) = 0.0
      ENDIF
      IF(NUMSLIQ .GT. 0) THEN
         DO 600 K=1, NUMSLIQ
 600     THKLIQ(2,K) = SPLBOT(K) - SPLTOP(K)
        ZTOPLIQ(2) = SPLTOP(1)
        ZBOTLIQ(2) = SPLBOT(NUMSLIQ)
      ELSE
        ZTOPLIQ(2) = 0.0
        ZBOTLIQ(2) = 0.0
      ENDIF

CEP---Compute index of liquefied thickness
      CALL XINDXLIQ(RINDXLIQ, CRRINT, CSRINT, NUMINT, DINC)

CEP---Compute average parameters in liq/nonliq soil
      CALL XAVGPAR (AVFSLIQ, SPTLIQ, D50LIQ, CULIQ, CLAYLIQ,
     +              FINELIQ, SPTCAP, FINECAP, FINEBOT,
     +              ZTOPLIQ, ZBOTLIQ, SDEPTP, NUMDEP, NUMSPT,
     +              DEPTHN, FSLIQ, SPTN160, D50, CU, PF, PC,
     +              NUMSLIQ, SPLTOP, SPLBOT)

CEP---Compute minimum shear resistance parameters
      CALL XMINPAR (RMNFS, SPTMNFS, ZMNFS, RMNSPT, ZMNSPT,
     +              NUMSPT, DEPTHN, SPTN160, FSLIQ, PLIQ, DWT)

CEP---Write out EPOLLS liquefied thickness parameters
      WRITE(2,*) '         Liquefied sublayer:     Thick_Liq           T
     +hick_Liq '

        STHKLIQ(1) = 0.0
        STHKLIQ(2) = 0.0
        DO 610 IT=1,NLAY
          STHKLIQ(1) = STHKLIQ(1) + THKLIQ(1,IT)
          STHKLIQ(2) = STHKLIQ(2) + THKLIQ(2,IT)
 610      WRITE(2,616) IT,THKLIQ(1,IT),THKLIQ(2,IT)
        IF(NUMPLIQ .GT. NLAY) THEN
          DO 612 JT=NLAY+1, NUMPLIQ
          STHKLIQ(1) = STHKLIQ(1) + THKLIQ(1,JT)
 612      WRITE(2,616) JT,THKLIQ(1,JT)
        ELSEIF(NUMSLIQ .GT. NLAY) THEN
          DO 614 JT=NLAY+1, NUMSLIQ
          STHKLIQ(2) = STHKLIQ(2) + THKLIQ(2,JT)
 614      WRITE(2,618) JT,THKLIQ(2,JT)
        ENDIF
 616    FORMAT(26X,I2,8X,F4.1,16X,F4.1)
 618    FORMAT(26X,I2,28X,F4.1)

      WRITE(2,620) STHKLIQ(1), STHKLIQ(2)
 620  FORMAT(/,4X,'Sum liquefied thickness =',7X,F4.1,16X,F4.1)
      WRITE(2,625) (ZBOTLIQ(1)-ZTOPLIQ(1)), (ZBOTLIQ(2)-ZTOPLIQ(2))
 625  FORMAT(2X,'Gross liquefied thickness =',7X,F4.1,16X,F4.1,/)

CEP---Write out other EPOLLS parameters
      IF (NUMSLIQ .EQ. 0) THEN
      WRITE(2,681) DWT
      WRITE(2,682) ZTOPLIQ(1)
      WRITE(2,683) ZBOTLIQ(1)
      WRITE(2,684) RINDXLIQ
      WRITE(2,685) AVFSLIQ(1)
      WRITE(2,686) SPTLIQ(1)
      WRITE(2,687) RMNFS
      WRITE(2,688) SPTMNFS
      WRITE(2,689) ZMNFS
      WRITE(2,690) RMNSPT
      WRITE(2,691) ZMNSPT
      WRITE(2,692) D50LIQ(1)
      WRITE(2,693) CULIQ(1)
      WRITE(2,694) CLAYLIQ(1)
      WRITE(2,695) FINELIQ(1)
      WRITE(2,696) SPTCAP(1)
      WRITE(2,697) FINECAP(1)
      WRITE(2,698) FINEBOT(1)
      ELSE
      WRITE(2,681) DWT       ,DWT
      WRITE(2,682) ZTOPLIQ(1),ZTOPLIQ(2)
      WRITE(2,683) ZBOTLIQ(1),ZBOTLIQ(2)
      WRITE(2,684) RINDXLIQ  ,RINDXLIQ
      WRITE(2,685) AVFSLIQ(1),AVFSLIQ(2)
      WRITE(2,686) SPTLIQ(1) ,SPTLIQ(2)
      WRITE(2,687) RMNFS     ,RMNFS
      WRITE(2,688) SPTMNFS   ,SPTMNFS
      WRITE(2,689) ZMNFS     ,ZMNFS
      WRITE(2,690) RMNSPT    ,RMNSPT
      WRITE(2,691) ZMNSPT    ,ZMNSPT
      WRITE(2,692) D50LIQ(1) ,D50LIQ(2)
      WRITE(2,693) CULIQ(1)  ,CULIQ(2)
      WRITE(2,694) CLAYLIQ(1),CLAYLIQ(2)
      WRITE(2,695) FINELIQ(1),FINELIQ(2)
      WRITE(2,696) SPTCAP(1) ,SPTCAP(2)
      WRITE(2,697) FINECAP(1),FINECAP(2)
      WRITE(2,698) FINEBOT(1),FINEBOT(2)
      ENDIF
      WRITE(2,*) '=====================================================
     +==============='
      ENDIF

 681  FORMAT(18X,'Z_GrWater =',7X,F5.2,15X,F5.2)
 682  FORMAT(18X,'Z_TopLiq  =',7X,F4.1,16X,F4.1)
 683  FORMAT(18X,'Z_BotLiq  =',7X,F4.1,16X,F4.1)
 684  FORMAT(18X,'Index_Liq =',8X,F5.3,15X,F5.3)
 685  FORMAT(18X,'FS_Liq    =',8X,F5.3,15X,F5.3)
 686  FORMAT(18X,'N160_Liq  =',7X,F5.2,15X,F5.2)
 687  FORMAT(18X,'FS_Min    =',8X,F5.3,15X,F5.3)
 688  FORMAT(18X,'N160_MnFS =',7X,F5.2,15X,F5.2)
 689  FORMAT(18X,'Z_MnFS    =',7X,F5.2,15X,F5.2)
 690  FORMAT(18X,'N160_Min  =',7X,F5.2,15X,F5.2)
 691  FORMAT(18X,'Z_MnN160  =',7X,F5.2,15X,F5.2)
 692  FORMAT(18X,'D50_Liq   =',7X,F5.2,15X,F5.2)
 693  FORMAT(18X,'Cu_Liq    =',7X,F5.2,15X,F5.2)
 694  FORMAT(18X,'Clay_Liq  =',7X,F5.2,15X,F5.2)
 695  FORMAT(18X,'Fine_Liq  =',7X,F5.2,15X,F5.2)
 696  FORMAT(18X,'N160_Cap  =',7X,F5.2,15X,F5.2)
 697  FORMAT(18X,'Fine_Cap  =',7X,F5.2,15X,F5.2)
 698  FORMAT(18X,'Fine_Base =',7X,F5.2,15X,F5.2)

CAN---Analysis of this boring complete, read next boring
      GOTO 10

CR--No more borings to process (FLGEND = FALSE at line 10)
      ENDIF
      CLOSE(1)
C--Print end banner and close output file
      CALL BANNER(2,2)
      CLOSE(2)
C--Display end banner and terminate execution
      CALL BANNER(0,2)
      END

C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C++  SUBROUTINE OPENFILES                                           ++
C++     Prompts for filename of input file (fname)                  ++
C++     Opens input file  (fname.DAT)                               ++
C++     Opens output file (fname.LIQ)                               ++
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      SUBROUTINE OPENFILES
      IMPLICIT REAL*4 (A-H,O-Z), INTEGER (I-N)
      CHARACTER FNAME*40

      WRITE(0,*) ' '
      WRITE(0,*) '..... Enter filename for boring logs'
      WRITE(0,*) '.....    ( .DAT filetype assumed ) :'

C--Get filename, process, and open input file
      READ(0,'(A40)') FNAME
          NFT=0
          DO 5 I=1, 40
   5      IF (FNAME(I:I) .EQ. ' ' .AND. NFT .EQ. 0) NFT=I
          FNAME(NFT:NFT+3) = '.DAT'
      OPEN(1,FILE=FNAME,STATUS='OLD',ERR=10)

C--Re-process filename and open output file
      FNAME(NFT:NFT+3) = '.LIQ'
      OPEN(2,FILE=FNAME,STATUS='UNKNOWN')
      WRITE(2,7) FNAME
   7  FORMAT(' -- Output File Name: ',A40)

      RETURN
  10  CALL BANNER(0,3)
      WRITE(0,*) '!!!!!!           Could not open input file '
      CALL BANNER(0,3)
      STOP
      END

C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C++  SUBROUTINE BANNER                                              ++
C++     Displays and prints banners indicating:                     ++
C++          MESSAGE = 1   program name and version                 ++
C++          MESSAGE = 2   normal exit from program                 ++
C++          MESSAGE = 3   error flag                               ++
C++     To destination:                                             ++
C++          IUNIT = 0     screen                                   ++
C++          IUNIT = 2     output file                              ++
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      SUBROUTINE BANNER(IUNIT,MESSAGE)
      IMPLICIT INTEGER (I-N)
      CHARACTER*72 BEPOLL, BTITBL, BTITLE, BDESCR, BPREND, BERROR

C--Define character strings for banners
      BEPOLL=' ((((((((((((((((((((((((((  EPOLIQAN  ))))))))))))))))
     +))))))))))))'
      BTITBL=' (
     +          )'
      BTITLE=' (                       EPOLIQAN Version 2.0
     +          )'
      BDESCR=' (                  EPOLLS Liquefaction Analysis Code
     +          )'
      BPREND=' (.................. Execution complete - Normal exit .....
     +..........)'
      BERROR=' !!!!!!!!!!!!!!!!!!!!!!!!!!!!  ERROR  !!!!!!!!!!!!!!!!!!
     +!!!!!!!!!!!!'

C--Write program banner
      IF (MESSAGE .EQ. 1) THEN
        WRITE(IUNIT,'(A72)') BEPOLL
        WRITE(IUNIT,'(A72)') BTITBL
        WRITE(IUNIT,'(A72)') BTITLE
        WRITE(IUNIT,'(A72)') BDESCR
        WRITE(IUNIT,'(A72)') BTITBL
        WRITE(IUNIT,'(A72)') BEPOLL

C--Write end of program banner
      ELSEIF (MESSAGE .EQ. 2) THEN
        WRITE(IUNIT,'(A72)')  BEPOLL
        WRITE(IUNIT,'(A72)')  BPREND
        WRITE(IUNIT,'(A72)')  BEPOLL

C--Write error banner
      ELSEIF (MESSAGE .EQ. 3) THEN
        WRITE(IUNIT,'(/,A72,/)') BERROR

      ENDIF
      RETURN
      END

C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C++  SUBROUTINE GETTITLE                                            ++
C++     Reads input data and echoes to output file:                 ++
C++       BORING = Boring title or ID                               ++
C++       EQM    = Earthquake moment magnitude                      ++
C++       AMAX   = Peak horizontal surface acceleration             ++
C++       DWT    = Depth to hydrostatic water level (water table)   ++
C++       CFG    = Conversion factor from acceleration units to g's ++
C++       CFM    = Conversion factor from length units to meters    ++
C++       CFK    = Conversion factor from denisty units to kN/m3    ++
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      SUBROUTINE GETTITLE(FLGEND, EQM, AMAX, DWT, CFM, CFK)
      IMPLICIT REAL*4 (A-H,O-Z), INTEGER (I-N)
      CHARACTER BORGID*72, SKIP*1
      LOGICAL FLGEND,FLBOT

      FLBOT = .FALSE.

C--Read identification for boring log
      READ(1,'(A72)',END=100) BORGID
      WRITE(2,'(/,1X,A72,/)') BORGID
      WRITE(2,*) '      (values in meters, kN, and kPa unless otherwise s
     +tated)'
```

```
      WRITE(2,*) '+++++++++++++++++++++++++++++++++++++++++++++++++++++
     ++++++++++++++++'

C--Read site parameters and unit conversion factors
      READ(1,'(A1)',ERR=10,IOSTAT=IOERR,END=100) SKIP
      READ(1,*,ERR=10,IOSTAT=IOERR,END=100)EQM, AMAX, DWT, CFG, CFM, CFK

C--Perform unit conversions
      AMAX = AMAX * CFG
      DWT  = DWT  * CFM

C--Echo values to output file
      WRITE(2,*) '    EQ Mw   Amax(g)    Depth WT       CFg       CFm
     +  CFkN/m3'
      WRITE(2,200) EQM, AMAX, DWT, CFG, CFM, CFK
      RETURN

  10  CALL PROCREADERR(IOERR,FLBOT)
      RETURN

 100  FLGEND = .FALSE.
      RETURN

 200  FORMAT(6X,F4.2,2X,E8.3,5X,F5.2,6X,F7.4,3X,F7.4,3X,F7.4,/)
      END

C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C++  SUBROUTINE GETSOILS                                             ++
C++     Reads input data and echoes to output file:                  ++
C++         NUMDEP = Number of soil deposits given                   ++
C++         SDEPTP = Depth to top of deposit                         ++
C++         SGAMMA = Unit weight of deposit                          ++
C++         SD50   = avg. D50 size for deposit (in mm)               ++
C++         SCU    = avg. Uniformity coeffcient                      ++
C++         SPF    = avg. Percent fines (%)                          ++
C++         SPC    = avg. Percent clay  (%)                          ++
C++         SPLIQ  = TRUE if this deposit is potentially liquefiable ++
C++     For each, array element IS corresponds to deposit #IS        ++
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      SUBROUTINE GETSOILS(NUMDEP, SDEPTP, SGAMMA, SD50, SCU, SPF, SPC,
     +                    CFM, CFK, SPLIQ)
      IMPLICIT REAL*4 (A-H,O-Z), INTEGER (I-N)
      COMMON /MEMORY/ MDEP, MSPT, MINT
      LOGICAL FLBOT, FLZERO, FLDUPL, SPLIQ, XF
      CHARACTER SKIP*1
      DIMENSION SDEPTP(MDEP), SGAMMA(MDEP), SD50(MDEP), SCU(MDEP),
     +          SPF(MDEP), SPC(MDEP), SPLIQ(MDEP)

      FLBOT  = .TRUE.
      FLZERO = .FALSE.
      FLDUPL = .TRUE.
      NUMDEP = 0

      WRITE(2,*) '...Site soil strata and properties...................
     +..............'

C--Read parameters for each soil deposit at this boring
      READ(1,'(A1)',ERR=10,IOSTAT=IOERR) SKIP

C--Read until error occurs, process error to see if at 'bottom'
   5  NUMDEP = NUMDEP + 1
      READ(1,*,ERR=10,IOSTAT=IOERR) SDEPTP(NUMDEP), SGAMMA(NUMDEP),
     +                      SD50(NUMDEP), SCU(NUMDEP), SPF(NUMDEP),
     +                      SPC(NUMDEP), SPLIQ(NUMDEP)

C--Perform unit conversions
      SGAMMA(NUMDEP) = SGAMMA(NUMDEP) * CFK
      SDEPTP(NUMDEP) = SDEPTP(NUMDEP) * CFM

C--Reset unspecified parameters so will output '*****'
      IF(SD50(NUMDEP).LT. 0.0) SD50(NUMDEP) = -1000.
      IF(SCU(NUMDEP) .LT. 0.0) SCU(NUMDEP)  = -1000.
      IF(SPF(NUMDEP) .LT. 0.0) SPF(NUMDEP)  = -1000.
      IF(SPC(NUMDEP) .LT. 0.0) SPC(NUMDEP)  = -1000.

C--Check for a line with SDEPTP=0.0
      IF(SDEPTP(NUMDEP) .EQ. 0.0) FLZERO = .TRUE.
C--Check for duplicate specifications
      IF (NUMDEP .GT. 1) THEN
       DO 7 K=1, NUMDEP-1
   7   IF (SDEPTP(NUMDEP) .EQ. SDEPTP(K)) FLDUPL = .FALSE.
      ENDIF
      GOTO 5

  10  NUMDEP = NUMDEP - 1
C--If encountered 'bottom' will continue execution
      CALL PROCREADERR(IOERR,FLBOT)

C--Sort data into ascending depths using straight insertion
C--(Ref. Press et al. 1989)
      DO 100 J=2,NUMDEP
        X0 = SDEPTP(J)
        X1 = SGAMMA(J)
        X2 = SD50(J)
        X3 = SCU(J)
        X4 = SPF(J)
        X5 = SPC(J)
        XF = SPLIQ(J)
        DO 50 I=J-1,1,-1
          IF(SDEPTP(I) .LE. X0) GOTO 80
          SDEPTP(I+1) = SDEPTP(I)
          SGAMMA(I+1) = SGAMMA(I)
          SD50(I+1)   = SD50(I)
          SCU(I+1)    = SCU(I)
          SPF(I+1)    = SPF(I)
          SPC(I+1)    = SPC(I)
          SPLIQ(I+1)  = SPLIQ(I)
  50    CONTINUE
        I = 0
  80    SDEPTP(I+1) = X0
        SGAMMA(I+1) = X1
        SD50(I+1)   = X2
        SCU(I+1)    = X3
        SPF(I+1)    = X4
        SPC(I+1)    = X5
        SPLIQ(I+1)  = XF
 100  CONTINUE

C--Echo data to output file in correct order
      WRITE(2,*) '  Depth Top  Unit Wt   D50(mm)      Cu     % fines   %
     + clay  Pot.Liq?'

      DO 150 I=1, NUMDEP
 150  WRITE(2,200) SDEPTP(I), SGAMMA(I), SD50(I), SCU(I), SPF(I),
     +              SPC(I), SPLIQ(I)
      WRITE(2,*) '   < Bottom >'

C--Continue only if have properties for zero depth and no duplicates
      IF(FLZERO .AND. FLDUPL) THEN
        RETURN
      ELSE
        CALL BANNER(0,3)
        IF (.NOT. FLDUPL) THEN
         WRITE(0,*) '!!!!!!  Duplicate properties for one depth  !!!'
        ENDIF
        IF (.NOT. FLZERO) THEN
         WRITE(0,*) '!!!!!!  Must specify soil properties for a  !!!'
         WRITE(0,*) '!!!!!!   deposit that starts at zero depth  !!!'
        ENDIF
        CALL BANNER(0,3)
        STOP '..... Program terminated at error in properties ...'
      ENDIF

 200  FORMAT(5X,F5.2,4X,F6.2,4X,F6.2,4X,F6.2,5X,F5.2,5X,F5.2,5X,L1)
      END

C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C++  SUBROUTINE GETNSPTS                                             ++
C++     Reads input data and echoes to output file:                  ++
C++         ER     = SPT energy ratio to normalize to 60% standard   ++
C++         CB     = Correction factor for borehole diameter         ++
C++         CS     = Correction factor for nonstandard sampler       ++
C++         NUMSPT = Number of values recorded                       ++
C++         DEPTHN = Mid-depth of SPT measurement                    ++
C++         SPTN   = Measured, uncorrected Nspt value                ++
C++         D50    = D50 size (in mm) at SPT                         ++
C++         CU     = Uniformity coeffcient at SPT                    ++
C++         PF     = Percent fines (%) at SPT                        ++
C++         PC     = Percent clay  (%) at SPT                        ++
C++         PLIQ   = TRUE if in potentially liquefiable deposit      ++
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      SUBROUTINE GETNSPTS(NUMSPT, DEPTHN, SPTN, D50, CU, PF, PC, PLIQ,
     +     CFM, ER, CB, CS, NUMDEP, SDEPTP, SD50, SCU, SPF, SPC, SPLIQ)
      IMPLICIT REAL*4 (A-H,O-Z), INTEGER (I-N)
      COMMON /MEMORY/ MDEP, MSPT, MINT
      LOGICAL FLBOT, FLDUPL, SPLIQ, PLIQ
      CHARACTER SKIP*1
      DIMENSION SDEPTP(MDEP), SD50(MDEP), SCU(MDEP),
     +          SPF(MDEP), SPC(MDEP), SPLIQ(MDEP)
      DIMENSION DEPTHN(MSPT), SPTN(MSPT), D50(MSPT), CU(MSPT), PF(MSPT),
     +          PC(MSPT), PLIQ(MSPT)

      FLBOT  = .TRUE.
      FLDUPL = .TRUE.
      NUMSPT = 0

      WRITE(2,*) '...Measured SPT values matched with soil properties...
     +..............'

C--Read correction parameters for SPT values
      READ(1,'(A1)',ERR=10,IOSTAT=IOERR) SKIP
      READ(1,*,ERR=10,IOSTAT=IOERR) ER, CB, CS
      READ(1,'(A1)',ERR=10,IOSTAT=IOERR) SKIP

C--Echo SPT correction parameters to output file
      WRITE(2,1) ER
   1  FORMAT(' ....SPT hammer Energy Ratio (%)          = ',F5.1)
      WRITE(2,2) CB
   2  FORMAT(' ....Correction for borehole diameter    =   ',F4.2)
      WRITE(2,3) CS
   3  FORMAT(' ....Correction for non-standard sampler =   ',F4.2)

C--Read values measured in SPT
C--Read until error occurs, process error to see if at 'bottom'
   5  NUMSPT = NUMSPT + 1
      READ(1,*,ERR=10,IOSTAT=IOERR) DEPTHN(NUMSPT), SPTN(NUMSPT),
     +     D50(NUMSPT), CU(NUMSPT), PF(NUMSPT), PC(NUMSPT)

C--Perform unit conversions
      DEPTHN(NUMSPT) = DEPTHN(NUMSPT) * CFM
C--Check for duplicate specifications
      IF (NUMSPT .GT. 1) THEN
       DO 7 K=1, NUMSPT-1
   7   IF (DEPTHN(NUMSPT) .EQ. DEPTHN(K)) FLDUPL = .FALSE.
      ENDIF
      GOTO 5

  10  NUMSPT = NUMSPT - 1
C--If encountered 'bottom' will continue execution
      CALL PROCREADERR(IOERR,FLBOT)

C--Sort data into ascending depths using straight insertion
C--(Ref. Press et al. 1989)
      DO 100 J=2,NUMSPT
        X0 = DEPTHN(J)
        X1 = SPTN(J)
        X2 = D50(J)
        X3 = CU(J)
        X4 = PF(J)
        X5 = PC(J)
        DO 50 I=J-1,1,-1
          IF(DEPTHN(I) .LE. X0) GOTO 80
          DEPTHN(I+1) = DEPTHN(I)
          SPTN(I+1)   = SPTN(I)
          D50(I+1)    = D50(I)
          CU(I+1)     = CU(I)
          PF(I+1)     = PF(I)
          PC(I+1)     = PC(I)
  50    CONTINUE
        I = 0
  80    DEPTHN(I+1) = X0
        SPTN(I+1)   = X1
        D50(I+1)    = X2
        CU(I+1)     = X3
        PF(I+1)     = X4
        PC(I+1)     = X5
 100  CONTINUE

C--Echo data to output file in correct order
      WRITE(2,*) '    Depth      Nspt    D50(mm)      Cu     % fines   %
     + clay  Pot.Liq?'

      DO 150 I=1, NUMSPT
```

```
C--Look up appropriate soil properties at depths where not specified
C--Find location based on SDEPTP in ordered array
      LOC = 1
      DO 110 J=2,NUMDEP
 110   IF(SDEPTP(J) .LE. DEPTHN(I)) LOC = J
C--Replace unspecified values with deposit average
      IF(D50(I) .LT. 0.0) D50(I) = SD50(LOC)
      IF(CU(I)  .LT. 0.0)  CU(I)  = SCU(LOC)
      IF(PF(I)  .LT. 0.0)  PF(I)  = SPF(LOC)
      IF(PC(I)  .LT. 0.0)  PC(I)  = SPC(LOC)
      PLIQ(I) = SPLIQ(LOC)

 150  WRITE(2,200) DEPTHN(I),SPTN(I),D50(I),CU(I),PF(I),PC(I),PLIQ(I)
      WRITE(2,*) '  < Bottom >'

C--Check for specification of duplicate values
      IF (FLDUPL) RETURN
      CALL BANNER(0,3)
      WRITE(0,*) '!!!!!!  Duplicate SPT values for one depth  !!!'
      CALL BANNER(0,3)
      STOP '..... Program terminated at error in SPT values ...'

 200  FORMAT(5X,F5.2,5X,F5.1,4X,F6.2,4X,F6.2,5X,F5.2,5X,F5.2,5X,L1)
      END

C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C++  SUBROUTINE GETSPLIQ                                             ++
C++     Reads input data and echoes to output file:                  ++
C++      NUMSLIQ =  Number of liquefied layers specified             ++
C++      SPLTOP  =  Top of specified liquefied soil                  ++
C++      SPLBOT  =  Bottom of specified liquefied soil               ++
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      SUBROUTINE GETSPLIQ(NUMSLIQ, SPLTOP, SPLBOT, CFM)
      IMPLICIT REAL*4 (A-H,O-Z), INTEGER (I-N)
      COMMON /MEMORY/ MDEP, MSPT, MINT
      CHARACTER SKIP*1
      LOGICAL FLBOT, FLOVR
      DIMENSION SPLTOP(MDEP), SPLBOT(MDEP)

      FLOVR  = .FALSE.
      FLBOT  = .TRUE.
      NUMSLIQ = 0

C--Read depths to top and bottom of specified liquefied layers
C--Read until error occurs, process error to see if at 'bottom'
      READ(1,'(A1)',ERR=10,IOSTAT=IOERR) SKIP
   5  NUMSLIQ = NUMSLIQ + 1
      READ(1,*,ERR=10,IOSTAT=IOERR) SPLTOP(NUMSLIQ), SPLBOT(NUMSLIQ)

C--Perform unit conversions
      SPLTOP(NUMSLIQ) = SPLTOP(NUMSLIQ) * CFM
      SPLBOT(NUMSLIQ) = SPLBOT(NUMSLIQ) * CFM
      GOTO 5

  10  NUMSLIQ = NUMSLIQ - 1
C--If encountered 'bottom' will continue execution
      CALL PROCREADERR(IOERR,FLBOT)

C--Sort data into ascending TOP depths using straight insertion
C--(Ref. Press et al. 1989)
      DO 100 J=2,NUMSLIQ
        X0 = SPLTOP(J)
        X1 = SPLBOT(J)
        DO 50 I=J-1,1,-1
          IF(SPLTOP(I) .LE. X0) GOTO 80
          SPLTOP(I+1) = SPLTOP(I)
          SPLBOT(I+1) = SPLBOT(I)
  50    CONTINUE
        I = 0
  80    SPLTOP(I+1) = X0
        SPLBOT(I+1) = X1
 100  CONTINUE

C--Check for overlap of specified layers
      DO 150 K=2,NUMSLIQ
 150  IF (SPLBOT(K-1) .GT. SPLTOP(K)) FLOVR = .TRUE.

      IF (FLOVR) THEN
       CALL BANNER(0,3)
       WRITE(0,*) '!!!!!!  Specified liquefied layers overlap  !!!'
       CALL BANNER(0,3)
       STOP '..... Program terminated at error in read    ......'
      ENDIF
      RETURN
      END

C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C++  SUBROUTINE GETOUTOP                                             ++
C++     Reads options for data to be printed in the output file      ++
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      SUBROUTINE GETOUTOP(OUTSPT, OUTCPF, OUTEPL)
      LOGICAL OUTSPT, OUTCPF, OUTEPL, FLBOT
      CHARACTER SKIP*1

      FLBOT  = .FALSE.
      OUTSPT = .TRUE.
      OUTCPF = .TRUE.
      OUTEPL = .TRUE.

C--Read output parameters
      READ(1,'(A1)',ERR=10,IOSTAT=IOERR) SKIP
      READ(1,*,ERR=10,IOSTAT=IOERR) OUTSPT, OUTCPF, OUTEPL
      RETURN

C--Process read errors if encountered
  10  CALL PROCREADERR(IOERR,FLBOT)
      END

C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C++  SUBROUTINE PROCREADERR                                          ++
C++     Processes errors encountered during reading of input         ++
C++     files, based on value of IOERR.                              ++
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      SUBROUTINE PROCREADERR(IOERR,FLBOT)
      IMPLICIT REAL*4 (A-H,O-Z), INTEGER (I-N)
      CHARACTER BREAK*3
      LOGICAL FLBOT

C--If IOERR = 6103 "Invalid REAL", or IOERR = 6101 "Invalid INTEGER"
C--then check for specified delimiter "bot" or "BOT"
      IF (IOERR .EQ. 6101 .OR. IOERR .EQ. 6103) THEN
         BACKSPACE (1)
         READ(1,'(A3)') BREAK
           IF (BREAK .EQ. 'BOT' .OR. BREAK .EQ. 'bot') THEN
C--return only if "bot" is valid at this point (FLBOT is TRUE)
           IF (.NOT. FLBOT) GOTO 50
           RETURN
           ENDIF
      ENDIF
C--Fatal read error
50    CALL BANNER(0,3)
      WRITE(0,*) '!!!!!!         Error reading input file        !!!!!!'
      IF (IOERR .EQ. 6405)
     +WRITE(0,*) '!!!!!!         ...End-of-File encountered      !!!!!!'
      IF (IOERR .EQ. 6101 .OR. IOERR .EQ. 6103)
     +WRITE(0,*) '!!!!!!         ...check "BOTTOM" delimiter     !!!!!!'
      IF (IOERR .EQ. 6507)
     +WRITE(0,*) '!!!!!!  ...T or F required for logical variable  !!!!'
      CALL BANNER(0,3)

      STOP '..... Program terminated at read error .....'
      END

C=====================================================================
C==  SUBROUTINE XVSTRS                                              ==
C==     Computes total and effective overburden stresses            ==
C==     at a specified DEPTH:                                       ==
C==         SIGVT = Total vertical stress (kPa)                     ==
C==         SIGVE = Effective vertical stress (kPa)                 ==
C==   (-) DWT indicates standing water above ground surface         ==
C==       which is not included in SIGVT but is in SIGVE            ==
C=====================================================================
      SUBROUTINE XVSTRS(DEPTH, SIGVT, SIGVE, DWT,
     +                  NUMDEP, SDEPTP, SGAMMA)
      IMPLICIT REAL*4 (A-H,O-Z), INTEGER (I-N)
      COMMON /MEMORY/ MDEP, MSPT, MINT
      DIMENSION SDEPTP(MDEP), SGAMMA(MDEP)

      SIGVT = 0.0
      LOC = 1

C--Compute total stress contributed by complete overburden layers
      DO 50 I=2, NUMDEP
        IF (SDEPTP(I) .LE. DEPTH) THEN
          LOC = I
          SIGVT = SIGVT + (SDEPTP(I)-SDEPTP(I-1))*SGAMMA(I-1)
        ENDIF
  50  CONTINUE
C--Add soil within same deposit above depth of interest
      SIGVT = SIGVT + (DEPTH-SDEPTP(LOC))*SGAMMA(LOC)

C--Compute pore water pressure (assumes hydrostatic conditions)
C--Unit wieght of water = 9.80665 kN/m3, DWT in meters
      U = (DEPTH-DWT) * 9.80665
      IF (U .LT. 0.0) U = 0.0

C--Compute vertical effective stress and return
      SIGVE = SIGVT - U
      IF (DWT .LT. 0.0) SIGVE = SIGVE - (DWT*9.80665)
      RETURN
      END

C=====================================================================
C==  SUBROUTINE XSPTN160                                            ==
C==     Computes a corrected value (N1,60) for recorded Nspt        ==
C==         SPTN160 = corrected Nspt value                          ==
C==         SPTN    = recorded Nspt value                           ==
C==         DEPTHN  = depth of measured value                       ==
C==         SIGVE   = effective vertical stress (kPa)               ==
C==     Correction factors are applied for:                         ==
C==         CN  = effective vertical stress of 1 tsf                ==
C==         CE  = 60% energy delivery to drill rods                 ==
C==         CB  = bore hole > 4.5 inch in diameter                  ==
C==         CR  = energy loss in short drill rods                   ==
C==         CS  = non-standard sampler                              ==
C=====================================================================
      SUBROUTINE XSPTN160(SPTN160, SPTN, DEPTHN, SIGVE, ER, CB, CS)
      IMPLICIT REAL*4 (A-H,O-Z), INTEGER (I-N)

C--Compute correction factor for 1 atm overburden pressure
C--Set maximum correction to CN=2.0 for very shallow depths
      CN = 2.0
      SIGMIN = 101.325 / CN**2.0
      IF (SIGVE .GT. SIGMIN) CN = SQRT(101.325/SIGVE)

C--Compute correction factor for 60% energy standard
      CE = ER / 60.0

C--Compute correction factor for energy loss in short drill rods
      IF (DEPTHN .GE. 9.0) THEN
          CR = 1.0
      ELSEIF( DEPTHN .LE. 3.0) THEN
          CR = 0.75
      ELSE
          CR = 5.0/8.0 + DEPTHN/24.0
      ENDIF

C--Compute corrected Nspt value
      SPTN160 = SPTN * CN * CE * CB * CR * CS

      RETURN
      END

C=====================================================================
C==  SUBROUTINE XCSN160                                             ==
C==     Computes a clean-sand equivalent N160 value (CSN160)        ==
C==     based on the percent fines (PF)                             ==
C=====================================================================
      SUBROUTINE XCSN160(CSN160, SPTN160, PF)
      IMPLICIT REAL*4 (A-H,O-Z), INTEGER (I-N)

C--No correction for less than 5% fines
C--Also, when % fines is not given, will make no correction
      IF (PF .LE. 5.0) THEN
        DELN1 = 0.0

C--Use linear approximation for 5 to 35% fines
      ELSEIF (PF .LT. 35.0) THEN
        DELN1 = (PF - 5.0) * 7.0 / 30.0

C--Correction is 7.0 if more than 35% fines
      ELSE
```

```
        DELN1 = 7.0
      ENDIF

C--Add correction to get clean-sand equivalent value
      CSN160 = SPTN160 + DELN1

      RETURN
      END

C======================================================================
C==  SUBROUTINE XCRR                                                 ==
C==     Computes CRR liquefaction resistance of soil                 ==
C==         CSN160   = clean-sand equivalent, corrected Nspt value   ==
C==         EQM      = earthquake moment magnitude                   ==
C==         PLIQ     = TRUE if in potentially liquefiable deposit    ==
C==         DEPTH    = depth of interest (meters)                    ==
C==         DWT      = depth to groundwater table                    ==
C==     Method used:                                                 ==
C==     Consensus recommendation of 1996 NCEER Liquefaction Workshop ==
C======================================================================
      SUBROUTINE XCRR(CRR, CSN160, EQM, PLIQ, DEPTH, DWT)
      IMPLICIT REAL*4 (A-H,O-Z), INTEGER (I-N)
      REAL*4 MSF
      LOGICAL PLIQ

C--If soil specified as not liquefiable, set CRR = -1
      IF (.NOT. PLIQ) THEN
          CRR = -1.0
          RETURN
C--If soil unsaturated, set CRR = -1
      ELSEIF (DEPTH .LE. DWT) THEN
          CRR = -1.0
          RETURN
C--If N160 > 30, soil is not liquefiable
      ELSEIF (CSN160 .GT. 30.0) THEN
          CRR = -1.0
          RETURN
      ENDIF

C--Compute magnitude scaling factor
      IF (EQM .LT. 7.0) THEN
          MSF = (10.0**3.00) / (EQM**3.46)
      ELSE
          MSF = (10.0**2.24) / (EQM**2.56)
      ENDIF

C--Compute CRR from M=7.5 base curve
      TM1 = 95.0 / (34.0 - CSN160)
      TM2 = TM1 + (CSN160/1.3) - 0.5
      CRR = MSF * TM2/100

      RETURN
      END

C======================================================================
C==  SUBROUTINE XCSR                                                 ==
C==     Computes CSR induced by earthquake                           ==
C==         DEPTH = depth of interest (meters)                       ==
C==         AMAX  = peak surface acceleration (g's)                  ==
C==         SIGVT = Total vertical stress                            ==
C==         SIGVE = Effective vertical stress                        ==
C======================================================================
      SUBROUTINE XCSR(CSR, DEPTH, AMAX, SIGVT, SIGVE)
      IMPLICIT REAL*4 (A-H,O-Z), INTEGER (I-N)

      CSR = 0.0

C--Compute stress reduction factor using quartic equation
      TM1 = (DEPTH**4.0) - (42.0*DEPTH**3.0) + (105.0*DEPTH**2.0)
      RD = 1.0 + 1.6*(TM1 - 4200.0*DEPTH)/(10**6.0)
      IF (DEPTH .GT. 30.0) RD = 0.43

C--Compute CSR
      IF (SIGVT .LE. 0.0) RETURN
      CSR = 0.65 * AMAX * (SIGVT/SIGVE) * RD

      RETURN
      END

C======================================================================
C==  SUBROUTINE XFSLIQ                                               ==
C==     Computes Factor of Safety against liquefaction (FSLIQ)       ==
C==         CRR = CSR required to induce liquefaction                ==
C==         CSR = CSR induced by the earthquake                      ==
C==   Note: set FSLIQ to -1 if soil is identified as nonliquefiable  ==
C======================================================================
      SUBROUTINE XFSLIQ(FSLIQ, CRR, CSR)
      IMPLICIT REAL*4 (A-H,O-Z), INTEGER (I-N)

C--If soil unsaturated or otherwise nonliquefiable, set FSLIQ = -1
      IF( CRR .LT. 0.0 ) THEN
        FSLIQ = -1.0

C--If CSR < or = 0.0, set FSLIQ = -1
      ELSEIF ( CSR .LE. 0.0 ) THEN
        FSLIQ = -1.0

C--Compute factor of safety
      ELSE
        FSLIQ = CRR / CSR
      ENDIF

      RETURN
      END

C======================================================================
C==  SUBROUTINE INTCRR                                               ==
C==     Interpolates CRR (computed at measured SPT locations).       ==
C==     Returns value of CRRINT for DEPTH.                           ==
C==     Does not extrapolate to interface boundaries between         ==
C==        soil deposits indicated in array SDEPTP                   ==
C======================================================================
      SUBROUTINE INTCRR(CRRINT, DEPTH, NUMDEP, SDEPTP, SPLIQ, DWT,
     +                     NUMSPT, DEPTHN, CSN160, CRR)
      IMPLICIT REAL*4 (A-H,O-Z), INTEGER (I-N)
      COMMON /MEMORY/ MDEP, MSPT, MINT
      LOGICAL SPLIQ, FLEXT, FLAST
      DIMENSION SDEPTP(MDEP), SPLIQ(MDEP)
      DIMENSION DEPTHN(MSPT), CSN160(MSPT), CRR(MSPT)

C--Determine which soil deposit are in
```

```
      FLAST = .FALSE.
      LSD   = 1
      DO 50 J=2,NUMDEP
   50 IF(SDEPTP(J) .LE. DEPTH) LSD = J
      IF(LSD .EQ. NUMDEP) FLAST = .TRUE.

C--Search for last measured point that is shallower than DEPTH
      LNS = 0
      DO 100 I=1,NUMSPT
  100 IF(DEPTHN(I) .LE. DEPTH) LNS = I

C--If specified as not liquefiable or above water table, CRR=-1
      IF(.NOT. SPLIQ(LSD) .OR. DEPTH .LE. DWT ) THEN
          CRRINT = -1.0

C--Check to see if at very top or bottom of profile
      ELSEIF (LNS .EQ. 0) THEN
          CRRINT = CRR(1)
      ELSEIF (LNS .EQ. NUMSPT) THEN
          CRRINT = CRR(NUMSPT)
      ELSE

C--Set limits for interpolation
          D1 = DEPTHN(LNS)
          D2 = DEPTHN(LNS+1)
          C1 = CRR(LNS)
          C2 = CRR(LNS+1)

C--If no SPTs measured in this layer, assume not liquefiable
          IF((.NOT. FLAST) .AND.
     +       (D1.LT.SDEPTP(LSD)) .AND. (D2.GE.SDEPTP(LSD+1))) THEN
             CRRINT = -1.0

C--Check to see if near deposit interface
          ELSEIF ((D1 .LT. SDEPTP(LSD)) .OR. (D1 .LE. DWT)) THEN
              CRRINT = C2
          ELSEIF ((.NOT. FLAST) .AND. (D2 .GE. SDEPTP(LSD+1))) THEN
              CRRINT = C1
          ELSE

C--Interpolate between adjacent points measured in same deposit

C--First, check if go from CSN160 < 30 to > 30 in this interval
C--If so, find interpolated depth where CSN160 = 30, and reset
C--interpolation limits for CRRINT
              FLEXT = .FALSE.
              S1 = CSN160(LNS)
              S2 = CSN160(LNS+1)
              IF     ( S1.GT.30. .AND. S2.LE.30.) THEN
                D1 = D1 + (30.0-S1)*(D2-D1)/(S2-S1)
                C1 = 0.5
                IF (DEPTH .LT. D1) FLEXT = .TRUE.
              ELSEIF ( S1.LE.30. .AND. S2.GT.30.) THEN
                D2 = D1 + (30.0-S1)*(D2-D1)/(S2-S1)
                C2 = 0.5
                IF (DEPTH .GT. D2) FLEXT = .TRUE.
              ENDIF

C--Compute interpolated value
              CRRINT = C1 + (DEPTH-D1)*(C2-C1)/(D2-D1)
              IF (FLEXT) CRRINT = -1.0
          ENDIF
      ENDIF

      RETURN
      END

C======================================================================
C==  SUBROUTINE XINDXLIQ                                             ==
C==     Computes the Index of Liquefied Thickness  (RINDXLIQ)        ==
C==     Performs integration using FS midway between given points    ==
C==         CRRINT = interpolated profile of CRR                     ==
C==         CSRINT = interpolated profile of CSR                     ==
C==         NUMINT = number of points in interpolated arrays         ==
C==         DINC   = depth increment used in interpolations          ==
C======================================================================
      SUBROUTINE XINDXLIQ(RINDXLIQ, CRRINT, CSRINT, NUMINT, DINC)
      IMPLICIT REAL*4 (A-H,O-Z), INTEGER (I-N)
      COMMON /MEMORY/ MDEP, MSPT, MINT
      DIMENSION CRRINT(MINT), CSRINT(MINT)

C--Compute index in uppermost 20 m of profile, or full profile if <20 m
      N20M = NINT(20.0 / DINC)
      LAST = MIN0(N20M,NUMINT)

      DEPTH   = -0.5 * DINC
      F1      = 0.0
      TOPINDX = 0.0
      BOTINDX = 0.0
      DO 100 N=1, LAST

C--Compute Factor of Safety and F function at midpoint of this DINC
        CALL XFSLIQ(FS, CRRINT(N), CSRINT(N))
        F2 = 1.0 - FS
        IF( (FS.LT.0.0) .OR. (FS.GT.1.0)) F2 = 0.0
        F  = (F1 + F2) / 2.0
        F1 = F2

C--Compute interpolation depth corresponding to F
        DEPTH = DEPTH + DINC

C--Compute depth-weighting function, W
        W = 10.0 - 0.5*(DEPTH)

C--Accumulate numerator and denominator
        TOPINDX = TOPINDX + (F*W*DINC)
        BOTINDX = BOTINDX + (W*DINC)
  100 CONTINUE

C--Compute index of liquefied thickness
      RINDXLIQ = TOPINDX / BOTINDX
      RETURN
      END

C======================================================================
C==  SUBROUTINE XAVGPAR                                              ==
C==     Computes "average" parameters in liq/nonliq soil             ==
C==         AVFSLIQ = avg. FSliq in liquefied soil                   ==
C==         SPTLIQ  = avg. N1,60 in liquefied soil                   ==
C==         D50LIQ  = avg. D50 in liquefied soil                     ==
C==         CULIQ   = avg. Cu in liquefied soil                      ==
```

```
C==        CLAYLIQ = avg. clay content in liquefied soil           ==
C==        FINELIQ = avg. fines content in liquefied soil          ==
C==        SPTCAP  = avg. N1,60 measured above liquefied soil      ==
C==        FINECAP = avg. fines content above liquefied soil       ==
C==        FINEBOT = avg. fines content below liquefied soil       ==
C==        N(i)    = counters for computing averages               ==
C======================================================================
      SUBROUTINE XAVGPAR (AVFSLIQ, SPTLIQ, D50LIQ, CULIQ, CLAYLIQ,
     +                    FINELIQ, SPTCAP, FINECAP, FINEBOT,
     +                    ZTOPLIQ, ZBOTLIQ, SDEPTP, NUMDEP, NUMSPT,
     +                    DEPTHN, FSLIQ, SPTN160, D50, CU, PF, PC,
     +                    NUMSLIQ, SPLTOP, SPLBOT)
      IMPLICIT REAL*4 (A-H,O-Z), INTEGER (I-N)
      LOGICAL FLACUM
      COMMON /MEMORY/ MDEP, MSPT, MINT
      DIMENSION SDEPTP(MDEP), D50(MSPT), CU(MSPT), PF(MSPT), PC(MSPT)
      DIMENSION DEPTHN(MSPT), SPTN160(MSPT), FSLIQ(MSPT), N(8)
      DIMENSION ZTOPLIQ(2), ZBOTLIQ(2), AVFSLIQ(2), SPTLIQ(2),
     +          D50LIQ(2), CULIQ(2), CLAYLIQ(2), FINELIQ(2),
     +          SPTCAP(2), FINECAP(2), FINEBOT(2)
      DIMENSION SPLTOP(MDEP), SPLBOT(MDEP)

C---Perform caluculations twice for:
C   (M=1) predicted liq. layers and (M=2) specified liq. layers
      DO 500 M=1,2

      AVFSLIQ(M) = 0.0
      SPTLIQ(M)  = 0.0
      D50LIQ(M)  = 0.0
      CULIQ(M)   = 0.0
      CLAYLIQ(M) = 0.0
      FINELIQ(M) = 0.0
      SPTCAP(M)  = 0.0
      FINECAP(M) = 0.0
      FINEBOT(M) = 0.0

      DO 5 I=1,8
  5   N(I) = 0

C--Find deposit number for top and bottom of liquefied soil
      DO 10 J=1,NUMDEP
      IF(SDEPTP(J) .LE. ZTOPLIQ(M)) LTOP = J
 10   IF(SDEPTP(J) .LE. ZBOTLIQ(M)) LBOT = J

C--Accumulate values as scroll through measured data
      DO 100 K=1, NUMSPT

C--Averages at locations where liquefaction predicted or specified
C--but do not consider locations where parameter unspecified
      FLACUM = .FALSE.

C--For predicted layers (M=1), accumulate all with FS<1
        IF (M.EQ.1) THEN
           IF ((FSLIQ(K).GE.0.0) .AND. (FSLIQ(K).LE.1.0))
     +     FLACUM = .TRUE.
C--For specified layers (M=2), accumulate all within depth interval
        ELSEIF(M.EQ.2) THEN
            DO 15 L=1, NUMSLIQ
            IF((DEPTHN(K).GE.SPLTOP(L)) .AND. (DEPTHN(K).LE.SPLBOT(L)))
     +      FLACUM = .TRUE.
 15         CONTINUE
        ENDIF

      IF (FLACUM) THEN
         N(1) = N(1) + 1
         AVFSLIQ(M) = AVFSLIQ(M) + FSLIQ(K)
         SPTLIQ(M)  = SPTLIQ(M)  + SPTN160(K)

         IF (D50(K) .GE. 0.0) THEN
            N(2)      = N(2) + 1
            D50LIQ(M) = D50LIQ(M) + D50(K)
         ENDIF

         IF (CU(K) .GE. 0.0) THEN
            N(3)     = N(3) + 1
            CULIQ(M) = CULIQ(M) + CU(K)
         ENDIF

         IF (PF(K) .GE. 0.0) THEN
            N(4)       = N(4) + 1
            FINELIQ(M) = FINELIQ(M) + PF(K)
         ENDIF

         IF (PC(K) .GE. 0.0) THEN
            N(5)       = N(5) + 1
            CLAYLIQ(M) = CLAYLIQ(M) + PC(K)
         ENDIF
      ELSE
C--Find out what soil deposit are in
         DO 20 J=1,NUMDEP
  20     IF(SDEPTP(J) .LE. DEPTHN(K)) LS = J

C--Averages in soil above top of liquefied soil
         IF (DEPTHN(K) .LT. ZTOPLIQ(M)) THEN
            N(6)      = N(6) + 1
            SPTCAP(M) = SPTCAP(M) + SPTN160(K)

C--Consider fines content only in same layer and one just above
            IF((LS .GE. (LTOP-1)) .AND. (PF(K) .GE. 0.0)) THEN
             N(7)       = N(7) + 1
             FINECAP(M) = FINECAP(M) + PF(K)
            ENDIF

C--Averages in soil below bottom of liquefied soil
C--Consider fines content only in same layer and one just below
         ELSEIF ( (DEPTHN(K)   .GT. ZBOTLIQ(M) )  .AND.
     +            (   LS       .LE. (LBOT+1)) .AND.
     +            ( ZBOTLIQ(M) .GT.   0.0   )  .AND.
     +            (  PF(K)     .GE.   0.0   )) THEN
            N(8)       = N(8) + 1
            FINEBOT(M) = FINEBOT(M) + PF(K)
         ENDIF
      ENDIF
 100  CONTINUE

C--Compute averages from summations, if N=0, set so will output "****"
      IF (N(1) .EQ. 0) THEN
        AVFSLIQ(M) = -1000.
        SPTLIQ(M)  = -1000.
      ELSE
        AVFSLIQ(M) = AVFSLIQ(M) / N(1)
        SPTLIQ(M)  = SPTLIQ(M)  / N(1)
      ENDIF

      IF (N(2) .EQ. 0) THEN
        D50LIQ(M) = -1000.
      ELSE
        D50LIQ(M) = D50LIQ(M) / N(2)
      ENDIF

      IF (N(3) .EQ. 0) THEN
        CULIQ(M) = -1000.
      ELSE
        CULIQ(M) = CULIQ(M) / N(3)
      ENDIF

      IF (N(4) .EQ. 0) THEN
        FINELIQ(M) = -1000.
      ELSE
        FINELIQ(M) = FINELIQ(M) / N(4)
      ENDIF

      IF (N(5) .EQ. 0) THEN
        CLAYLIQ(M)  = -1000.
      ELSE
        CLAYLIQ(M)  = CLAYLIQ(M) / N(5)
      ENDIF

      IF(N(6) .EQ. 0) THEN
        SPTCAP(M)  = -1000.
      ELSE
        SPTCAP(M)  = SPTCAP(M) / N(6)
      ENDIF

      IF(N(7) .EQ. 0) THEN
        FINECAP(M) = -1000.
      ELSE
        FINECAP(M) = FINECAP(M) / N(7)
      ENDIF

      IF(N(8) .EQ. 0) THEN
        FINEBOT(M) = -1000.
      ELSE
        FINEBOT(M) = FINEBOT(M) / N(8)
      ENDIF

 500  CONTINUE
      RETURN
      END

C======================================================================
C==  SUBROUTINE XMINPAR                                              ==
C==     Computes minimum strength parameters in liquefied soil       ==
C==        RMNFS   = min. FSliq in liquefied soil                    ==
C==        SPTMNFS = N1,60 measured at RMNFS                         ==
C==        ZMINFS  = depth of RMNFS                                  ==
C==        RMNSPT  = min. N1,60 in a potentially liquefiable soil    ==
C==        ZMNSPT  = depth of RMNSPT                                 ==
C======================================================================
      SUBROUTINE XMINPAR (RMNFS, SPTMNFS, ZMNFS, RMNSPT, ZMNSPT,
     +                    NUMSPT, DEPTHN, SPTN160, FSLIQ, PLIQ, DWT)
      IMPLICIT REAL*4 (A-H,O-Z), INTEGER (I-N)
      COMMON /MEMORY/ MDEP, MSPT, MINT
      LOGICAL PLIQ
      DIMENSION DEPTHN(MSPT), SPTN160(MSPT), FSLIQ(MSPT), PLIQ(MSPT)

C--Find minimum values at measured SPT locations
      RMNFS  = 100.0
      MINFS  = 0
      RMNSPT = 1000.0
      MINSPT = 0

      DO 100 N=1, NUMSPT

C--Do not consider unsaturated or specified, nonliquefiable soils
      IF(PLIQ(N) .AND. (DEPTHN(N).GT.DWT)) THEN

C--Find minimum measured factor of safety
        IF( (FSLIQ(N) .GE. 0.0) .AND.
     +      (FSLIQ(N) .LT. RMNFS) ) THEN
          MINFS = N
          RMNFS = FSLIQ(N)
        ENDIF

C--Find minimum measured N1,60
        IF( SPTN160(N) .LT. RMNSPT) THEN
          MINSPT = N
          RMNSPT = SPTN160(N)
        ENDIF

      ENDIF
 100  CONTINUE

C--Check for undefined values, set remaining values and return
      IF(MINFS .EQ. 0) THEN
        RMNFS   = -1000.
        SPTMNFS = -1000.
        ZMNFS   = -1000.
      ELSE
        SPTMNFS = SPTN160(MINFS)
        ZMNFS   = DEPTHN(MINFS)
      ENDIF

      IF(MINSPT .EQ. 0) THEN
        RMNSPT = -1000.
        ZMNSPT = -1000.
      ELSE
        ZMNSPT = DEPTHN(MINSPT)
      ENDIF

      RETURN
      END
```