DEPARTMENT OF THE INTERIOR

U.S. GEOLOGICAL SURVEY

Source codes of CLAYFORM and associated file of data constants, a FORTRAN 77 computer program for calculating structural formulae of clay minerals and other silicates

by

Marc W. Bodine, Jr.[1]

Open-File Report 87-0004-A

DISCLAIMER

This program was written in FORTRAN 77 for the PRIME computer. Although program tests have been made, no guarentee (expressed or implied) is made by the author regarding program correctness, accuracy, or proper execution on all computer systems.

1987

[1]U.S. Geological Survey
MS 939, Box 25046, Denver Federal Center
Denver, CO 80225

---

* Any use of trade names in this report is for descriptive purposes only and does not imply endorsement by the U.S. Geological Survey. This report is preliminary and has not been reviewed for conformity with U.S. Geological Survey editorial standards.

TABLE OF CONTENTS

Abstract..... 1

Introduction..... 1

CLAYFORM source code..... 2

Program CLAYFORM..... 2

Documentation..... 2

Specification statements..... 3

Input identification and review..... 6

Entering mineral analysis data..... 10

Selecting structural formula specifications..... 14

Structural formula calculation..... 26

Formatting results of calculation..... 32

Printing out sample analysis and structural formula..... 39

Program continuation or termination..... 40

Format statements..... 41

Subroutine YESNO..... 47

Subroutine INSERT..... 47

Subroutine LISTCLAYS..... 48

Subroutines ARRAY and DGELG..... 50

FORMDAT source code, file of data constants for CLAYFORM..... 52

References..... 53

SOURCE CODES OF CLAYFORM AND ASSOCIATED FILE OF DATA CONSTANTS, A FORTRAN 77 COMPUTER PROGRAM FOR CALCULATING STRUCTURAL FORMULAE OF CLAY MINERALS AND OTHER SILICATES

by

Marc W. Bodine, Jr.

U.S. Geological Survey

MS 939, Box 25046, Denver Federal Center

Denver, Colorado 80225

ABSTRACT

The computer program CLAYFORM with its associated file of data constants calculates the structural formula of clay minerals and other silicate minerals from their chemical analyses. This report contains the source code for the program and a listing of the associated data file, FORMDAT.

INTRODUCTION

The FORTRAN 77 computer program CLAYFORM and its associated data file of constants, FORMDAT, were written to carry out rapid calculations of structural formulae of clay minerals and other silicate minerals from their conventional (weight percentages of constituent oxides) chemical analyses. Program rationale, use, and user options have been given in Bodine (1987) and only the program source code and a listing of FORMDAT is presented here. ARRAY and DGELG, two short subroutines in the listing, are published IBM codes (IBM Application Program, 1970) for solving linear simultaneous equations.

The program is operating on the U.S. Geological Survey's PRIME system at the Denver Federal Center and the ASCII source code is available on an IBM PC diskette (Open-File Report 87-0004-B) from Open-File Sevices, U.S. Geological Survey, Mail Stop 517, Box 25046, Denver, CO 80225.

SOURCE CODES

Program CLAYFORM

Documentation

```
c*********************************************************************
c                              CLAYFORM
c*********************************************************************
c                                                              May 1986
c
c  CLAYFORM, an interactive FORTRAN 77 program, calculates and displays
c  the structural formula of a clay or other silicate mineral from its
c  conventional (weight percentages of constituent oxides) chemical
c  analysis.  The program also creates and reads data files of silicate
c  mineral analyses.  The structural formula may be expressed as (1) a
c  general silicate mineral formula with cation sites undesignated, (2)
c  a general silicate mineral formula with cation sites designated but
c  unlabeled, or (3) a clay-mineral formula with cation sites desig-
c  nated and labeled (tetrahedral, octahedral, interlayer) and charge
c  distribution among the layers calculated.  The user enters the type
c  of structural formula and, if a clay-mineral type is specified, de-
c  fault specifications are assigned that can be modified by the user.
c  If a general silicate mineral formula is specified the user enters
c  the specifications of the structural formula.  These include the
c  parameters of the ideal stoichiometry that are to define the struc-
c  tural formula and a variety of properties (cation site assignments,
c  components to be deleted from the structural formula, redox state of
c  any iron in the analysis, character of the analyzed water as OH, H2O,
c  H3O+, or omitted from the structural formula).  The default specifi-
c  cations for the ten structural types of clay minerals can be similar-
c  ly modified.
c
c  References:
c
c    Bodine, M. W., Jr., 1987, CLAYFORM, a FORTRAN 77 computer program
c         apportioning the constituents in the chemical analysis of a
c         clay or other silicate mineral into a structural formula:
c         Computers & Geosciences, v. 13, p. 77-88.
c
c         A discussion of the rationale and use of CLAYFORM.
c
c    Bodine, M. W., Jr., 1987, Source codes of CLAYFORM and associated
c         file of data constants, a FORTRAN 77 program for calculating
c         sturcutral formulae of clay minerals and other silicates:
c         U.S. Geological Survey Open-File Report 87-0007, 53 p.
c
c         A listing (87-0007-A) and IBM-PC diskette (87-0007-B) of
c         CLAYFORM and FORMDAT (data constants) ASCII source codes.
c
c*********************************************************************
```

Specification statements

```
      character*1 ly,ln,la,lr,id,ll,al
      character*2 lid,lib,li
      character*4 none,lc,jc,a4,a4b
      character*5 lox,kfile,a5,a5b
      character*8 lcap,lfile,inpt,lform,a8,llst,a8b
      character*16 lcat,lbl,lbig,lll
      character*20 lraa,lras
      character*22 ltaa,ltas
      character*25 lt7
      character*27 lcz
      character*28 ha,habl,hff
      character*29 lt4,lcaa,lcas
      character*30 lt8
      character*31 lqa,lqb
      character*32 ltll
      character*33 lsit
      character*34 lt10,lwat,lctn
      character*35 hfa,hfp,lt2
      character*36 lt6
      character*37 lt5
      character*38 ltoc
      character*39 lneg
      character*40 lbt,ltl
      character*41 llt
      character*47 lt3
      character*49 lt9
      character*51 hf,hfbl
      character*75 ltit,175
      character*79 179,a79,a79b
      character*80 ldes,180
      logical*4 mol
      double precision adp,xdp
      dimension adp(3),xdp(9)
      dimension wm(50),kc(50),nc(50),kn(50),nn(50),io(50),ky(50),ny(50),
     a ndto(10),nd(10),nh(10),pw(50),xm(50),pm(50),xc(50),c(50),xa(50),
     a jtet(10),cz(3),nt(7),ntt(10),no(20),xnn(50),xnc(50),ndel(50),
     a mt(20),mo(20),mi(20),js(10),ne(6),qe(6),stot(5),a(50),nz(50),
     a ktt(7),na(10),ta(3),tx(3,3),ty(9),ioo(50),iww(3),xmm(50),
     a nw(10),ntto(10),
     s lr(20),lox(50),lc(50),jc(50),lbt(10),lcap(4),llt(10),lcat(5),
     s lcz(3),li(50),ha(99),hf(99),lcaa(5),lwat(3),a8(8),
     s id(5),al(8),a4(30),lraa(2),ltaa(4),llst(3,30),a79(20),a5(30)
      data lbt/'1:1 layer clay (kaolinite).              ',
     -'2:1 layer clay (mica).                  ',
     -'2:1 + 6OH layer clay (chlorite).        ',
     -'Mica-kaolinite 1:1 mixed-layer.         ',
     -'Mica-mica mixed-layer.                  ',
     -'Chlorite-kaolinite 1:1 mixed-layer.     ',
     -'Chlorite-mica 1:1 mixed-layer.          ',
     -'2:1 ribbon clay (palygorskite).         ',
     -'2:1 ribbon clay (sepiolite-8 oct.).     ',
```

```
-'2:1 ribbon clay (sepiolite-9 oct.).     '/
 data lcap/'weight %',' mole % ','--------','Total   '/
 data lraa/' with changes.      ','.                  '/
 data ltaa/'Tetr. filling order:  ','To octahedral sites:  ',
-'To interlayer sites:  ','To big cation sites:  '/
 data llt/'     1:1 layer clay (kaolinite type)     ',
-'       2:1 layer clay (mica type)        ',
-'  2:1 + 6OH layer clay (chlorite type)   ',
-'  Mixed-layer clay (1:1 mica-kaolinite)  ',
-'      Mixed-layer clay (mica-mica)       ',
-'Mixed-layer clay (1:1 chlorite-kaolinite)',
-'  Mixed-layer clay (1:1 chlorite-mica)   ',
-'     2:1 double chain (palygorskite)     ',
-'2:1 triple chain (sepiolite) 8 oct. sites',
-'2:1 triple chain (sepiolite) 9 oct. sites'/
 data lwat/'as OH anion in structural formula.',
-'as H2O molecular water.          ',
-'omitted from structural formula.  '/
 data lcz/'Tetrahedral layer charge = ',
-'Octahedral layer charge  = ',
-'Interlayer cation charge = '/
 data lt1/'Character of water in structural formula'/
 data lt2/'Iron redox in structural formula:  '/
 data lt3/'Component(s) deleted from structural formula:  '/
 data lt4/'Type of structural formula:  '/
 data lt5/'Oxygen equivalents per formula unit: '/
 data lt6/'Tetrahedral sites per formula unit: '/
 data lt7/'Cation site assignments: '/
 data lt8/'Anion sites per formula unit: '/
 data lt9/'Tetrahedral-octahedral cations per formula unit: '/
 data lt10/'(OH,F,Cl) sites per formula unit: '/
 data lt11/'Total cations per formula unit: '/
 data lcaa/'Standard clay-mineral default',
-'Chlorite clay-mineral default',
-'"Ribbon" clay-mineral default',
-'Reassign selected cations    ',
-'As previously user-modified  '/
 data hff/'Set to Fe2+/(Fe2+ + Fe3+) = '/
 data hfa/'FeO and Fe2O3 values in analysis. '/
 data a8b/'        '/
 data a5b/'     '/
 data a4b/'    '/
 data none/'None'/
 data lb1/'                '/
 data lib/'  '/
 data nd/7,11,14,18,22,21,25,21,32,33/
 data na/9,12,18,21,24,27,30,26,39,39/
 data jtet/2,4,4,6,8,6,8,8,12,12/
 data nt/1,3,0,0,0,0,0/
 data ntt/1,22,30,3,16,4,2,0,0,0/
 data nw/4,2,8,6,4,12,10,2,4,6/
 data ndto/4,6,8,10,12,12,14,13,20,21/
 data ntto/5,7,10,12,14,15,17,13,20,21/
```

```
      data ly/'y'/
      data ln/'n'/
      data lbig/'Large cations   '/
      data lqa/'Negative charge to be balanced?'/
      data lqb/'Type of clay structural formula'/
      data lneg/'Oxygen equivalents per formula unit =  '/
      data lsit/'Total anions per formula unit =  '/
      data ltoc/'Tet + oct cations per formula unit =  '/
      data lctn/'Total cations per formula unit =  '/
      data kfile/'CLAY.'/
      data id/'&','$','=','*',' '/
      data inpt/' UNFILED'/
      data ioo/1,2,14,30,3,15,16,4,5,6,17,
     -18,19,20,21,22,7,8,23,24,25,
     -9,10,26,27,28,29,11,12,13,20*0/
c
c*********************************************************************
c                     READING IN DATA FILE "FORMDAT"
c*********************************************************************
c
      write(hab1,9232)
      write(hfb1,9233)
      nwat=1
      xwat=1.0
      la=ln
      jntt=7
      nq = 0
      nl=0
      kal=0
      lr(8)=ln
      nox = 0
      ntoc=0
      itoc=0
      xtoc=0.0
      nofile=0
      nono=2
      nbeg=1
      notet=0
      nfic=0
      nfus=0
      open(11,status='old',file='formdat')
c                                                Molecular components
      do 12 i=1,50
      if(nq.ne.0) go to 10
      read(11,881)lox(i),jc(i),wm(i),kc(i),kn(i),ky(i)
      if(wm(i).eq.0.0) go to 10
      nox=nox+1
      go to 11
   10 nq=1
      lox(i)=lbl
      jc(i)=lbl
      wm(i)=0.0
      kc(i)=0
```

```
      kn(i)=0
      ky(i)=0
   11 continue
      lc(i)=jc(i)
      nc(i)=kc(i)
      nn(i)=kn(i)
      ny(i)=ky(i)
      io(i)=ioo(i)
      xnc(i)=float(nc(i))
      xnn(i)=float(nn(i))
   12 continue
      nos=nox
      nax=nox
      lr(8)=ln
      close(11)
      write(180,891)
      read(180,892) (lcat(i),i=1,5)
      hdel=0
      nl=100
      xiron=2.0
      nll=0
      ndbl=1
      do 80 i=1,50
      if(i.lt.11) nh(i)=nd(i)
      if(i.lt.21) lr(i)=ln
      if(i.lt.4) iww(i)=0
      ndel(i)=0
      nz(i)=0
   80 continue
      kloi=0
      iwat=0
      nform=0
      kwat=0
      kkwat=0
      ncats=0
      nzz=0
      itet=0
      go to 110
c
```

Input identification and review

```
c*********************************************************************
c          ANALYSIS FILE IDENTIFICATION, OPENING, AND REVIEWING
c*********************************************************************
c
  100 continue
      do 108 i=1,10
      ii=i
  103 continue
      if(ntt(i).eq.0) go to 105
      if(ntt(i).le.nax) go to 104
      ii=ii+1
```

```
      if(ii.gt.10) ntt(i)=0
      if(ntt(i).eq.0) go to 105
      ntt(i)=ntt(ii)
      go to 103
  104 jntt=i+1
  105 ii=i
      if(i.gt.7) go to 108
  106 continue
      if(nt(i).eq.0) go to 108
      if(nt(i).le.nax) go to 108
      ii=ii+1
      if(ii.gt.7) nt(i)=0
      if(nt(i).eq.0) go to 108
      nt(i)=nt(ii)
      go to 106
  108 continue
  110 continue
      nono=2
      do 112 i=11,20
  112 lr(i)=ln
      nal=0
  114 write(1,851)
      call yesno(lr(12),nerr)
      if(nerr.ne.0) go to 114
      if(lr(12).eq.ly) go to140
  115 write(1,852)
      call yesno(lr(11),nerr)
      if(nerr.ne.0) go to 115
      if(lr(11).eq.ln) go to 117
c
c             Identifying a new data file to be created
c             -----------------------------------------
c
  116 write(1,8531)
      call yesno(lr(14),nerr)
      if(nerr.ne.0) go to 116
      do 125 i=1,999
      write(lfile,854) i
      inquire(file=lfile,exist=mol)
      if(mol) go to125
      open(12,status='new',file=lfile,err=125)
      write(1,857) lfile
  117 write(1,855)
      read(1,999,err=117) ltit
      write(1,999) ltit
      write(1,994)
      call yesno(la,nerr)
      if(nerr.ne.0) go to 117
      if(la.eq.ln) go to 117
      if(lr(11).eq.ln) go to 200
      write(12,999)ltit
      open(14,file='CLAY.ID',status='old',err=200)
      read(14,998,end=123) 180
```

```
      do 121 j=1,999
      read(14,853,end=123) jj,175
      if(jj.eq.i) go to122
  121 continue
  122 continue
      backspace(14)
  123 continue
      write(14,853) i,1tit
      close(14)
      go to 200
  125 continue
c
c               Naming and opening an existing data file
c               ----------------------------------------
c
  140 continue
      write(1,858)
      read(1,9921,err=140) nq
      if(nq.ne.0) go to 141
      call listclays
      go to 140
  141 write(1file,854) nq
      if(nox.eq.nax) go to 145
      n=nax+1
      do 142 i=1,50
      io(i)=ioo(i)
      if(i.1t.n) go to 142
      xnn(i)=0.0
      xnc(i)=0.0
      lox(i)=1b1
      jc(i)=1b1
      kc(i)=0
      kn(i)=0
      ky(i)=0
      wm(i)=0.0
  142 continue
      nox=nax
  145 continue
      open(13,file=1file,status='old',err=146)
      go to 150
  146 write(1,859) 1file
      call yesno(1a,nerr)
      if(nerr.ne.0) go to 146
      close(13)
      if(1a.eq.1y) go to140
      go to 799
  150 write(1,861)
      call yesno(1a,nerr)
      if(nerr.ne.0) go to 150
      na1 = 0
      if(1a.eq.1y) write(1,869) 1file
      read(13,850,err=146,end=146) 11,179
c
```

```
c                       Scanning existing data file
c                       ---------------------------
c
      do 163 i=1,1000
      read(13,850,err=762,end=164) 11,179
      if(11.eq.id(1).or.11.eq.id(3)) go to 163
      if(11.eq.id(2).and.1r(11).eq.1y) go to 163
      write(180,850) 11,179
      if(11.ne.id(2)) go to 162
c                                   Reading file for non-bank components
      nox=nox+1
      n=nox
      read(180,873,err=762)11,1ox(n),jc(n),wm(n),kc(n),kn(n),
     -ky(n)
      io(n)=n
      1c(n)=jc(n)
      nc(n)=kc(n)
      nn(n)=kn(n)
      ny(n)=ky(n)
      xnc(n)=float(nc(n))
      xnn(n)=float(nn(n))
      call insert(ny,nc,nn,io,n)
      if(ny(n).ne.0) go to 161
      ny(n)=1
      ntt(jntt)=n
      jntt=jntt+1
  161 continue
      go to 163
c                                                       Listing samples
  162 nal=nal+1
      if(1a.eq.1y) write(1,863) nal,180
  163 continue
  164 continue
      if(1a.eq.1y) write(1,997)
      rewind(13)
      nal=0
c                     Designating specific samples in file for calculation
  165 write(1,864)
      call yesno(1r(13),nerr)
      if(nerr.ne.0) go to 165
      if(1r(13).eq.1n) go to 180
  166 write(1,865)
      read(1,866,err=166) (js(i),i=1,10)
  180 continue
c
c           Reading a single analysis from existing data file
c           -------------------------------------------------
c
      do 181 i=1,50
  181 pw(i)=0.0
      if(nal.eq.0) read(13,999)1tit
      read(13,850,end=762) 11,179
      if(11.eq.id(2)) go to 762
```

```
      if(ll.eq.id(1).or.ll.eq.id(3)) go to 761
      write(180,850) ll,179
      ldes=180
      nal=nal+1
      do 184 i=1,20
      read(13,850,end=761) ll,179
      write(180,850) ll,179
      if(ll.ne.id(1)) go to 183
      read(180,872,err=761) ll,(ne(k),qe(k),k=1,6)
      do 182 k=1,6
      if(ne(k).gt.0) pw(ne(k))=qe(k)
  182 continue
      go to 184
  183 if(ll.ne.id(3)) go to 761
      read(180,874) ll,lr(7)
      if(lr(7).ne.ly.and.lr(7).ne.ln) go to 761
      go to 185
  184 continue
  185 continue
      if(lr(13).eq.ln) go to 200
      do 186 i=1,10
      if(nal.eq.js(i)) go to 200
      if(js(i).eq.0) go to 180
  186 continue
      go to 180
  200 continue
c
```

Entering mineral analysis data

```
c*********************************************************************
c                    ENTERING MINERAL ANALYSIS DATA
c*********************************************************************
c
      lr(15)=ln
      if(nono.eq.0) go to 210
      if(lr(14).eq.ly) go to 210
  205 write(1,9871)
      call yesno(lr(15),nerr)
      if(nerr.ne.0) go to 205
  210 continue
c                                          Entering major oxides
      if(lr(12).eq.ly) go to 280
      write(1,901)
      do 212 i=1,50
      pw(i)=0.0
      if(i.gt.13) go to 212
  211 write(1,902) lox(i)
      read(1,903,err=211) pw(i)
  212 continue
c                                          Entering minor oxides
  213 write(1,904)
      call yesno(la,nerr)
```

```
      if(nerr.ne.0) go to 213
      if(la.eq.ln) go to 231
      write(1,905)
      nl=18
      do 215 n3=1,10
      if(nl.ge.nox) go to 216
  215 nl=nl+5
  216 nl=14
      do 217 i=1,n3
      n2=nl+(4*n3)
      write(1,906) (k,lox(k),k=nl,n2,n3)
  217 nl=nl+1
  218 write(1,907)
      read(1,908,err=218) n,q
      if(n.eq.0) go to 220
      pw(n)=q
      go to 218
  220 continue
c                                        Entering unlisted components
  221 write(1,909)
      call yesno(la,nerr)
      if(nerr.ne.0) go to 221
      if(la.eq.ln) go to 231
  222 write(1,911)
      n=nox+1
      read(1,912,err=122) lox(n)
  223 write(1,913)
      read(1,914,err=223) jc(n)
      lc(n)=jc(n)
  224 write(1,915)
      read(1,903,err=224) wm(n)
      if(wm(n).eq.0.0) go to 231
  225 write(1,916)
      read(1,991,err=225) kc(n)
      nc(n)=kc(n)
  226 write(1,917)
      read(1,991,err=226) kn(n)
      nn(n)=kn(n)
  227 write(1,918)
      read(1,991,err=227) ky(n)
      ny(n)=ky(n)
      nq=ky(n)+1
      write(1,926) lox(n),jc(n),wm(n),kc(n),kn(n),lcat(nq)
  228 write(1,994)
      call yesno(la,nerr)
      if(nerr.ne.0) go to 228
      if(la.eq.ln) go to 222
      nox=n
      if(ny(n).ne.0) go to 229
      ny(n)=1
      ntt(jntt)=n
      jntt=jntt+1
  229 continue
```

```
      io(n)=nox
      xnc(n)=float(nc(n))
      xnn(n)=float(nn(n))
      call insert(ny,nc,nn,io,n)
      write(1,901)
  230 write(1,902) lox(nox)
      read(1,903,err=230) pw(nox)
      go to 221
  231 continue
c                                        Status of iron analytical data
      lr(7)=ln
      n=0
      if(pw(4).gt.0.0) n=n+1
      if(pw(5).gt.0.0) n=n+1
      if(n.ne.1) go to 233
  232 write(1,919)
      call yesno(lr(7),nerr)
      if(nerr.ne.0) go to 232
  233 continue
c                                           Entering sample description
  234 write(1,920)
      read(1,998,err=234) ldes
  235 continue
      nofile=0
c
c                     Confirming analysis entry
c                     -------------------------
c
  238 write(1,921)
      do 240 i=1,nox
      if(pw(i).eq.0.0) go to 240
      write(1,922) i,lox(i),pw(i)
  240 continue
  241 write(1,923)
      call yesno(la,nerr)
      if(nerr.ne.0) go to 241
      if(la.eq.ly) go to 250
  242 write(1,924)
  243 write(1,907)
      read(1,908,err=243) i,q
      if(i.eq.99) go to 200
      if(i.eq.0) go to 250
      if(i.le.nox) go to 249
      nl=5
      do 245 n3=1,10
      if(nl.ge.nox) go to 246
  245 nl=nl+5
  246 nl=1
      do 247 i=1,n3
      n2=nl+(4*n3)
      write(1,906) (k,lox(k),k=nl,n2,n3)
  247 nl=nl+1
      go to 243
```

```
  249 pw(i)=q
      go to 242
  250 continue
      if(la.eq.ln) go to 238
  255 write(1,925) ldes
      call yesno(la,nerr)
      if(nerr.ne.0) go to 255
      if(la.eq.ly) go to 260
  256 write(1,920)
      read(1,998) ldes
  260 continue
c
c                Writing clay analysis into new data file
c                ----------------------------------------
c
      if(lr(12).eq.ly) go to 280
      if(nofile.eq.1) go to 280
      nal=nal+1
      if(lr(11).eq.ln) go to 280
      write(12,998)ldes
      nq=0
      do 265 i=1,nox
      if(pw(i).eq.0.0.and.i.lt.nox) go to 265
      if(pw(i).eq.0.0) go to 262
      nq=nq+1
      ne(nq)=i
      qe(nq)=pw(i)
      if(nq.eq.6) go to 264
      if(i.lt.nox) go to 265
  262 n=nq+1
      if(n.gt.6) go to 264
      do 263 k=n,6
      ne(k)=0
  263 qe(k)=0.0
  264 write(12,872) id(1),(ne(k),qe(k),k=1,6)
      nq=0
  265 continue
      write(12,874) id(3),lr(7)
      if(lr(14).eq.ly) go to 755
      go to 280
c
c             Writing non-bank components into new data file
c             ----------------------------------------------
c
  270 continue
      if(nax.eq.nox) go to 275
      nnqq=nax+1
      do 271 i=nnqq,nox
  271 write(12,873,err=761,end=275) id(2),lox(i),jc(i),wm(i),kc(i),kn(i)
     -,ky(i)
  275 lr(11)=ln
      close(12)
      go to 780
```

```
  280 continue
c
c                 Status of LOI (loss-on-ignition)
c                 --------------------------------
c
      if(pw(13).eq.0.or.lr(15).eq.ln) go to 282
      write(1,875)
      read(1,991,err=280) kloi
      if(kloi.lt.0) go to 280
  282 continue
      if(lr(15).eq.ly) go to 400
      qiron=pw(4)+pw(5)
      nfic=0
      nfus=0
      if(qiron.eq.0.0) go to 284
      if(pw(4).gt.0.0) nfic=1
      if(pw(5).gt.0.0) nfus=1
      if(xiron.gt.1.1) go to 284
      nfic=1
      nfus=1
      if(xiron.eq.0.0) nfus=0
      if(xiron.eq.1.0) nfic=0
  284 continue
c
```

Selecting structural formula specification

```
c*********************************************************************
c            SELECTING STRUCTURAL FORMULA SPECIFICATIONS
c*********************************************************************
c
c
c             Summary listing of current specifications
c               -------------------------------------
c
      if(nono.eq.0) go to 400
      if(lr(12).eq.ly) write(1,998) ldes
      if(nl.ne.100) go to 305
      nnoh=0
      q=pw(11)+pw(12)
      if(kloi.ne.0) q=q+pw(13)
      if(q.gt.0.0) nnoh=1
      nform=0
      go to 340
  305 continue
      nbeg=2
      write(1,8001)
c                                   Listing of water characetistics
      nnoh=0
      qwat=pw(11)+pw(12)+pw(13)
      if(qwat.eq.0.0) go to 3053
      write(1,8002) ltl
      if(pw(13).eq.0.0) go to 3051
```

```
      if(klio.eq.0) write(1,876)
      if(klio.ne.0) write(1,877)
      if(klio.eq.0.and.pw(13).ne.0.0) iww(3)=3
 3051 continue
      do 3052 i=1,3
      if(i.eq.3.and.kloi.eq.0) go to 3052
      k=i+10
      if(pw(k).eq.0.0) go to 3052
      if(i.eq.1.and.iww(i).eq.0) iww(i)=1
      if(i.gt.1.and.iww(i).eq.0) iww(i)=3
      write(1,8003) lox(k),lwat(iww(i))
      if(iwat.gt.0.and.iww(i).eq.1) nnoh=1
 3052 continue
      if(nnoh.eq.0) go to 3053
      if(iwat.eq.1) write(1,8004)
      if(iwat.eq.2) write(1,8005)
c                                     Listing of iron redox properties
 3053 continue
      if(qiron.eq.0.0) go to 3055
      if(xiron.gt.1.1) hfp=hfa
      if(xiron.lt.1.1) write(hfp,8006) hff,xiron
      write(1,8007) lt2,hfp
c                 Listing of components deleted from structural formula
 3055 continue
      if(hdel.ne.0) go to 3057
      write(1,8008) lt3
      do 3056 i=1,nox
 3056 ndel(i)=0
      go to 306
 3057 nq=0
      do 3058 i=1,nox
      if(i.lt.21) no(i)=40
      if(ndel(i).eq.0) go to 3058
      nq=nq+1
      no(nq)=i
 3058 continue
      if(nq.ne.0) go to 3059
      hdel=0
      go to 3055
 3059 write(1,8009) lt3,(lox(no(i)),i=1,5)
      if(nq.gt.5) write(1,8010) (lox(no(i)),i=6,20)
c                                   Listing of structural formula type
  306 continue
      if(nl.eq.0) write(1,8031) lt4
      if(nl.gt.10) write(1,8032) lt4
      if(nl.gt.0.and.nl.lt.11) write(1,8012) lt4,lbt(nl)
c               Listing of calculation parameters in structural formula
      write(1,8025)
      if(nform.eq.0) write(1,8013) lt5,nneg
      if(nform.eq.1) write(1,8020) lt8,nneg
      if(nform.eq.2.and.ncats.eq.0) write(1,8021) lt9,ntoc
      if(nform.eq.2.and.ncats.eq.1) write(1,8022) lt11,ntoc
      if(nl.ne.0) write(1,8014) lt6,ntet
```

```
      if(iwat.ne.0.and.nnoh.ne.0) write(1,8026) lt10,nwat
c                 Listing of cation site assignments in a phyllosilicate
      if(nl.eq.0) go to 307
      if(nzz.eq.0) lras=lraa(2)
      if(nzz.ne.0) lras=lraa(1)
      write(1,8015) lt7,lcas,lras
      if(nzz.eq.0) go to 307
      do 3069 i=1,3
      do 3061 k=1,30
 3061 a4(k)=a4b
      if(i.eq.1.and.itet.eq.0) go to 3069
      if(i.ne.1) go to 3065
      ltas=ltaa(1)
      nq=0
      do 3062 k=1,7
      if(nt(k).eq.0) go to 3063
      nq=nq+1
 3062 a4(nq)=lc(nt(k))
 3063 continue
      go to 3068
 3065 nq=0
      l=i-1
      ltas=ltaa(i)
      do 3066 k=1,nox
      if(nz(k).eq.0.or.ny(k).ne.1) go to 3066
      nq=nq+1
      a4(nq)=lc(k)
 3066 continue
      if(nq.eq.0) go to 3069
 3068 write(1,8018) ltas,(a4(k),k=1,10)
      if(nq.gt.10) write(1,8019) (a4(k),k=11,20)
      if(nq.gt.20) write(1,8019) (a4(k),k=21,30)
 3069 continue
      if(notet.eq.0) go to 307
      nq=0
      do 3067 i=1,7
      a4(i)=a4b
      if(nt(i).eq.0) go to 3067
      if(ny(nt(i)).ne.0) go to 3067
      nq=nq+1
      a4(nq)=lc(nt(i))
 3067 continue
      if(nq.eq.0) a4(1)=none
      write(1,8030) (a4(i),i=1,7)
c
c     Electing options for modifying structural formula parameters
c     ------------------------------------------------------------
c
  307 continue
      if(nbeg.eq.2) go to 308
      if(nl.eq.0) write(1,8028)
      if(nl.ne.0) write(1,8029)
  308 write(1,8017)
```

```
      read(1,991,err=307) n
      if(n.lt.0) go to 307
      if(n.gt.6) go to 305
      nbeg=0
      if(n.eq.0) go to 390
      if(n.eq.1) go to 310
      if(n.eq.2) go to 320
      if(n.eq.3) go to 330
      if(n.eq.4) go to 340
      if(n.eq.5) go to 350
      if(n.eq.6) go to 370
c
c         Modifying water characteristics in structural formula
c         -----------------------------------------------------
c
  310 continue
      if(pw(13).eq.0.0) go to 3101
      write(1,875)
      read(1,991,err=310) kloi
 3101 continue
      do 311 i=1,3
  311 iww(i)=0
      do 315 i=1,3
      ii=i+10
      if(pw(ii).eq.0.0) go to 315
      if(kloi.eq.0.and.i.eq.3) go to 315
      k1=1
      k2=3
      if(i.gt.1.and.iww(1).gt.1) k1=2
      if(i.gt.1.and.iww(1).gt.2) k1=3
      if(i.ne.3) go to 312
      if(iww(2).gt.1.and.k1.lt.2) k1=2
      if(iww(2).gt.2.and.k1.lt.3) k1=3
  312 write(1,8210) lox(ii)
  313 continue
      do 314 k=k1,k2
  314 write(1,8211) k,lwat(k)
      read(1,991,err=313) iww(i)
      if(iww(i).lt.1.or.iww(i).gt.3) go to 312
  315 continue
      meg=0
      if(iwat.ne.0) meg=1
      iwat=0
      do 3161 i=1,3
      if(iww(i).eq.1) go to 3162
 3161 continue
      go to 317
 3162 write(1,8212)
      read(1,991,err=3162) iwat
      if(iwat.lt.0.or.iwat.gt.2) go to 3162
      mug=0
      if(iwat.eq.0) go to 317
      if(meg.ne.0) go to 317
```

```
      if(nl.gt.0.and.nl.lt.11) go to 317
      mug=1
      nnoh=1
      go to 3571
  317 continue
      mug=0
      go to 307
c
c      Modifying iron redox characteristics in structural formula
c      ----------------------------------------------------------
c
  320 xiron=2.0
  321 write(1,8220)
      call yesno(lr(3),nerr)
      if(nerr.ne.0) go to 321
      nfic=0
      nfus=0
      if(pw(4).gt.0.0) nfic=1
      if(pw(5).gt.0.0) nfus=1
      if(lr(3).eq.ln) go to 307
  322 write(1,8221)
      read(1,903,err=322) xiron
      if(xiron.lt.0.0.or.xiron.gt.1.0) go to 322
  323 write(1,8222) hff,xiron
      nfic=1
      nfus=1
      if(xiron.eq.0.0) nfus=0
      if(xiron.eq.1.0) xfic=0
      call yesno(la,nerr)
      if(nerr.ne.0) go to 323
      if(la.eq.ln) go to 320
      go to 307
c
c   Modifying list of components for deletion from structural formula
c   -----------------------------------------------------------------
c
  330 nq=0
      nng=0
      if(hdel.eq.0) go to 332
  331 write(1,8230)
      read(1,991,err=331) nng
      if(nng.lt.1.or.nng.gt.3) go to 331
      if(nng.eq.1) go to 3320
  332 write(1,8231)
      read(1,991,err=332) nm
 3320 hdel=0
      nq=0
      nnq=0
      do 333 i=1,nox
      if(nng.ne.3) ndel(i)=0
      if(nng.eq.1) go to 333
      if(nnq.ne.0) go to 3322
      do 3321 k=1,8
```

```
      a8(k)=a8b
 3321 al(k)=id(5)
 3322 continue
      if(nm.ne.0) go to 3326
      if(i.ne.4.and.i.ne.5) go to 3325
      if(xiron.gt.1.1.or.qiron.eq.0.0) go to 3325
      go to 3326
 3325 continue
      if(pw(i).eq.0.0.and.i.lt.nox) go to 333
 3326 nq=nq+1
      if(ndel(i).ne.0) al(nq)=id(4)
      if(pw(i).ne.0.0.or.nm.ne.0) write(a8(nq),8233) i,lox(i)
      nnq=1
      if(nq.ne.8.and.i.ne.nox) go to 333
      write(1,8234) (al(k),a8(k),k=1,8)
      nnq=0
      nq=0
  333 continue
      if(nng.eq.1) go to 307
      write(1,8235)
      if(nng.eq.3) write(1,8236)
  334 write(1,8237)
      read(1,8662,err=334) (mo(i),i=1,20)
      do 335 i=1,20
      if(mo(i).eq.0) go to 336
      if(ndel(mo(i)).eq.0) ndel(mo(i))=2
      if(ndel(mo(i)).eq.1) ndel(mo(i))=0
      if(ndel(mo(i)).eq.2) ndel(mo(i))=1
  335 continue
  336 continue
      do 337 i=1,nox
      if(hdel.ne.0) go to 338
      if(ndel(i).eq.1) hdel=1
  337 continue
  338 continue
      go to 307
c
c             Modifying structural formula type
c             ---------------------------------
c
  340 continue
      write(1,8240)
      read(1,991,err=340) n
      if(n.lt.1.or.n.gt.2) go to 340
      if(n.eq.2) go to 341
      n1=0
      n11=0
  348 write(1,8242)
      call yesno(la,nerr)
      if(nerr.ne.0) go to 348
      if(la.eq.ln) go to 350
      lcas=lcaa(1)
      lras=lraa(2)
```

```
      nl=11
      go to 350
  341 continue
      do 342 i=1,20
      ii=i+10
  342 mi(i)=ii
  343 write(1,824) (mi(i),lbt(i),i=1,10)
      read(1,992,err=343) n
      nl=n-10
      if(nl.lt.1.or.nl.gt.10) go to 343
      do 344 i=1,nox
      if(nz(i).ne.0) go to 344
      ny(i)=ky(i)
      if(nl.eq.3.and.ky(i).eq.2) ny(i)=1
  344 continue
      if(nl.gt.7.and.nz(9).eq.0) ny(9)=1
      if(nl.gt.7.and.nz(19).eq.0) ny(19)=2
      lcas=lcaa(1)
      if(nl.eq.3) lcas=lcaa(2)
      if(nl.gt.7) lcas=lcaa(3)
      if(nform.eq.0) kneg=nd(nl)
      if(nform.ne.0) kneg=na(nl)
      ktoc=ndto(nl)
      kwat=nw(nl)
      ntet=jtet(nl)
      nqq=1
      go to 3501
  346 nll=1
      if(nl.eq.3) nll=2
      if(nl.gt.7) nll=3
  345 xneg=float(nneg)
      xtet=float(ntet)
      xtoc=float(ntoc)
      xwat=float(nwat)
      if(nbeg.eq.1) go to 305
      if(mug.eq.1) go to 317
      go to 307
c
c             Modifying parameters of structural formula
c             ------------------------------------------
c
  350 nqq=0
c                                   Selecting method of calculation
 3501 write(1,8250)
      read(1,991,err=3501) nform
      if(nform.lt.0.or.nform.gt.2) go to 350
      if(nl.lt.1.or.nl.gt.10) go to 359
c                                 Assignment of default parameters
      if(nform.eq.0) kneg=nd(nl)
      if(nform.eq.1) kneg=na(nl)
      ktoc = ndto(nl)
      kwat=nw(nl)
  359 continue
```

```
      if(nform.ne.2) go to 352
      write(1,8244)
      read(1,991,err=359) ncats
      if(ncats.lt.0.or.ncats.gt.1) go to 359
      if(nl.lt.1.or.nl.gt.10) go to 353
      if(ncats.eq.1) go to 353
  351 write(1,8243)
      call yesno(la,nerr)
      if(nerr.ne.0) go to 351
      if(la.eq.ln) ktoc=ntto(nl)
  352 continue
      if(nl.lt.1.or.nl.gt.10) go to 353
      write(1,8252)
      call yesno(la,nerr)
      if(nerr.ne.0) go to 352
      ndbl=1
      if(la.eq.ly) ndbl=2
      nneg=ndbl*kneg
      ntoc=ndbl*ktoc
      nwat=ndbl*kwat
      ntet=ndbl*jtet(nl)
      if(nqq.eq.1) go to 346
      go to 360
c                                Modifying basic calculation parameter
  353 continue
      if(nform.eq.0) write(1,8253) lt5,nneg
      if(nform.eq.1) write(1,8254) lt8,nneg
      if(nform.eq.2.and.ncats.eq.0) write(1,8255) lt9,ntoc
      if(nform.eq.2.and.ncats.eq.1) write(1,8258) lt11,ntoc
      write(1,994)
      call yesno(la,nerr)
      if(nerr.ne.0) go to 353
      if(la.eq.ly.and.nqq.eq.1) go to 352
      if(la.eq.ly) go to 355
  354 write(1,8251)
      read(1,992,err=354) n
      if(nform.lt.2) nneg=n
      if(nform.eq.2) ntoc=n
      if(nqq.eq.1) ktoc=ntoc
      if(nqq.eq.1) go to 352
      go to 353
c                  Modifying number of tetrahedral sites in formula unit
  355 continue
      if(nl.eq.0) go to 357
      write(1,8256) lt6,ntet
      write(1,994)
      call yesno(la,nerr)
      if(nerr.ne.0) go to 355
      if(la.eq.ly) go to 357
  356 write(1,8251)
      read(1,992,err=356) ntet
      go to 355
c                    Modifying number of (OH,F,Cl) sites in formula unit
```

```
  357 continue
      if(iwat.eq.0.or.nnoh.eq.0) go to 360
 3571 write(1,8257) lt10,nwat
      write(1,994)
      call yesno(la,nerr)
      if(nerr.ne.0) go to 3571
      if(la.eq.ly.and.mug.eq.1) go to 345
      if(la.eq.ly) go to 360
  358 write(1,8251)
      read(1,992,err=358) nwat
      go to 357
c                                       Check-list of formula parameters
  360 write(1,8260)
      if(nform.eq.0) write(1,8253) lt5,nneg
      if(nform.eq.1) write(1,8254) lt8,nneg
      if(nform.eq.2.and.ncats.eq.0) write(1,8255) lt9,ntoc
      if(nform.eq.2.and.ncats.eq.1) write(1,8258) lt11,ntoc
      if(nl.gt.0) write(1,8256) lt6,ntet
      if(iwat.eq.0.or.nnoh.eq.0) go to 362
      write(1,8257) lt10,nwat
  362 write(1,994)
      call yesno(la,nerr)
      if(nerr.ne.0) go to 362
      if(la.eq.ln) go to 353
      go to 345
c
c        Modifying cation site assignments in clay-mineral formula
c        ---------------------------------------------------------
c
  370 write(1,8270) (i,lcaa(i),i=1,3)
      read(1,991,err=370) n
      nm=2
      if(n.lt.1.or.n.gt.3) go to 370
c                                           Reassigning default-type
      la=ln
      if(nzz.eq.0) go to 3703
 3701 write(1,8278)
      call yesno(la,nerr)
      if(nerr.ne.0) go to 3701
 3703 continue
      if(n.gt.0.and.n.lt.4) nl1=n
      do 371 i=1,nox
      if(la.eq.ly.and.nz(i).ne.0) go to 371
      nz(i)=0
      ny(i)=ky(i)
      if(nl1.eq.2.and.ky(i).eq.2) ny(i)=1
  371 continue
      if(nl1.ne.3) go to 372
      if(nz(9).eq.0) ny(9)=1
      if(nz(19).eq.0) ny(19)=2
  372 lcas=lcaa(n)
      if(la.eq.ly) go to 373
      nt(1)=1
```

```
      nt(2)=3
      do 3725 i=3,7
 3725 nt(i)=0
      nzz=0
      itet=0
      notet=0
  373 write(1,8271)
      call yesno(la,nerr)
      if(nerr.ne.0) go to 373
      if(la.eq.ln) go to 307
c                          Displaying listing of cation site assignments
  374 jt=0
      jo=0
      ji=0
      do 3742 k=1,30
      do 3741 i=1,3
 3741 llst(i,k)=a8b
 3742 continue
      if(nm.lt.2) go to 3744
 3743 write(1,8231)
      read(1,991,err=3743) nm
 3744 nzz=0
      itet=0
      do 375 i=1,nox
      if(ny(i).gt.2) go to 375
      if(nm.ne.0) go to 3746
      if(i.ne.4.and.i.ne.5) go to 3745
      if(i.eq.4.and.nfic.eq.1) go to 3746
      if(i.eq.5.and.nfus.eq.1) go to 3746
      go to 375
 3745 continue
      if(pw(i).eq.0.0) go to 375
 3746 ll=id(5)
      if(nz(i).ne.0) ll=id(4)
      if(ll.eq.id(4)) nzz=1
      if(ny(i).ne.1) go to 3747
      jo=jo+1
      write(llst(2,jo),8273) ll,i,lc(i)
 3747 continue
      if(ny(i).ne.2) go to 375
      ji=ji+1
      write(llst(3,ji),8273) ll,i,lc(i)
  375 continue
      h1=0
      h2=0
      do 376 i=1,7
      ll=id(5)
      if(nt(i).eq.0) go to 376
      jt=jt+1
      if(nt(i).ne.1.and.nt(i).ne.3) go to 3751
      if(nt(i).eq.1) h1=1
      if(nt(i).eq.1.and.h2.eq.1) go to 3751
      if(nt(i).eq.3) h2=1
```

```
      if(nt(i).eq.3.and.h1.eq.0) go to 3751
      go to 3752
 3751 itet=1
      if(nt(i).ne.1.and.nt(i).ne.3) l1=id(4)
 3752 write(llst(1,jt),8273) l1,nt(i),lc(nt(i))
  376 continue
      if(h1.eq.0.or.h2.eq.0) itet=1
      if(itet.ne.0) nzz=1
  377 ii=jt
      if(jo.gt.ii) ii=jo
      if(ji.gt.ii) ii=ji
      write(1,8274)
      if(n1.gt.10) write(1,8288) (lcat(i),i=1,2)
      if(n1.gt.0.and.n1.lt.11) write(1,8289) (lcat(i),i=1,3)
      do 378 i=1,ii
  378 write(1,8275) (llst(j,i),j=1,3)
      nq=0
      do 3781 i=1,7
      a4(i)=a4b
      if(nt(i).eq.0) go to 3781
      if(ny(nt(i)).ne.0) go to 3781
      nq=nq+1
      a4(nq)=lc(nt(i))
 3781 continue
      if(nq.eq.0) a4(1)=none
      write(1,8030) (a4(i),i=1,7)
      if(nzz.ne.0.or.itet.ne.0) write(1,8276)
c                                       Requesting reassignment option
  379 write(1,8277)
      if(nzz.ne.0) write(1,8279)
      read(1,991,err=379) n
      if(n.lt.0.or.n.gt.4) go to 379
      if(n.eq.0) go to 307
      if(n.eq.3) go to 382
c                           Octahedral and interlayer site reassignments
  380 continue
      if(n.lt.4) write(1,8280) lcat(n+1)
      if(n.eq.4) write(1,8287)
      read(1,992,err=380) nq
      if(nq.lt.0.or.nq.gt.nox) go to 380
      if(nq.eq.0) go to 374
      if(n.eq.4) go to 381
       nz(nq)=1
      ny(nq)=n
      go to 380
  381 nz(nq)=0
      ny(nq)=ky(nq)
      if(n11.eq.2.and.ny(nq).eq.2) ny(nq)=1
      if(n11.eq.3.and.nq.eq.9) ny(nq)=1
      if(n11.eq.3.and.nq.eq.19) ny(nq)=2
      go to 380
c                         Reassigning order of filling tetrahedral sites
  382 write(1,8281)
```

```
      do 383 i=1,7
      if(nt(i).eq.0) go to 3831
  383 write(1,8282) i,nt(i),lc(nt(i))
 3831 write(1,994)
      call yesno(la,nerr)
      if(nerr.eq.1) go to 3831
      if(la.eq.ly) go to 389
      write(1,8283)
      do 384 i=1,10
      if(ntt(i).eq.0) go to 385
  384 write(1,8284) ntt(i),lc(ntt(i))
  385 write(1,8285)
      read(1,8286,err=385) (nt(i),i=1,7)
      go to 382
  389 continue
      write(1,8293)
      call yesno(la,nerr)
      if(nerr.ne.0) go to 389
      if(la.eq.ln) go to 3901
 3891 write(1,8294)
      read(1,8286,err=3891) (ktt(i),i=1,7)
      j=1
      jj=0
      do 3895 i=1,7
      if(ktt(i).eq.0) go to 3896
      if(ktt(i).eq.1) j=0
      if(ktt(i).gt.1) jj=1
      ny(ktt(i))=0
      do 3892 k=1,7
      if(ktt(i).eq.nt(k)) go to 3895
      if(nt(k).eq.0) go to 3893
 3892 continue
 3893 nt(k)=ktt(i)
 3895 continue
 3896 continue
      notet=0
      if(j.eq.1.or.jj.eq.1) notet=1
      k=0
      do 3897 i=1,7
      ktt(i)=0
      if(nt(i).eq.0) go to 3897
      if(ny(nt(i)).ne.0) go to 3897
      k=k+1
      ktt(k)=nt(i)
      nt(i)=0
 3897 continue
      do 3898 i=1,7
      if(nt(i).eq.0) go to 3898
      k=k+1
      ktt(k)=nt(i)
 3898 nt(i)=0
      do 3899 i=1,7
      nt(i)=ktt(i)
```

```
 3899 ktt(i)=0
 3901 continue
      do 3906 i=1,nox
      if(ny(i).ne.0) go to 3905
      do 3903 k=1,7
      if(nt(k).eq.i) go to 3905
 3903 continue
      ny(i)=ky(i)
      if(ny(i).eq.0) ny(i)=1
      if(ny(i).ne.ky(i)) nz(i)=1
 3905 continue
      if(i.gt.1.and.ny(i).eq.0) notet=1
 3906 continue
      if(ny(1).ne.0) notet=1
      go to 374
c                    Setting constants for calculating structural formula
  390 continue
      if(nono.ne.2) go to 400
  391 write(1,8290)
      call yesno(la,nerr)
      if(nerr.ne.0) go to 391
      if(la.eq.ln) nono=1
c
```

Structural formula calculation

```
c*********************************************************************
c                    STRUCTURAL FORMULA CALCULATION
c*********************************************************************
c
c    Mole percentages and abundances in general silicate formula
c    ------------------------------------------------------------
c
  400 continue
      ibor=0
      isil=0
      kbor=0
      if(pw(30).gt.0.0.and.ny(30).eq.1.and.nl.eq.11) kbor=1
      do 401 i=1,50
      xm(i)=0.0
      xmm(i)=0.0
      pm(i)=0.0
      xc(i)=0.0
      c(i)=0.0
      xa(i)=0.0
      a(i)=0.0
  401 continue
      kkwat=0
      tpw=0.0
      txm=0.0
      tpm=0.0
      txa=0.0
      toc=0.0
```

```
      oxeq=0.0
      anno=0.0
c                                          Molar and atom proportions
      do 405 i=1,nox
      if(pw(i).eq.0.0) go to 405
      tpw=tpw+pw(i)
      xm(i)=pw(i)/wm(i)
  405 continue
      if(kloi.eq.0) xm(13)=0.0
c                                       F and Cl in molar proportions
      ho=0
      q=pw(28)+pw(29)
      if(q.lt.1.e-6) go to 406
      qho=(pw(28)*0.4210723)+(pw(29)*0.2256424)
      qhot=tpw-qho
      ho=1
  406 continue
c                                  Structural formula molar proportions
      do 412 i=1,nox
      if(i.lt.14.and.i.gt.10) go to 412
      if(ndel(i).eq.0) xmm(i)=xm(i)
  412 continue
c                                           Iron redox redistribution
      if(xm(4).eq.0.0.and.xm(5).eq.0.0) go to 413
      if(xiron.gt.1.0) go to 413
      x2=xm(5)
      x3=xm(4)*2.0
      q=x2+x3
      xmm(5)=xiron*q
      xmm(4)=0.0
      if(xiron.lt.1.00) xmm(4)=(q-xmm(5))/2.0
  413 continue
c                Redistribution of H2O+, H2O-, LOI as formula components
      do 416 i=1,2
      ii=i+10
      do 415 k=1,3
      kk=k+10
      if(ndel(kk).ne.0) go to 415
      if(iww(k).eq.i) xmm(ii)=xmm(ii)+xm(kk)
  415 continue
  416 continue
c                                 Atom proportions in structural formula
      do 417 i=1,nox
      if(xm(i).eq.0.0.and.xmm(i).eq.0.0) go to 417
      if(i.ne.28.and.i.ne.29) txm=txm+xm(i)
      xc(i)=xmm(i)*xnc(i)
  417 continue
      scat=0.0
      swat=xc(11)+xc(28)+xc(29)
      do 418 i=1,nox
      if(xm(i).ne.0.0) pm(i)=(xm(i)/txm)*100.0
      if(i.ne.28.and.i.ne.29) tpm=tpm+pm(i)
      if(ny(i).gt.2) go to 418
```

```
      if(xc(i).eq.0.0) go to 418
      scat=scat+(xc(i)*(xnn(i)/(2.0*xnc(i))))
      if(ny(i).lt.2) toc=toc+xc(i)
      if(ny(i).eq.2.and.ncats.eq.1) toc=toc+xc(i)
  418 continue
c                                   Atom abundances in structural formula
      xsite=scat+(0.5*swat)
      if(nform.eq.0) xsite=scat
      if(nform.eq.2) xsite=toc
      if(nform.ne.2) qq=xneg
      if(nform.eq.2) qq=xtoc
      do 419 i=1,nox
      if(xc(i).eq.0.0) go to 419
      c(i)=(xc(i)*qq)/xsite
  419 continue
      if(xc(11).eq.0.0) go to 440
      if(iwat.eq.0) go to 440
      if(iwat.eq.2) go to 432
c                                       Structural formula with H3O+
      if(nform.ne.2) go to 421
      if(ncats.eq.1) go to 421
      toh=c(11)+c(28)+c(29)
      if(toh.le.xwat) go to 440
      q=toh-xwat
      if(q.gt.c(11)) q=c(11)
      c(13)=q/3.0
      c(11)=c(11)-q
      if(c(13).gt.0.0) kkwat=1
      go to 440
  421 continue
      if(xwat.ne.0.0) go to 422
      xc(13)=xc(11)/3.0
      xc(11)=0.0
      xsite=xsite+(0.5*xc(13))
      if(nform.eq.2) xsite=xsite+(0.5*xc(13))
      if(xc(13).gt.0.0) kkwat=1
      go to 440
  422 xrat=xneg/xwat
       if(nform.eq.2) xrat=xtoc/xwat
      toh=xc(11)+xc(28)+xc(29)
      yrat=xsite/toh
      if(yrat.ge.xrat) go to 440
      do 425 i=1,3
      ta(i)=0.0
      do 424 k=1,3
  424 tx(k,i)=0.0
  425 continue
      ta(1)=xsite
      if(nform.eq.1) ta(1)=ta(1)-(0.5*xc(11))
      ta(2)=xc(28)+xc(29)
      ta(3)=xc(11)
      tx(1,1)=1.0
      tx(1,2)=-0.5
```

```
      if(nform.eq.2) tx(1,2)=-1.0
      if(nform.eq.1) tx(1,3)=-0.5
      tx(2,1)=xwat/xneg
      tx(2,3)=-1.0
      tx(3,2)=3.0
      tx(3,3)=1.0
      call array(2,3,3,3,3,ty,tx)
      do 426 i=1,9
      if(i.lt.4) adp(i)=dble(ta(i))
      xdp(i)=dble(ty(i))
  426 continue
      call dgelg(adp,xdp,3,1,1.e-10,ier)
      if(ier.ne.0) go to 427
      go to 428
  427 write(1,955) ier
      go to 305
  428 continue
      do 429 i=1,3
      ta(i)=sngl(adp(i))
  429 continue
      q=(ta(2)*xneg)/ta(1)
      if(q.lt.1.e-6.and.q.gt.-1.e-6) ta(2)=0.0
      if(ta(2).lt.0.0) go to 427
      if(ta(2).eq.0.0) go to 440
      if(ta(3).ge.0.0) go to 431
      xc(13)=xc(11)/3.0
      xc(11)=0.0
      if(nform.lt.2) xsite=xsite+(0.5*xc(13))
      if(nform.eq.1) xsite=xsite-(0.5*3.0*xc(13))
      if(nform.eq.2) xsite=xsite+xc(13)
      if(xc(13).gt.0.0) kkwat=1
      go to 420
  431 xsite=ta(1)
      xc(13)=ta(2)
      xc(11)=ta(3)
      if(xc(13).gt.0.0) kkwat=1
  420 continue
      do 430 i=1,nox
      if(xc(i).ne.0.0) c(i)=(xc(i)*xneg)/xsite
  430 continue
      go to 440
c                                        Excess OH as interlayer H2O
  432 continue
      toh=c(11)+c(28)+c(29)
      if(toh.le.xwat) go to 440
      if(nform.eq.1) go to 433
      q=xwat-(c(28)+c(29))
      if(q.lt.0.0) q=0.0
      if(c(11).gt.q) c(12)=c(12)+(0.5*(c(11)-q))
      if(c(11).gt.q) kkwat=1
      if(c(11).gt.q) c(11)=q
      go to 440
  433 toh=xc(11)+xc(28)+xc(29)
```

```
      q=xwat/(xneg-(0.5*xwat))
      qa=q*scat
      q=toh-qa
      if(q.le.0.0) go to 440
      if(q.gt.xc(11)) q=xc(11)
      if(q.gt.0.0) kkwat=1
      xc(12)=xc(12)+(0.5*q)
      xc(11)=xc(11)-q
      xsite=scat+(0.5*(xc(11)+xc(28)+xc(29)))
      qq=xneg
      go to 420
  440 continue
      if(ho.eq.0) go to 442
      xfs=0.0
      xcs=0.0
      if(pm(28).gt.0.0) xfs=pm(28)/4.0
      if(pm(29).gt.0.0) xcs=pm(29)/4.0
  442 continue
      do 444 i=1,nox
      if(ny(i).gt.2) go to 444
      if(c(i).eq.0.0) go to 444
      oxeq=oxeq+(c(i)*(xnn(i)/(2.0*xnc(i))))
  444 continue
      q=c(11)+c(28)+c(29)
      anno=oxeq+(0.5*q)
      if(q.ne.0.0) go to 445
      if(nfor.eq.0) anno=0.0
      if(norm.gt.0) oxeq=0.0
  445 continue
      if(nform.eq.0) oxeq=0.0
      if(nform.eq.1) anno=0.0
      if(lr(15).eq.ly) go to 600
      if(nl.lt.1) go to 600
c
c          Cation distribution in clay-mineral structural formula
c          ------------------------------------------------------
c
c                                                Tetrahedral layer cations
      stot(1) = 0.0
      n = 0
      cz(1)=0.0
      cz(2)=0.0
      cz(3)=0.0
      do 501 i=41,47
      c(i)=0.0
      n=n+1
      if(nt(n).eq.0) go to 501
      xnn(i)=xnn(nt(n))
      xnc(i)=xnc(nt(n))
      lc(i)=lc(nt(n))
  501 continue
  503 continue
      do 508 i=1,7
```

```
      if(nt(i).eq.0) go to 508
      if(ny(nt(i)).ne.0) go to 508
      if(c(nt(i)).eq.0.0) go to 508
      m=i+40
      c(m)=c(nt(i))
      c(nt(i))=0.0
      stot(1)=stot(1)+c(m)
      cz(1)=cz(1)+((xnn(m)/xnc(m))*c(m))
  508 continue
      qq=xtet-stot(1)
      if(qq.lt.0.0) go to 511
      n=0
      do 507 i=41,47
      if(stot(1).ge.xtet) go to 511
      n=n+1
      if(nt(n).eq.0) go to 507
      if(ny(nt(n)).eq.0) go to 507
      c(i)=c(nt(n))
      q=qq-c(i)
      if(q.lt.0.0) go to 504
      c(nt(n))=0.0
      qq=qq-c(i)
      go to 505
  504 continue
      c(i)=qq
      c(nt(n))=c(nt(n))-c(i)
  505 continue
      cz(1)=cz(1)+(c(i)*(xnn(i)/xnc(i)))
      stot(1)=stot(1)+c(i)
  507 continue
  511 cz(1)=cz(1)-(4.0*xtet)
c               Octahedral and interlayer cations, anions, and molecules
      do 514 i=2,5
  514 stot(i)=0.0
      do 521 i=1,nox
      if(c(i).eq.0.0) go to 521
      if(ny(i).ne.1) go to 515
      if(i.eq.1.and.n1.eq.11) go to 521
      if(i.eq.30.and.kbor.eq.1) go to 521
      stot(2)=stot(2)+c(i)
      cz(2)=cz(2)+(c(i)*(xnn(i)/xnc(i)))
      go to 521
  515 continue
      if(ny(i).ne.2) go to 516
      stot(3)=stot(3)+c(i)
      cz(3)=cz(3)+(c(i)*(xnn(i)/xnc(i)))
      go to 521
  516 continue
      if(ny(i).ne.3) go to 517
      stot(4)=stot(4)+c(i)
      go to 521
  517 continue
      if(ny(i).ne.4) go to 521
```

```
      stot(5)=stot(5)+c(i)
  521 continue
      if(nform.eq.2) cz(2)=0.0-(cz(1)+cz(3))
      if(nform.eq.2) go to 600
      q=xneg*2.0
      if(nform.eq.1) q=q-(c(11)+c(28)+c(29))
      cz(2)=cz(2)-(q-(4.0*xtet))
c
```

Formatting results of calculation

```
c*********************************************************************
c                    FORMATTING RESULTS OF CALCULATIONS
c*********************************************************************
c
c                         Table of analytical data
c                         ------------------------
c
  600 continue
      do 601 i=1,99
      ha(i)=habl
  601 hf(i)=hfbl
      maxa=0
      maxl=0
      mm=0
c                                                   Chemical analysis data
      write(ha(2),9200)
      write(ha(4),9201) lcap(1),lcap(2)
      write(ha(5),9201) lcap(3),lcap(3)
      k=0
      do 603 i=6,99
  602 k=k+1
      iii=0
      if(k.gt.nox) go to 605
      if(pw(io(k)).eq.0.0.and.pm(io(k)).eq.0.0) go to 602
      if(io(k).ne.28.and.io(k).ne.29) write(ha(i),9203)
     -lox(io(k)),pw(io(k)),pm(io(k))
      if(io(k).eq.28.or.io(k).eq.29) write(ha(i),9204)
     -lox(io(k)),pw(io(k))
  603 iii=1
  605 i=i+iii
      write(ha(i),9201) lcap(3),lcap(3)
      i=i+1
      write(ha(i),9205) lcap(4),tpw,tpm
      i=i+1
      if(ho.eq.0) go to 607
      write(ha(i),9206) qho
      i=i+1
      write(ha(i),9207) qhot
      i=i+1
  607 maxa=i-1
c                                       Generalized structural formula
      if(lr(15).eq.ly) go to 645
```

```
      if(nl.gt.0) go to 620
      sc=0.0
      write(hf(1),9211)
      if(nform.eq.0) write(hf(2),9212) lneg,nneg
      if(nform.eq.1) write(hf(2),9210) lsit,nneg
      if(nform.eq.2.and.ncats.eq.0) write(hf(2),9216) ltoc,ntoc
      if(nform.eq.2.and.ncats.eq.1) write(hf(2),9217) lctn,ntoc
      i=5
      nq=0
  610 continue
      do 612 k=1,nox
      if(ny(io(k)).ne.nq) go to 612
      if(c(io(k)).eq.0.0) go to 612
      if(ny(io(k)).lt.3) sc=sc+c(io(k))
      write(hf(i),9213) lc(io(k)),c(io(k))
      i=i+1
  612 continue
      if(nq.ne.2) go to 615
      write(hf(i),9214) lcap(3)
      i=i+1
      write(hf(i),9215) sc
      i=i+2
  615 nq=nq+1
      if(nq.lt.5) go to 610
      maxl=i
      go to 645
c                                    Phyllosilicate structural formula
  620 continue
      if(nl.gt.10) write(hf(1),9220)
      if(nl.lt.11) write(hf(1),9221)
      if(nl.lt.11) write(hf(2),9222) llt(nl)
      k=3
      if(nl.gt.10) k=2
      if(nform.eq.0) write(hf(k),9223) lneg,nneg
      if(nform.eq.1) write(hf(k),9224) lsit,nneg
      if(nform.eq.2.and.ncats.eq.0) write(hf(k),9228) ltoc,ntoc
      if(nform.eq.2.and.ncats.eq.1) write(hf(k),9229) lctn,ntoc
      ii=k+2
      if(nl.lt.11.or.ny(1).ne.1.or.c(1).eq.0.0) go to 622
      write(hf(ii),9225) lbl,lc(1),c(1)
      isil=1
      c(1)=0.0
      ii=ii+2
  622 il=1
      kz=47
      iz=0
      k=40
  621 continue
      if(stot(il).eq.0.0) go to 630
      if(il.lt.5) lll=lcat(il)
      if(il.eq.3.and.nl.gt.7) lll=lbig
      lines=0
      if(nl.gt.10) lll=lbl
```

```
      do 625 i=ii,99
      ic=0
      iii=0
  623 k=k+1
      if(k.gt.kz) go to 627
      if(k.gt.40) io(k)=k
      if(il.gt.1.and.ny(io(k)).ne.iz) go to 623
      if(c(io(k)).eq.0.0) go to 623
      if(il.eq.2.and.io(k).eq.30.and.kbor.eq.1) go to 623
      write(hf(i),9225) lll,lc(io(k)),c(io(k))
      lines=lines+1
      lll=lbl
  625 iii=1
  627 i=i+iii
      if(il.eq.5) go to 629
      if(lines.lt.2) go to 628
      write(hf(i),9226) lcap(4),stot(il)
      i=i+1
  628 continue
      if(nl.ne.ll.or.il.ne.1.or.kbor.ne.1) go to 629
      i=i+1
      write(hf(i),9225) lll,lc(30),c(30)
      ibor=1
      i=i+1
  629 i=i+1
  630 continue
      il=il+1
      k=0
      kz=nox
      iz=iz+1
      if(il.ne.4) go to 635
      if(nl.gt.7) go to 632
      ii=i
      do 631 i=ii,99
      k=k+1
      iii=0
      if(k.gt.3) go to 632
      if(k.eq.3.and.cz(k).eq.0.0) go to 632
      write(hf(i),9227) lcz(k),cz(k)
  631 iii=0
  632 k=0
      i=i+iii
      q=stot(4)+stot(5)
      if(q.eq.0.0) go to 640
      if(nl.lt.ll) i=i+1
  635 continue
      if(il.eq.6) go to 640
      if(il.eq.5.and.stot(5).eq.0.0) go to 640
      ii=i
      go to 621
  640 maxl=i-1
  645 maxx=maxa
      if(maxl.gt.maxa) maxx=maxl
```

```
  650 continue
c
c                          Footnotes
c                          ---------
c
      nf=0
c                                                  Analytical data altered
      if(nofile.eq.0) go to 652
      nf=nf+1
      mm=mm+1
      write(a79(mm),9659) nf
c                                                      Halides in analysis
  652 continue
      if(ho.eq.0) go to 655
      if(xfs.lt.1.e-5) go to 654
      nf=nf+1
      mm=mm+1
      write(a79(mm),9651) nf,xfs
  654 continue
      if(xcs.lt.1.e-5) go to 655
      if(mm.eq.0) nf=nf+1
      mm=mm+1
      if(mm.eq.1) write(a79(mm),9652) nf,xcs
      if(mm.eq.2) write(a79(mm),9653) xcs
c                                                         Iron character
  655 continue
      if(qiron.eq.0.0) go to 658
      if(lr(7).eq.ln) go to 657
      nf=nf+1
      mm=mm+1
      write(a79(mm),9655) nf
  657 continue
      if(lr(15).eq.ly) go to 658
      if(xiron.gt.1.1) go to 658
      nf=nf+1
      mm=mm+1
      write(a79(mm),9656) nf,xiron
c                                  Treatment of LOI in chemical analysis
  658 continue
      if(pw(13).eq.0.0) go to 660
      nf=nf+1
      mm=mm+1
      if(kloi.ne.0) write(a79(mm),9657) nf
      if(kloi.eq.0) write(a79(mm),9658) nf
c                                  Water character in structural formula
  660 continue
      if(lr(15).eq.ly) go to 690
      if(qwat.eq.0.0) go to 665
      nf=nf+1
      mm=mm+1
      write(a79(mm),9661) nf
      do 661 i=1,3
      ii=i+10
```

```
      if(pm(ii).eq.0.0) go to 661
      mm=mm+1
      write(a79(mm),9663) lox(ii),lwat(iww(i))
  661 continue
      if(iwat.eq.0) go to 664
      if(kkwat.eq.0) go to 664
      mm=mm+1
      if(iwat.eq.1) write(a79(mm),9664)
      if(iwat.eq.2) write(a79(mm),9665)
  664 continue
c                              Components deleted from structural formula
  665 continue
      if(hdel.eq.0) go to 670
      nq=0
      do 666 i=1,nox
      if(i.lt.30) a5(i)=a5b
      if(ndel(i).eq.0) go to 666
      if(xm(i).eq.0.0) go to 666
      nq=nq+1
      a5(nq)=lox(i)
  666 continue
      if(nq.eq.0) go to 670
      nf=nf+1
      mm=mm+1
      write(a79(mm),9668) nf,(a5(k),k=1,5)
      if(nq.lt.6) go to 670
      mm=mm+1
      write(a79(mm),9669) (a5(k),k=6,17)
      if(nq.lt.18) go to 670
      mm=mm+1
      write(a79(mm),9669) (a5(k),k=18,29)
  670 continue
c                                     Tetrahedral sites per formula unit
      if(nl.eq.0) go to 672
      nf=nf+1
      mm=mm+1
      write(a79(mm),9673) nf,ntet
c                                       (OH,F,Cl) sites per formula unit
  672 continue
      if(kkwat.eq.0) go to 675
      mm=mm+1
      if(nl.eq.0) nf=nf+1
      if(nl.ne.0) write(a79(mm),9674) nwat
      if(nl.eq.0) write(a79(mm),9678) nf,nwat
  675 continue
c                                 Anion character in structural formula
      n=0
      if(oxeq.ne.0.0) n=n+1
      if(anno.ne.0.0) n=n+1
      if(n.eq.0) go to 680
      if(anno.eq.0.0) go to 678
      mm=mm+1
      nf=nf+1
```

```
      write(a79(mm),9675) nf,anno
  678 continue
      if(oxeq.eq.0.0) go to 680
      if(n.eq.1) nf=nf+1
      mm=mm+1
      if(n.eq.1) write(a79(mm),9676) nf,oxeq
      if(n.eq.2) write(a79(mm),9677) oxeq
c                                   Cation site assignment in clay formula
  680 nzzz=0
      nitet=0
      kotet=0
      if(nl.eq.0) go to 690
      nitet=0
      i=0
      k=-1
 6801 k=k+2
      if(k.eq.2.or.k.gt.3) k=0
 6802 i=i+1
      if(i.gt.7) go to 6803
      if(nt(i).eq.k) go to 6801
      if(nt(i).eq.0) go to 6802
      if(xc(nt(i)).eq.0.0) go to 6802
      nitet=1
 6803 continue
      if(k.ne.0) nitet=1
      do 6805 i=1,nox
      if(xc(i).eq.0.0) go to 6805
      if(nz(i).ne.0) nzzz=1
 6805 continue
      do 6807 i=1,7
      if(nt(i).eq.1) go to 6806
      if(xc(nt(i)).gt.0.0.and.ny(nt(i)).eq.0) kotet=1
      go to 6807
 6806 continue
      if(ny(1).ne.0.and.xc(1).gt.0.0) kotet=1
 6807 continue
      if(xc(1).gt.0.0.and.ny(1).ne.0) kotet=1
 6808 nzzt=nzzz+nitet+kotet
      if(nzzt.eq.0) lras=lraa(2)
      if(nzzt.ne.0) lras=lraa(1)
      ii=1
      iii=2
      nf=nf+1
      mm=mm+1
      write(a79(mm),9681) nf,lt7,lcas,lras
      if(nzzt.eq.0) go to 690
      nq=0
      if(kotet.eq.0) go to 683
      do 6823 i=1,7
      a4(i)=a4b
      if(xc(nt(i)).eq.0.0.or.ny(nt(i)).ne.0) go to 6823
      nq=nq+1
      a4(nq)=lc(nt(i))
```

```
 6823 continue
      mm=mm+1
      if(nq.eq.0) a4(1)=none
      write(a79(mm),9684) (a4(i),i=1,7)
  683 nq=0
      if(itet.eq.0) go to 684
      do 681 i=1,7
      a4(i)=a4b
      if(nt(i).eq.0) go to 681
      if(xc(nt(i)).eq.0.0) go to 681
      nq=nq+1
      a4(nq)=lc(nt(i))
  681 continue
  682 mm=mm+1
      write(a79(mm),9685) (a4(k),k=1,7)
  684 nq=0
      if(nzzz.eq.0) go to 690
      do 685 i=1,nox
      if(i.lt.31) a4(i)=a4b
      if(xc(i).eq.0.0) go to 685
      if(nz(i).eq.0) go to 685
      if(ny(i).ne.ii) go to 685
      nq=nq+1
      a4(nq)=lc(i)
  685 continue
      if(nq.eq.0) go to 687
      nqq=11
      nqqq=20
      mm=mm+1
      write(a79(mm),9682) ltaa(iii),(a4(k),k=1,10)
  686 continue
      if(nq.lt.nqq) go to 687
      mm=mm+1
      write(a79(mm),9683) (a4(k),k=nqq,nqqq)
      nqq=nqq+10
      nqqq=nqqq+10
      if(nqq.lt.22) go to 686
  687 ii=ii+1
      iii=iii+1
      if(iii.eq.3.and.nl.gt.7) iii=4
      if(ii.lt.3) go to 684
      mm=mm+1
      if(ibor.eq.1.and.isil.eq.0) write(a79(mm),9686)
      if(ibor.eq.0.and.isil.eq.1) write(a79(mm),9687)
      if(ibor.eq.1.and.isil.eq.1) write(a79(mm),9688)
  690 continue
c
```

Printing out sample analysis and structural formula

```
c*********************************************************************
c                      PRINT-OUT OF SAMPLE ANALYSIS
c*********************************************************************
c
      if(lr(11).eq.ln.and.lr(12).eq.ln) lfile=inpt
      if(kal.ne.0) go to 705
  701 write(1,951)
      read(1,991,err=701) nouts
      if(nouts.lt.0.or.nouts.gt.2) go to 701
      if(nouts.lt.1) go to 705
      do 702 i=1,999
      write(lform,8541) i
      inquire(file=lform,exist=mol)
      if(mol) go to 702
      open(15,status='new',file=lform,err=702)
      write(1,8571) lform
      go to 703
  702 continue
  703 write(1,8575)
      call yesno(la,nerr)
      if(nerr.ne.0) go to 703
      ncarr=0
      if(la.eq.ly) ncarr=1
  705 m=1
      if(nouts.eq.1) m=15
      if(nono.eq.0) go to 706
      if(nono.eq.2) nono=0
      if(m.eq.1) write(1,9511)
      if(m.eq.1)  read(1,990,err=705) la
  706 kal=1
      kout=maxx+mm+5
      if(m.eq.15.and.ncarr.eq.1) write(m,9512)
      if(m.eq.1) ii=66
      if(m.eq.15) ii=55
      if(m.eq.15.and.ncarr.eq.1) ii=ii-2
      nskip=ii-kout
      if(nskip.le.0) go to 710
      do 707 i=1,33
      k=2*i
      if(k.ge.nskip) go to 708
  707 continue
  708 kpre=i
      kpost=i
      if(k.gt.nskip) kpre=kpre-1
      if(kpre.lt.1) go to 710
      do 709 i=1,kpre
  709 write(m,997)
  710 continue
      write(m,952) ltit,ldes
  712 continue
      do 715 i=1,maxx
```

```
      write(m,9231) ha(i),hf(i)
  715 continue
      if(lfile.ne.inpt) write(m,960) nal,lfile
      if(lfile.eq.inpt) write(m,961) nal
      do 720 i=1,mm
  720 write(m,9235) a79(i)
      if(m.eq.15.and.ncarr.eq.1) go to 740
      if(nskip.le.0) go to 740
      do 730 i=1,kpost
  730 write(m,997)
  740 continue
      if(nouts.ne.2.or.m.ne.1.or.nono.eq.0) go to 745
      write(1,8577)
      call yesno(la,nerr)
      if(nerr.ne.0) go to 740
      if(la.eq.ln) go to 755
  745 continue
      if(nouts.lt.2.or.m.eq.15) go to 755
      m=15
      go to 706
c
```

Program continuation or termination

```
c*********************************************************************
c                 PROGRAM CONTINUATION OR TERMINATION
c*********************************************************************
c
  755 continue
      nq=0
      if(lr(14).eq.ly.or.lr(15).eq.ly) go to 770
      if(nono.eq.0) go to 770
      go to 765
  761 write(1,989)
  762 nq=8
      if(lr(11).eq.ly) close(12)
      if(lr(12).eq.ly) close(13)
      lr(11)=ln
      go to 774
c                                   Other parameters for this analysis?
  765 write(1,985)
      call yesno(la,nerr)
      if(nerr.ne.0) go to 765
      if(la.eq.ln) go to 770
      nq=1
c                           Change any component abundances in analysis?
  767 write(1,988)
      call yesno(la,nerr)
      if(nerr.ne.0) go to 767
      if(la.eq.ly) nq=2
      go to 780
  770 continue
      if(lr(12).eq.ly) nq=4
```

```
      if(lr(12).eq.ly) go to 780
c                                        Another analysis in this group?
  772 write(1,986)
      call yesno(la,nerr)
      if(nerr.ne.0) go to 772
      if(la.eq.ln) nq=8
      if(la.eq.ly) go to 780
c                                            Another group of analyses?
  774 write(1,984)
      call yesno(la,nerr)
      if(nerr.ne.0) go to 774
      if(la.eq.ln) nq=16
  780 continue
      m=1
      nofile = 0
      if(nq.eq.2) nofile = 1
      if(nq.eq.0) go to 200
      if(nq.eq.1) go to 305
      if(nq.eq.2) go to 238
      if(nq.eq.4) go to 180
      if(nq.gt.7.and.lr(11).eq.ly) go to 270
      if(nq.eq.8) go to 100
  798 continue
      if(nouts.gt.0) close(15)
  799 continue
      stop
c
```

Format statements

```
c*********************************************************************
c                          FORMAT STATEMENTS
c*********************************************************************
c
 8001 format(18x,'STRUCTURAL FORMULA SPECIFICATIONS')
 8002 format('1 - ',a40,':')
 8003 format(6x,a5,a34)
 8004 format(4x,'Excess OH recast as H3O+.')
 8005 format(4x,'Excess OH recast as H2O.')
 8006 format(a28,f5.3,'. ')
 8007 format('2 - ',a35,a35)
 8008 format('3 - ',a47,'None.')
 8009 format('3 - ',a47,4(a5,1x),a5)
 8010 format(4x,15a5)
 8012 format('4 - ',a29,a40)
 8013 format(6x,a37,i3,'.')
 8014 format(6x,a36,i2,'.')
 8015 format('6 - ',a25,a29,a20)
 8017 format(/,5x,'Enter ID (1-dig. int.) of specification to modify:',
     -/,10x,'["0" = All specifications satisfactory.]',/,10x,
     -'["7" = Print summary of specifications.]')
 8018 format(6x,a22,10(a4,1x))
 8019 format(28x,10(a4,1x))
```

```
8020 format(6x,a30,i2,'.')
8021 format(6x,a49,i2,'.')
8022 format(6x,a32,i2,'.')
8025 format('5 - Parameters of structural formula:')
8026 format(6x,a34,i2,'.')
8028 format('1 - Water options',12x,'3 - Deletions',15x,
    -'5 - Formula parameters',/,'2 - Iron redox',15x,
    -'4 - Formula type')
8029 format('1 - Water options',12x,'3 - Deletions',15x,
    -'5 - Formula parameters',/,'2 - Iron redox',15x,
    -'4 - Formula type',12x,'6 - Site assignments')
8030 format(6x,'Must be tetrahedral: ',7(a4,1x))
8031 format('4 - ',a29,'General (cation sites undesignated).')
8032 format('4 - ',a29,'General (cation sites designated).')
8210 format('Elect character ID (1-dig. int.) for ',a5)
8211 format(5x,i1,' - ',a34)
8212 format('Disposition of excess OH (enter 1-dig. int. ID):'
    -,/,5x,'"0" - Retain as OH.',/,5x,
    -'"1" - Recast as H3O+ ion in interlayer sites.',/,5x,
    -'"2" - Recast as H2O (hydration waters).')
8213 format('Enter # of (OH,F,Cl) sites in formula (2-dig. int.):')
8220 format('Iron redox ratio other than in analysis? (y/n):')
8221 format('Enter value for Fe2+/[Fe2+ + Fe3+] atom ratio (0.0-1.0):')
8222 format(a28,f6.4,5x,'OK? (y/n):')
8230 format('Select one option (1-dig. int.):',/,5x,
    -'1 - Remove all previous deletions.',/,5x,
    -'2 - Remove previous deletions and assign new ones.',/,5x,
    -'3 - Retain previous deletions but modify the list.')
8231 format('Selection of specific individual modifications '
    -,'(enter 1-dig. int.)'
    -/,5x,'"0" = From list of only components in analysis.',/,5x,
    -'"1" = From list of all components in program.')
8232 format(8(' '))
8233 format(i2.2,1x,a5)
8234 format(8(a1,a8,1x))
8235 format('Enter ID (2-dig. int.) for components to be deleted:')
8236 format(3x,'[If starred ("*") ID entered; previous deletion ',
    -'instruction removed.]')
8237 format(20('-- '))
 824 format('Clay-mineral formula type?  Enter 2-dig. int. ID:',
    -10(/,5x,i2,2x,a40))
8240 format('Type of formula (enter 1-dig. int.):',/,5x,
    -'"1" = General silicate-mineral structural formula.',/,5x,
    -'"2" = Phyllosilicate (clay-mineral) formula.')
8241 format('Enter type (1-dig. int.) of phyllosilicate formula:')
8242 format('Component distribution into formula sites?  (y/n):')
8243 format('Mineral dioctahedral ("n"=trioctahedral)?  (y/n):')
8244 format('Tet-oct ("0") or all ("1") cations?  ',
    -'(Enter 1-dig. int.):')
8250 format('Method for calculating structural formula (enter ',
    -'1-dig. int.):',/,5x,'"0" - Oxygen equivalents (anion ',
    -'compnents excluded) per formula unit.'
    -/,5x,'"1" - Number of anion sites (O,OH,F,Cl) per formula unit.'
```

```
    -,/,5x,'"2" - Number of cations per formula unit.')
8251 format('Enter (2-dig. int.) new value:')
8252 format('Double the formula charge/sites? (y/n):')
8253 format(a37,i2)
8254 format(a30,i2)
8255 format(a49,i2)
8256 format(a36,i2)
8257 format(a34,i2)
8258 format(a32,i2)
8260 format(10x,'STRUCTURAL FORMULA PARAMETERS')
8270 format('Site assignment change desired (enter 1-dig. int.):',
    -4(/,5x,i1,1x,a29))
8271 format('Any reassignment of specific cations? (y/n):')
8272 format('        ')
8273 format(a1,i2.2,1x,a4)
8274 format(/,'LISTING OF CATION SITE ASSIGNMENTS',/,
    -'(Tetrahedral listing is order of filling tetrahedral sites)')
8275 format(20x,a8,8x,a8,8x,a8)
8276 format('* Specifically user-reassigned to this site.')
8277 format('Identify (1-dig. int.) type of modification desired:'
    -,/,5x,'"0" = No change.',/,5x,
    -'"1" = Reassign to octahedral sites.',/,5x,
    -'"2" = Reassign to interlayer sites.',/,5x,
    -'"3" = Order of filling tetrahedral sites.')
8279 format(5x,'"4" = Remove individual non-default site assignments.')
8278 format('Retain non-default cation site assignments? (y/n)',
    -/,2x,['"No" returns all cations to default assignment.]')
8280 format('Enter ID (2-dig. int.) for reassignment to ',a1,
    -/,2x,'(An entry of "00" signifies no further changes)')
8281 format('Order of filling tetrahedral sites')
8282 format(5x,i1,'.  ',i2.2,1x,a4)
8283 format('Feasible tetrahedral cation ID''s')
8284 format(5x,i2.2,1x,a4)
8285 format('List ID''s (2-dig. int.) for order of filling tet. '
    -,'sites:',/,7('-- '))
8286 format(7(i2,1x))
8287 format('Enter ID (2-dig. int.) for return to default site ',
    -'assignment.')
8288 format(/,20x,2a16,'Big cations')
8289 format(/,20x,3a16)
8290 format('Same parameters for all samples in group?  (y/n):')
8293 format('Other ions that must be tetrahedral?  (y/n):')
8294 format('Enter ALL cation ID''s (2-dig. int.) that must ',
    -'be tetrahedral:',/,7('-- '))
 850 format(a1,a79)
 851 format(' ','Analysis on file? (y/n):')
 852 format(' ','A file of new analyses to be created? (y/n):')
 853 format(i3.3,2x,a75)
8531 format(' ','Data entry ONLY (no output)?  (y/n):')
 854 format('CLAY.',i3.3)
8541 format('FORM.',i3.3)
 855 format(' ','Enter title of this group of analyses ',
    -'(up to 75-char. string)')
```

```
 856 format(a8)
 857 format(' ','New file named: ',a8)
8571 format(' ','Output file named: ',a8)
8575 format('Output file with FORTRAN carraige control?  (y/n):')
8577 format('Into the output file?  (y/n):')
 858 format(' ','Enter 3-dig. int. to identify desired CLAY.nnn file:',
    -/,3x,'[An entry of 000 lists titles of CLAY.nnn files.]')
 859 format(' ',a8,' NOT FOUND.  Try again (y/n):')
 861 format(' ','A listing of samples in file? (y/n):')
 863 format(' ','FILE SAMPLE # ',i2.2,/,a80)
 864 format(' ','Want just designated samples in file? (y/n):')
 865 format(' ','Enter up to 10 integers (2-digit) ',
    -'designating samples desired:',/,10('--',1x))
 866 format(10(i2,1x))
8662 format(20(i2,1x))
 869 format(' ',10x,'SAMPLE DESCRIPTIONS: FILE ',a8,/)
 872 format(a1,1x,6(1x,i2.2,1x,e9.4))
 873 format(a1,1x,a5,1x,a4,f9.5,1x,3i1)
 874 format(a1,1x,a1)
 875 format('Disposition of LOI (loss-on-ignition)?  (1-dig. int.)'
    -,/,5x,'"0" - LOI omitted from all calculations.',/,5x,
    -'"1" LOI calculated as molar H2O.')
 876 format(6x,'LOI (loss-on-ignition) omitted from all calculations.')
 877 format(6x,'LOI (loss-on-ignition) recast as molar H2O.')
 881 format(a5,a4,f9.5,3i1)
 891 format('Tetrahedral     Octahedral      Interlayer      ',
    -'Anions/waters   Layer charge    ')
 892 format(5a16)
 901 format(' ','Enter oxide wt. %.')
 902 format(' ',2x,a5,':')
 903 format(f9.6)
 904 format(' ','Any other oxides? (y/n):')
 905 format(' ','Enter oxide ID# and wt.%:')
 906 format(' ',5(4x,i2.2,2x,a5))
 907 format(2('-'),1x,9('-'))
 908 format(i2,1x,f9.6)
 909 format(' ','Any other (unlisted) components? (y/n):')
 911 format(' ','Molecular formula (up to 5-char. string):')
 912 format(a5)
 913 format(' ','Formula atom (up to 4-char. string):')
 914 format(a4)
 915 format(' ','Molecular weight (with dec. pt.):')
 916 format(' ','Number of formula atoms per molecule (1 int.):')
 917 format(' ','Negative charges per molecule (1 int.):')
 918 format(' ','Site in clay structure (1 int. code ID)',/,4x,
    a '0=tet.',4x,'1=oct.',4x,'2=interlayer',4x,'3=anion',4x,
    -'4=molecule:')
 919 format(' ','Is total Fe expressed as single oxide? (y/n):')
 920 format(' ','Sample description (up to 80-char. string):')
9200 format(5x,'CHEMICAL ANALYSIS',6x)
9201 format(1x,8x,a8,2x,a8,1x)
9203 format(1x,a5,3x,f8.4,2x,f8.4,1x)
9204 format(1x,a5,3x,f8.4,10x,1x)
```

```
9205 format(a8,1x,f8.4,2x,f8.4,1x)
9206 format(1x,'O=F+Cl',2x,f8.4,10x,1x)
9207 format('TOTAL',4x,f8.4,10x,1x)
 921 format(' ',//,10x,'CHECK OF DATA ENTRY')
9210 format(14x,a32,i2,3x)
9211 format(16x,'GENERALIZED STRUCTURAL FORMULA',5x)
9212 format(11x,a38,i2)
9213 format(27x,a4,2x,f8.5,10x)
9214 format(33x,a8,10x)
9215 format(17x,'Total cations',3x,f8.5,10x)
9216 format(11x,a38,i2)
9217 format(14x,a32,i2,3x)
 922 format(' ',24x,i2.2,1x,a5,2x,f8.5)
9220 format(13x,'SILICATE MINERAL STRUCTURAL FORMULA',3x)
9221 format(15x,'CLAY-MINERAL STRUCTURAL FORMULA',5x)
9222 format(10x,a41)
9223 format(10x,a39,i2)
9224 format(13x,a33,i2,3x)
9225 format(10x,a16,2x,a4,1x,f8.5,10x)
9226 format(25x,a8,10x,f8.5)
9227 format(13x,a26,f9.5,3x)
9228 format(11x,a37,i2,1x)
9229 format(14x,a31,i2,4x)
 923 format(' ','Analysis entry OK? (y/n):')
9231 format(' ',a28,a51)
9232 format(28(' '))
9233 format(51(' '))
9235 format(' ',a79)
 924 format(' ','Enter component ID (2-dig.int.) and corrected value',
    a' (dec. pt.)',/,5x,'[ID:  00 = all OK;  98 = list ',
    a'all ID''s;  99 = re-enter whole analysis]')
 925 format(' ',/,a80,/,5x,'OK? (y/n):')
 926 format(' ','UNLISTED ANALYSIS MOLECULE CONSTANTS',/5x,
    a'Molecule formula = ',a5,10x,'Atom/radical = ',a4,5x,
    a/,5x,'Molecular weight = ',f9.5,10x,'Formula atoms in ',
    a'molecule = ',i1,/,5x,'Molecule negative charge = ',i1,
    a10x,'Assigned layer site = ',a16)
 944 format(' ','0 = No additional changes',14x,'1 = Reassign to',
    a' oct. sites',/,1x,'2 = Reassign to interlayer sites',7x,
    a'3 = Revise order of filling tet. sites',/,
    a'Enter desired integer code:')
 951 format(' ','Destination of output [one-digit entry]:',/,5x,
    -'0 - Terminal only',/,5x,'1 - Data file for line printer',
    -/,5x,'2 - Both of above')
9511 format(' ',23x,'SET PAPER IN CARRAIGE AND RETURN')
9512 format('1',10x)
 952 format(' ',4x,a75,/,a80,/,80('-'))
 955 format('Cannot compute H3O+!!  IER=',i2,'. Enter new parameters.')
 960 format(' ',53x,'Sample ',i2.2,' in file ',a8,/,80('-'))
 961 format(' ',46x,'Sample ',i2.2,' from direct data entry.'
    -,/,80('-'))
9651 format(i1,'. Mol % SiO2 includes ',f7.4,' mol % SiF4.',37x)
9652 format(i1,'. Mol % SiO2 includes ',f7.4,' mol % SiCl2.',36x)
```

```
9653 format(3x,'Mol % SiO2 includes ',f7.4,' mol % SiCl2.',36x)
9655 format(i1,'. Fe oxide in analysis is total Fe2+ + Fe3+.',32x)
9656 format(i1,'. Iron redox in structural formula set at ',
    -'Fe2+/[Fe2+ + Fe3+] = ',f6.4,'.',8x)
9657 format(i1,'. LOI (loss-on-ignition) calculated as molar H2O.',29x)
9658 format(i1,'. LOI (loss-on-ignition) omitted from all calculations.
    -',23x)
9659 format(i1,'. Chemical analysis altered from original data.')
9661 format(i1,'. Water character in structural formula set at:',31x)
9663 format(5x,a5,a34,35x)
9664 format(3x,' Excess OH sites recast as interlayer H3O+ ion.',36x)
9665 format(3x,' Excess OH sites recast as molecular H2O.',42x)
9668 format(i1,'. Components deleted from strucutral formula: ',
    -5(a5,1x),2x)
9669 format(5x,12(a5,1x),2x)
9673 format(i1,'. Number of tetrahedral sites per formula set at '
    -,i2,'.',26x)
9674 format(3x,'Number of (OH,F,Cl) sites per formula set at '
    -,i2,'.',28x)
9675 format(i1,'. Number of anions per formula unit =',f9.5,'.')
9676 format(i1,'. Number of oxygen equivalents per formula unit =',
    -f9.5,'.')
9677 format(3x,'Number of oxygen equivalents per formula unit =',
    -f9.5,'.')
9678 format(i1,'. Number of (OH,F,Cl) sites per formula unit set at '
    -,i2,'.')
9681 format(i1,'. ',a25,a29,a20,2x)
9682 format(5x,a22,10(a4,1x),2x)
9683 format(27x,10(a4,1x),2x)
9684 format(5x,'Must occupy tetrahedral sites:  ',
    -7(a4,1x),7x)
9685 format(5x,'Order of filling tetrahedral sites:  ',
    -7(a4,1x),2x)
9686 format(3x,'Default reassignment of B to unique sites.')
9687 format(3x,'Default reassignment of Si to unique sites.')
9688 format(3x,'Default reassignment of Si and B to unique sites.')
 984 format(' ','Another group (file) of analyses to enter? (y/n):')
 985 format(' ',////,1x,'Other parameters for this analysis? (y/n):')
 986 format(' ','Another analysis in this group to enter? (y/n):')
9871 format(' ','Only wt.% and mol.% in output (no struct. form.)',
    -'?  (y/n):')
 988 format(' ','Any modification of chemical analysis? (y/n):')
 989 format(' ',20x,'CANNOT READ DATA FILE!',/,16x,
    -'This file abandoned and closed.')
 990 format(a1)
 991 format(i1)
 992 format(i2)
9921 format(i3)
 994 format(' ','OK? (y/n):')
 997 format(' ',4x)
 998 format(a80)
 999 format(a75)
     end
```

```
c                                                                       2347
c***********************************************************************2348
```

Subroutine YESNO

```
c***********************************************************************0001
c                          SUBROUTINE YESNO                             0002
c***********************************************************************0003
c                                                                       0004
c     Allows main program to accept either one of the "y" and "n" pair  0005
c     or one of the "Y" and "N" pair in response to a yes-no question   0006
c     in the host program; program also flags an error for any entry    0007
c     other than "y", "Y", "n", or "N", and so informs the host program. 0008
c                                                                       0009
      subroutine yesno(lam,nerr)                                        0010
      character*1 lam,ly,ln,lyy,lnn                                     0011
      data ly/'y'/,ln/'n'/,lyy/'Y'/,lnn/'N'/                            0012
      nerr=0                                                            0013
      read(1,990,err=190) lam                                           0014
      if(lam.eq.lyy.or.lam.eq.lnn) go to 150                            0015
      if(lam.ne.ly.and.lam.ne.ln) go to 190                             0016
      go to 200                                                         0017
  150 continue                                                          0018
      if(lam.eq.lyy) lam=ly                                             0019
      if(lam.eq.lnn) lam=ln                                             0020
      go to 200                                                         0021
  190 nerr=1                                                            0022
  200 return                                                            0023
  990 format(a1)                                                        0024
      end                                                               0025
c                                                                       0026
c***********************************************************************0027
```

Subroutine INSERT

```
c***********************************************************************0001
c                          SUBROUTINE INSERT                            0002
c***********************************************************************0003
c                                                                       0004
c     Inserts non-bank component into sequence for listing components   0005
c     in output                                                         0006
c                                                                       0007
      subroutine insert(ny,nc,nn,io,n)                                  0008
      dimension ny(50),nc(50),nn(50),io(50)                             0009
      if(ny(n).lt.4) na=nn(n)/nc(n)                                     0010
      n1=n-1                                                            0011
      kk=0                                                              0012
      do 110 k=1,n1                                                     0013
      if(ny(n).gt.2) go to 105                                          0014
      if(ny(io(k)).gt.2) go to 110                                      0015
      nb=nn(io(k))/nc(io(k))                                            0016
```

```
      if(nb.lt.na) go to 110
      kk=k
      go to 110
  105 continue
      if(io(k).eq.11.or.io(k).eq.12) go to 110
      if(ny(n).eq.ny(io(k))) kk=k
      if(ny(n).eq.4) kk=k-1
  110 continue
      kkk=kk+1
      do 210 k=n1,kkk,-1
      kq=k+1
  210 io(kq)=io(k)
      io(kkk)=n
      return
      end
c
c*********************************************************************
```

Subroutine LISTCLAYS

```
c*********************************************************************
c                        SUBROUTINE LISTCLAYS
c*********************************************************************
c
c     Updates CLAY.ID and prints out listing clay analysis data files.
c
      subroutine listclays
      character*80 180
      character*75 175
      character*8 18
      logical*4 mol
      open(17,file='CLAY.ID',status='old',err=890)
      read(17,901,err=890,end=890) 180
      open(18,status='scratch',err=890)
      write(18,901) 180
      do 100 i=1,999
      write(18,903) i
      inquire(file=18,exist=mol)
      if(mol) go to 90
      go to 100
   90 continue
      open(19,file=18,status='old',err=95)
      read(19,905,end=95,err=95) 175
      write(18,902) i,175
      write(1,902) i,175
      close(19)
      go to 100
   95 continue
      close(19,status='delete')
  100 continue
  110 continue
      close(17,status='delete')
```

```
      rewind(18)
      open(16,file='CLAY.ID',status='new',err=810)
  200 continue
      do 210 i=1,1000
      read(18,901,end=220) 180
      write(16,901) 180
  210 continue
  220 continue
      close(16)
      close(18)
      go to 890
  800 continue
      write(1,921)
      close(17)
      close(18)
      go to 890
  810 continue
      close(16)
      i1=2
  811 continue
      if(i1.gt.9) go to 813
      do 812 i=i1,9
      i1=i1+1
      write(18,904) i
      inquire(file=18,exist=mo1)
      if(mo1) go to 812
      go to 814
  812 continue
  813 continue
      write(1,923)
      close(18)
      go to 890
  814 continue
      open(16,file=18,status='new',err=815)
      write(1,922) 18
      go to 200
  815 continue
      close(16)
      go to 811
  890 continue
      return
  901 format(a80)
  902 format(i3.3,2x,a75)
  903 format('CLAY.',i3.3)
  904 format('CLAY.ID',i1)
  905 format(a75)
  921 format('Cannot read present CLAY.ID; file edit required!')
  922 format('CLAY.ID renamed ',a8)
  923 format('CLAY.ID destroyed - SORRY')
      end
c
c*********************************************************************
```

Subroutines ARRAY and DGELG

```
c*********************************************************************
c                          SUBROUTINE ARRAY
c*********************************************************************
c
c     IBM software for converting two-dimensional to linear arrays.
c
      subroutine array(mode,i,j,n,m,s,d)
      dimension s(1),d(1)
      ni=n-i
      if(mode-1)100,100,120
  100 ij=i*j+1
      nm=n*j+1
      do 110 k=1,j
      nm=nm-ni
      do 110 l=1,i
      ij=ij-1
      nm=nm-1
      d(nm)=s(ij)
  110 continue
      go to 140
  120 ij=0
      nm=0
      do 130 k=1,j
      do 125 l=1,i
      ij=ij+1
      nm=nm+1
      s(ij)=d(nm)
  125 continue
      nm=nm+ni
  130 continue
  140 return
      end
c
c*********************************************************************

c*********************************************************************
c                          SUBROUTINE DGELG
c*********************************************************************
c
c     IBM software for solving simultaneous linear equations.
c
      subroutine dgelg(r,a,m,n,eps,ier)
      dimension a(1),r(1)
      double precision r,a,piv,tb,tol,pivi
      if(m)23,23,1
    1 ier=0
      piv=0.d0
      mm=m*m
      nm=n*m
      do 3 l=1,mm
      tb=dabs(a(l))
```

```
      if(tb-piv)3,3,2
    2 piv=tb
      i=1
    3 continue
      tol=eps*piv
      lst=1
      do 17 k=1,m
      if(piv)23,23,4
    4 if(ier)7,5,7
    5 if(piv-tol)6,6,7
    6 ier=k-1
    7 pivi=1.d0/a(i)
      j=(i-1)/m
      i=i-j*m-k
      j=j+1-k
      do 8 l=k,nm,m
      ll=l+i
      tb=pivi*r(ll)
      r(ll)=r(l)
      r(l)=tb
    8 continue
      if(k-m)9,18,18
    9 lend=lst+m-k
      if(j)12,12,10
   10 ii=j*m
      do 11 l=lst,lend
      tb=a(l)
      ll=l+ii
      a(l)=a(ll)
      a(ll)=tb
   11 continue
   12 do 13 l=lst,mm,m
      ll=l+i
      tb=pivi*a(ll)
      a(ll)=a(l)
      a(l)=tb
   13 continue
      a(lst)=j
      piv=0.d0
      lst=lst+1
      j=0
      do 16 ii=lst,lend
      pivi=-a(ii)
      ist=ii+m
      j=j+1
      do 15 l=ist,mm,m
      ll=l-j
      a(l)=a(l)+pivi*a(ll)
      tb=dabs(a(l))
      if(tb-piv)15,15,14
   14 piv=tb
      i=l
   15 continue
```

```
      do 16 l=k,nm,m                                                    0104
      ll=l+j                                                            0105
      r(ll)=r(ll)+pivi*r(l)                                             0106
   16 continue                                                          0107
      lst=lst+m                                                         0108
   17 continue                                                          0109
   18 if(m-1)23,22,19                                                   0110
   19 ist=mm+m                                                          0111
      lst=m+1                                                           0112
      do 21 i=2,m                                                       0113
      ii=lst-i                                                          0114
      ist=ist-lst                                                       0115
      l=ist-m                                                           0116
      l=a(l)+0.5d0                                                      0117
      do 21 j=ii,nm,m                                                   0118
      tb=r(j)                                                           0119
      ll=j                                                              0120
      do 20 k=ist,mm,m                                                  0121
      ll=ll+l                                                           0122
      tb=tb-a(k)*r(ll)                                                  0123
   20 continue                                                          0124
      k=j+l                                                             0125
      r(j)=r(k)                                                         0126
      r(k)=tb                                                           0127
   21 continue                                                          0128
   22 return                                                            0129
   23 ier=-1                                                            0130
      go to 22                                                          0131
      end                                                               0132
c*********************************************************************** 0133
```

FORMDAT SOURCE CODE, FILE OF DATA CONSTANTS FOR CLAYFORM

```
SiO2 Si  60.0843  140                    0001
TiO2 Ti  79.8988  141                    0002
Al2O3Al  101.96128261                    0003
Fe2O3Fe3+159.6922 261                    0004
FeO  Fe2+71.8464  121                    0005
MnO  Mn  70.9374  121                    0006
MgO  Mg  40.3044  121                    0007
CaO  Ca  56.0794  122                    0008
Na2O Na  61.97894 222                    0009
K2O  K   94.196   222                    0010
H2O+ OH  18.0152  223                    0011
H2O- H2O 18.0152  104                    0012
LOI  H3O+18.0152  112                    0013
ZrO2 Zr  123.2188 141                    0014
V2O3 V   149.8810 261                    0015
Cr2O3Cr  151.9902 261                    0016
CoO  Co  74.9326  121                    0017
NiO  Ni  74.6994  121                    0018
CuO  Cu  79.5454  121                    0019
```

```
ZnO  Zn  81.3794  121                    0020
PbO  Pb  223.1994 122                    0021
BeO  Be  25.01158 121                    0022
SrO  Sr  103.6194 122                    0023
BaO  Ba  153.3294 122                    0024
Li2O Li  29.8814  221                    0025
Rb2O Rb  186.935  222                    0026
NH3  NH4 17.0304  112                    0027
F    F   18.9984  113                    0028
Cl   Cl  35.453   113                    0029
B2O3 B   69.6182  261                    0030
End      0.0                             0031
```

REFERENCES

Bodine, M. W., Jr., 1987, CLAYFORM, a FORTRAN 77 computer program apportioning the constituents in the chemical analysis of a clay or other silicate mineral into a structural formula: Computers & Geosciences, v. 13, p. 77-88.

IBM Applications Program, 1970, System/360 scientific subroutine package, Version III, Programer's manual (Program No. 360A-CM-03X): IBM Corporation, no. GH20-0205-4, 5th edition, 454 p.