

U.S. DEPARTMENT OF THE INTERIOR

U.S. GEOLOGICAL SURVEY

Computer Simulation Program for Quantitative
Mineral Resource Assessment

by

D. H. Root, W. A. Scott and L. J. Drew¹

Open-File Report 92-203

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

¹Reston, VA

Introduction

The MARK3 computer simulation program is designed to aggregate probabilistic estimates of number of deposits and the data in a library of grade-tonnage deposit models to yield probabilistic estimates of metal tonnages and deposit tonnages. This report includes the computer code for the program written in Fortran 77, sample ancillary library files of the three different types, and a sample session of a run of the program.

The three sample library files are for the Sedimentary Exhalative zinc-lead deposit type. The first file, Appendix 1, is the data file. The second file, Appendix 2, has the parameters used if the lognormal approximations are used for metal grades and deposit tonnages option. The third file, Appendix 3, contains the parameters used if the piecewise linear approximations are used for metal grades and deposit tonnages.

Appendix 4 contains a sample run of the program.

```

00001:
00002:      PROGRAM MARK3
00003:C
00004:      MARK3 IS A MINERAL SIMULATION PROGRAM WITH OVER 80
00005:      DIFFERENT MODELS AVAILABLE.
00006:C
00007:C      ----- (234) IS THE NEW NUMBER OF MAX # OF MODELS
00008:C      "CALL QUANTILE" OUT TEMPORAILY TO REDUCE TIME W/ (C)OMMENT(Q) PRIN
00009:C      *****
00010:      COMMON/MET/ SMETAL
00011:      COMMON /BLK2/ XX(234), ND(6), D(6),INOD
00012:      COMMON /DFILE/ DINFILE,METN(10),IECOD,IEMPCOD(99,10)
00013:      COMMON /BLKE1/ ICO(234,10),INFOC(234),NSUITE(234),SPROB(234,12)
00014:      COMMON /BLKE2/ FEMP(234,12,11,10),DEMP(500,12),PVAL(11)
00015:      COMMON /BLKE3/ AEMP(234,12,10,10),ISUITE(234,12)
00016:      COMMON /BLK4/ JSUITE(234,12),P(234,12,10,10)
00017:      COMMON /BLK5/ R(10,10),ADUM(10,10),ICOMM(234),ERR(12)
00018:      COMMON /BLK6/ ICP1,IC,IDIST,NOR
00019:      COMMON /BLK7/ AA(11),BB(11),ZEMP(101),XNORE(2,1100),XEMP(101)
00020:      COMMON /BLK8/ METYES(10)
00021:      COMMON /BLK9/ ZD(12,10), ZM(12,10)
00022:      COMMON /OWN/L1(521),P2(32),LP,LQ,NQ1,MM,MEMP
00023:      COMMON /DRDIST/ PROBZ,IZDFLAG,DISTP3(3),DISTP5(5)
00024:      COMMON /CAL1/ XM(10),SD(10)
00025:      COMMON /METFLAG/ METNUM,NUMCOM,UNITMP
00026:      COMMON /NUMDEP/ RNUMD(3,12,10),NQ,KKONT(12,10)
00027:      COMMON /CLN/ CLINE(3,100,12,10)
00028:      COMMON /TABLE/ TTABLE(1100,2)
00029:      INTEGER TYPE,UNITIN,UNITMP,UNITSC1,UNITSC2,IPMESS
00030:      INTEGER * 4 IORDER(13), METTEM(45),MCNT
00031:      INTEGER * 2 IDER,MODN(234),NUMOFMTL(99)
00032:      INTEGER * 2 ICL1,ICL2,ICL3,IECOD,NPFLAG,JF
00033:      REAL SMETAL(4999,35)
00034:      REAL ZERO(35)
00035:      REAL ISZ,ISW,IS,EN,MT(44),MG(44),ST(44),SU(35),
00036:      1 SG(44),CPG1(44),CPS1(44),CPG2(44),CPS2(44),CPG3(44),CPS3(44)
00037:      CHARACTER * 32 NAM * 6,FILEIN,DPFILE,
00038:      1 TOTMOD,SCFILE1,SCFILE2
00039:      CHARACTER * 60 CMETAL(46)
00040:      CHARACTER * 60 DINFILE,CPLINE,CPLFILE
00041:      CHARACTER * 65 MTITLES(99),REQNAME,AREANM,TODAYDT
00042:      CHARACTER * 1 ELOPT,CPLP,STAR(234),TOTF
00043:      CHARACTER * 3 READPW,TEMPDF
00044:      CHARACTER * 4 PNUM1,PNUM2,PNUM3,MCOL(35)
00045:      CHARACTER * 5 TRACT(234)
00046:      CHARACTER * 12 PMET1,PMET2,PMET3
00047:      CHARACTER * 10 ELNLABEL
00048:      CHARACTER * 60 DATAFILES(99)
00049:      CHARACTER * 19 PMETAL(10)
00050:      CHARACTER * 80 AVMETS,PRINTFILE
00051:      CHARACTER * 80 MODELTS(99)
00052:      CHARACTER * 240 DATALINE
00053:      CHARACTER * 2 TMODN,TRTMOD(99)
00054:      CHARACTER * 12 TRACTIN(234),TRACTIN2
00055:      DIMENSION X(10),Z(10)
00056:      DIMENSION KONT(234),JORDER(13)
00057:      DIMENSION NWKONT(234),XZ(10)
00058:      DIMENSION T(200),G(200),TS(200),GS(200),Z48(200),X48(200)
00059:      DIMENSION TMPARR(35)

```

```

00060:      DIMENSION POFZ(234)
00061:      DATA (DISTP3(NDJ), NDJ = 1,3) /.90,.50,.10/
00062:      DATA (DISTP5(NDJ), NDJ = 1,5) /.90,.50,.10,.05,.01/
00063: COMMENT      DATA (DISTP5(NDJ), NDJ = 1,5) /.90,.70,.50,.10,.01/
00064:      DATA UNITIN/10//UNITMP/13//UNITSC1/11//UNITSC2/12/
00065:      DATA CMETAL/'CHROMIUM','COPPER','MOLYBDENUM',
00066:      1 'GOLD','IRON','TUNGSTEN',
00067:      2 'ZINC','SILVER','LEAD',
00068:      3 'NICKEL','ASBESTOS','MANGANESE',
00069:      4 'BARITE','MERCURY','PLATINUM',
00070:      5 'PHOSPHORUS','COBALT','TIN',
00071:      6 'ORE','THORIUM','R.E. OXIDE',
00072:      7 'FLUORINE','ANTIMONY','URANIUM','PALLADIUM',
00073:      8 'SILICON','CALCIUM','ALUMINUM','RUTILE','ILMENITE',
00074:      9 'LEUCOXENE','ZIRCON','MONAZITE REO','GYPSUM',
00075:      1 'NIOBIUM','IRIDIUM','37NA','38NA','39NA',
00076:      2 '40NA','41NA','42NA','43NA','44NA','45NA','46NA'/
00077:      DATA DATAFILES /'HOMESTAKE1','PORPHCM2',
00078:      C 'EVNBAR3','PORPHCU4',
00079:      C 'CLIMAXM05','MOLY6',
00080:      C 'FESKARN7','CUSKARN8',
00081:      C 'PCURELAT9','WSKARN10',
00082:      C 'CYPRUSMS11','VOLCANU12',
00083:      C 'SEDEX13','MBEDG14',
00084:      C 'CHGOLD15','COMSTOCK16',
00085:      C 'SEDHOST17','DISTALD18',
00086:      C 'SYNORVOL19','ALKHAU20',
00087:      C 'MXBASE21','LZSKARN22',
00088:      C 'BBARITE23','WVEINS24',
00089:      C 'OLYMPICPV25','CHUGACHLS26',
00090:      C 'LWSULF27','SADOEV28',
00091:      C 'HSPRINGHG29','BESSHIMS30',
00092:      C 'REPMN31','BEACHPL32',
00093:      C 'SMANG33','PORCUAU34',
00094:      C 'LATNI35','OOLITIC36',
00095:      C 'LATSAP37','EPQALOW38',
00096:      C 'SCBRECCIA39','RHOSTSN40',
00097:      C 'KARSTB41','SCARB42',
00098:      C 'PORPHAU43','SIERRAN44',
00099:      C 'HOTSPRNG45','POLYVEIN46',
00100:      C 'POLYR47','SKARN48',
00101:      C 'RPLACEM49','GREISEN50',
00102:      C 'VEINS51','PVEINKH52',
00103:      C 'CARBREE53','PLACER54',
00104:      C 'TVEIN55','SIMPSB56',
00105:      C 'THREV57','CREEDE58',
00106:      C 'PMVEINS59','IREPLACE60',
00107:      C 'SHEETEDV61','METVEINS62',
00108:      C 'SHOSTCU63','SHOSTCU64',
00109:      C 'SERPHABS65','NA',
00110:      C 'CUBANVMN67','PHOSUPW68',
00111:      C 'UNCONUAU69','LATBAUX70',
00112:      C 'DUNNICU71','FRANCMN72',
00113:      C 'CYPVOLMN73','EPITHMN74',
00114:      C 'SILCARBG75','SANDPBZN76',
00115:      C 'SUPERFE77','WARMCPH78',
00116:      C 'OLYPENMN79','KOMATNC80',
00117:      C '81-NA','82-NA','83-NA','84-NA',
00118:      C '85-NA','86-NA','87-NA','88-NA',
00119:      C '89-NA','90-NA','91-NA','92-NA',

```

00120: C '93-NA','94-NA','95-NA','96-NA',
 00121: C '97-NA','98-NA','99-NA',
 00122: DATA MTITLES /' 1 - HOMESTAKE AU(36B)',
 00123: C ' 2 - PORPHYRY CU-MO(21A)',
 00124: C ' 3 - EPIGENETIC VEINS BARITE(27E)',
 00125: C ' 4 - PORPHYRY COPPER(17)',
 00126: C ' 5 - STOCKWORK(CLIMAX) MO(16)',
 00127: C ' 6 - PORPHYRY MO, LOW-F(21B)',
 00128: C ' 7 - FE SKARN(18D)',
 00129: C ' 8 - COPPER SKARN(18B)',
 00130: C ' 9 - PORPHYRY CU, SKARN RELATED(18A)',
 00131: C '10 - TUNGSTEN SKARN(14A)',
 00132: C '11 - CYPRUS MASSIVE SULFIDE(24A)',
 00133: C '12 - VOLCANOGENIC U(25F)(06/06/91)',
 00134: C '13 - SEDIMENTARY EXHALATIVE ZN-PB(31A)',
 00135: C '14 - MARINE BEDDED GYPSUM(35AE)',
 00136: C '15 - CARBONATE-HOSTED AU-AG(26A)(CURRENT TO 01/15/91)',
 00137: C '16 - COMSTOCK EPITHERMAL VEINS(25C)',
 00138: C '17 - SEDIMENT-HOSTED AU(26A.1)(UPDATED 01/16/91)',
 00139: C '18 - DISTAL DISSEMINATED AG-AU(19C)',
 00140: C '19 - SYNOROGENIC AND SYNVOLCANIC NI-CU(7A)',
 00141: C '20 - ALKALINE-HOSTED AU(22B)',
 00142: C '21 - MIXED BASE & PRECIOUS METAL VEINS',
 00143: C '22 - ZN-PB SKARN(18C)',
 00144: C '23 - BEDDED BARITE(31B)',
 00145: C '24 - TUNGSTEN VEINS(15A)',
 00146: C '25 - VOLCANOGENIC MN, OLYMPIC PENINSULA(24C)',
 00147: C '26 - CHUGACH-TYPE METAMORPHIC AU(36A.1)(UPDATED 01/16/91)',
 00148: C '27 - LOW SULFIDE AU-QUARTZ VEINS(36A)(TO 01/15/91)',
 00149: C '28 - SADO EPITHERMAL VEIN(27D)',
 00150: C '29 - HOT SPRINGS HG(27A)',
 00151: C '30 - BESSHI MASSIVE SULFIDE(24B)',
 00152: C '31 - REPLACEMENT MN(19B)',
 00153: C '32 - BEACH PLACERS(39C)',
 00154: C '33 - SEDIMENTARY MN(34B)',
 00155: C '34 - PORPHYRY CU-AU(20C)',
 00156: C '35 - LATERITIC NI(38A)',
 00157: C '36 - OOLITIC IRONSTONES(34E)',
 00158: C '37 - LATERITE-SAPROLITE AU(38G)',
 00159: C '38 - EPITHERMAL QZ-AL VEINS(LOW GRADE)(25E)',
 00160: C '39 - SOLUTION-COLLAPSE BRECCIA PIPE U(32E)',
 00161: C '40 - RHYOLITE-HOSTED SN(25H)',
 00162: C '41 - KARST TYPE BAUXITE(38C)',
 00163: C '42 - S.E. MISSOURI PB-ZN(32A) & APPALACHIAN ZN(32B)',
 00164: C '43 - PORPHYRY GOLD(20D.1)',
 00165: C '44 - SIERRA KUROKO MASSIVE SULFIDE(28A)',
 00166: C '45 - HOT SPRINGS AU-AG(25A)',
 00167: C '46 - POLYMETALLIC VEINS(22C)',
 00168: C '47 - POLYMETALLIC REPLACEMENT(19A)',
 00169: C '48 - SN SKARN(14B)',
 00170: C '49 - SN REPLACEMENT(14C)',
 00171: C '50 - SN GREISEN(15C)',
 00172: C '51 - SN VEINS(15B)',
 00173: C '52 - POLYMETALLIC VEIN-KENO HILL(BORA)',
 00174: C '53 - (CARBONATITE) REE(10)',
 00175: C '54 - PLACER AU-PGE(39A)',
 00176: C '55 - THORIUM-RARE EARTH ELEMENTS(11C)(CURRENT TO 01/15/91)',
 00177: C '56 - SIMPLE SB(27D)',
 00178: C '57 - THORIUM-RARE EARTH VEINS(11C.1)(UPDATED 01/16/91)',
 00179: C '58 - CREEDE EPITHERMAL VEINS(25B)',

```

00180: C '59 - PRECIOUS METAL VEINS(?)',
00181: C '60 - IRREGULAR REPLACEMENT(?)',
00182: C '61 - SN SHEETED VEINS(?)',
00183: C '62 - METAMORPHIC AU-QUARTZ VEINS(?)',
00184: C '63 - SEDIMENT-HOSTED CU(30B)',
00185: C '64 - SEDIMENT-HOSTED CU (KOOTENAI N.F.)(30B ?)',
00186: C '65 - SERPERTINE-HOSTED ASBESTOS(8D)',
00187: C '66 - N-A',
00188: C '67 - CUBAN-TYPE VOLCANOGENIC MN(24C.2)',
00189: C '68 - UPWELLING-TYPE PHOSPHATE(34C)',
00190: C '69 - UNCONFORMITY U-AU(37A)',
00191: C '70 - LATERITIC-TYPE BAUXITE(38B)',
00192: C '71 - DUNITIC CU-NI(6B)',
00193: C '72 - FRANCISCAN-TYPE VOLCANOGENIC MN(24C.1)',
00194: C '73 - CYPRUS-TYPE VOLCANOGENIC MN(24.4)',
00195: C '74 - EPITHERMAL MN(25G)',
00196: C '75 - SILICA-CARBONATE HG(27C)',
00197: C '76 - SANDSTONE-HOSTED PB-ZN(30A)',
00198: C '77 - SUPERIOR FE & ALGOMA FE(34A)',
00199: C '78 - WARM-CURRENT TYPE PHOSPHATE(34D)',
00200: C '79 - OLYMPIC (?) MN-FE (?)',
00201: C '80 - KOMATIITIC CU-NI(6A)',
00202: C '81-NA','82-NA','83-NA','84-NA','85-NA',
00203: C '86-NA','87-NA','88-NA','89-NA','90-NA',
00204: C '91-NA','92-NA','93-NA','94-NA','95-NA',
00205: C '96-NA','97-NA','98-NA','99-NA'
00206: DATA (IORDER(I),I=1,13) /1,500,1000,1500,2000,2500,
00207: C 3000,3500,4000,4500,4750,4950,4999/
00208: DATA MCOL / 1 ' 2 ' 3 ' 4 ' 5 ' 6 ' 7 ' 8 ' 9
00209: C '10 ' '11 ' '12 ' '13 ' '14 ' '15 ' '16 ' '17 ' '18 ' '19 '
00210: C '20 ' '21 ' '22 ' '23 ' '24 ' '25 ' '26 ' '27 ' '28 '
00211: C '29 ' '30 ' '31 ' '32 ' '33 ' '34 ' '35 '
00212: DATA (NUMOFMTL(I),I=1,99)
00213: C /2,4,1,4,1,1,1,3,4,1,
00214: C 5,2,3,1,2,2,2,2,6,2,
00215: C 2,5,1,1,2,2,2,5,1,4,
00216: C 4,5,1,4,2,6,1,3,1,1,
00217: C 1,3,1,5,2,5,5,1,1,1,
00218: C 1,3,3,1,3,1,2,5,2,4,
00219: C 1,2,3,3,1,0,2,1,1,2,
00220: C 6,1,2,3,1,5,2,1,2,6,
00221: C 0,0,0,0,0,0,0,0,0,0,
00222: C 0,0,0,0,0,0,0,0,0,0/
00223: DATA (IEMPCOD(1,J), J = 1,10) /19,4,8,0,0,0,0,0,0,0/
00224: DATA (IEMPCOD(2,J), J = 1,10) /19,2,3,4,8,0,0,0,0,0/
00225: DATA (IEMPCOD(3,J), J = 1,10) /19,13,0,0,0,0,0,0,0,0/
00226: DATA (IEMPCOD(4,J), J = 1,10) /19,2,3,4,8,0,0,0,0,0/
00227: DATA (IEMPCOD(5,J), J = 1,10) /19,3,0,0,0,0,0,0,0,0/
00228: DATA (IEMPCOD(6,J), J = 1,10) /19,3,0,0,0,0,0,0,0,0/
00229: DATA (IEMPCOD(7,J), J = 1,10) /19,5,0,0,0,0,0,0,0,0/
00230: DATA (IEMPCOD(8,J), J = 1,10) /19,4,8,2,0,0,0,0,0,0/
00231: DATA (IEMPCOD(9,J), J = 1,10) /19,2,3,4,8,0,0,0,0,0/
00232: DATA (IEMPCOD(10,J), J = 1,10) /19,6,0,0,0,0,0,0,0,0/
00233: DATA (IEMPCOD(11,J), J = 1,10) /19,4,8,2,9,7,0,0,0,0/
00234: DATA (IEMPCOD(12,J), J = 1,10) /19,24,3,0,0,0,0,0,0,0/
00235: DATA (IEMPCOD(13,J), J = 1,10) /19,9,7,8,0,0,0,0,0,0/
00236: DATA (IEMPCOD(14,J), J = 1,10) /19,34,0,0,0,0,0,0,0,0/
00237: DATA (IEMPCOD(15,J), J = 1,10) /19,4,8,0,0,0,0,0,0,0/
00238: DATA (IEMPCOD(16,J), J = 1,10) /19,4,8,0,0,0,0,0,0,0/
00239: DATA (IEMPCOD(17,J), J = 1,10) /19,4,8,0,0,0,0,0,0,0/

```

```

00240: DATA (IEMPCOD(18,J), J = 1,10) /19,4,8,0,0,0,0,0,0,0/
00241: DATA (IEMPCOD(19,J), J = 1,10) /19,2,10,17,25,4,15,0,0,0/
00242: DATA (IEMPCOD(20,J), J = 1,10) /19,4,8,0,0,0,0,0,0,0/
00243: DATA (IEMPCOD(21,J), J = 1,10) /19,4,8,0,0,0,0,0,0,0/
00244: DATA (IEMPCOD(22,J), J = 1,10) /19,7,9,2,8,4,0,0,0,0/
00245: DATA (IEMPCOD(23,J), J = 1,10) /19,13,0,0,0,0,0,0,0,0/
00246: DATA (IEMPCOD(24,J), J = 1,10) /19,6,0,0,0,0,0,0,0,0/
00247: DATA (IEMPCOD(25,J), J = 1,10) /19,12,5,0,0,0,0,0,0,0/
00248: DATA (IEMPCOD(26,J), J = 1,10) /19,4,8,0,0,0,0,0,0,0/
00249: DATA (IEMPCOD(27,J), J = 1,10) /19,4,8,0,0,0,0,0,0,0/
00250: DATA (IEMPCOD(28,J), J = 1,10) /19,4,8,2,9,7,0,0,0,0/
00251: DATA (IEMPCOD(29,J), J = 1,10) /19,14,0,0,0,0,0,0,0,0/
00252: DATA (IEMPCOD(30,J), J = 1,10) /19,4,8,2,7,0,0,0,0,0/
00253: DATA (IEMPCOD(31,J), J = 1,10) /19,12,5,16,2,0,0,0,0,0/
00254: DATA (IEMPCOD(32,J), J = 1,10) /19,29,30,31,32,33,0,0,0,0/
00255: DATA (IEMPCOD(33,J), J = 1,10) /19,12,0,0,0,0,0,0,0,0/
00256: DATA (IEMPCOD(34,J), J = 1,10) /19,2,3,4,8,0,0,0,0,0/
00257: DATA (IEMPCOD(35,J), J = 1,10) /19,10,17,0,0,0,0,0,0,0/
00258: DATA (IEMPCOD(36,J), J = 1,10) /19,5,26,16,12,27,28,0,0,0/
00259: DATA (IEMPCOD(37,J), J = 1,10) /19,4,0,0,0,0,0,0,0,0/
00260: DATA (IEMPCOD(38,J), J = 1,10) /19,4,8,2,0,0,0,0,0,0/
00261: DATA (IEMPCOD(39,J), J = 1,10) /19,24,0,0,0,0,0,0,0,0/
00262: DATA (IEMPCOD(40,J), J = 1,10) /19,18,0,0,0,0,0,0,0,0/
00263: DATA (IEMPCOD(41,J), J = 1,10) /19,28,0,0,0,0,0,0,0,0/
00264: DATA (IEMPCOD(42,J), J = 1,10) /19,7,9,8,0,0,0,0,0,0/
00265: DATA (IEMPCOD(43,J), J = 1,10) /19,4,0,0,0,0,0,0,0,0/
00266: DATA (IEMPCOD(44,J), J = 1,10) /19,2,7,9,8,4,0,0,0,0/
00267: DATA (IEMPCOD(45,J), J = 1,10) /19,4,8,0,0,0,0,0,0,0/
00268: DATA (IEMPCOD(46,J), J = 1,10) /19,4,8,2,9,7,0,0,0,0/
00269: DATA (IEMPCOD(47,J), J = 1,10) /19,9,7,2,8,4,0,0,0,0/
00270: DATA (IEMPCOD(48,J), J = 1,10) /19,18,0,0,0,0,0,0,0,0/
00271: DATA (IEMPCOD(49,J), J = 1,10) /19,18,0,0,0,0,0,0,0,0/
00272: DATA (IEMPCOD(50,J), J = 1,10) /19,18,0,0,0,0,0,0,0,0/
00273: DATA (IEMPCOD(51,J), J = 1,10) /19,18,0,0,0,0,0,0,0,0/
00274: DATA (IEMPCOD(52,J), J = 1,10) /19,8,9,7,0,0,0,0,0,0/
00275: DATA (IEMPCOD(53,J), J = 1,10) /19,35,21,16,0,0,0,0,0,0/
00276: DATA (IEMPCOD(54,J), J = 1,10) /19,4,0,0,0,0,0,0,0,0/
00277: DATA (IEMPCOD(55,J), J = 1,10) /19,20,21,24,0,0,0,0,0,0/
00278: DATA (IEMPCOD(56,J), J = 1,10) /19,23,0,0,0,0,0,0,0,0/
00279: DATA (IEMPCOD(57,J), J = 1,10) /19,20,21,0,0,0,0,0,0,0/
00280: DATA (IEMPCOD(58,J), J = 1,10) /19,4,8,2,9,7,0,0,0,0/
00281: DATA (IEMPCOD(59,J), J = 1,10) /19,4,8,0,0,0,0,0,0,0/
00282: DATA (IEMPCOD(60,J), J = 1,10) /19,9,8,2,4,0,0,0,0,0/
00283: DATA (IEMPCOD(61,J), J = 1,10) /19,18,0,0,0,0,0,0,0,0/
00284: DATA (IEMPCOD(62,J), J = 1,10) /19,4,8,0,0,0,0,0,0,0/
00285: DATA (IEMPCOD(63,J), J = 1,10) /19,2,17,8,0,0,0,0,0,0/
00286: DATA (IEMPCOD(64,J), J = 1,10) /19,2,17,8,0,0,0,0,0,0/
00287: DATA (IEMPCOD(65,J), J = 1,10) /19,11,0,0,0,0,0,0,0,0/
00288: DATA (IEMPCOD(66,J), J = 1,10) /0,0,0,0,0,0,0,0,0,0/
00289: DATA (IEMPCOD(67,J), J = 1,10) /19,12,5,0,0,0,0,0,0,0/
00290: DATA (IEMPCOD(68,J), J = 1,10) /19,16,0,0,0,0,0,0,0,0/
00291: DATA (IEMPCOD(69,J), J = 1,10) /19,24,0,0,0,0,0,0,0,0/
00292: DATA (IEMPCOD(70,J), J = 1,10) /19,10,17,0,0,0,0,0,0,0/
00293: DATA (IEMPCOD(71,J), J = 1,10) /19,2,10,17,25,36,4,0,0,0/
00294: DATA (IEMPCOD(72,J), J = 1,10) /19,12,0,0,0,0,0,0,0,0/
00295: DATA (IEMPCOD(73,J), J = 1,10) /19,12,5,0,0,0,0,0,0,0/
00296: DATA (IEMPCOD(74,J), J = 1,10) /19,12,5,16,0,0,0,0,0,0/
00297: DATA (IEMPCOD(75,J), J = 1,10) /19,14,0,0,0,0,0,0,0,0/
00298: DATA (IEMPCOD(76,J), J = 1,10) /19,4,8,2,9,7,0,0,0,0/
00299: DATA (IEMPCOD(77,J), J = 1,10) /19,5,16,0,0,0,0,0,0,0/

```

00300: DATA (IEMPCOD(78,J), J = 1,10) /19,16,0,0,0,0,0,0,0,0/
 00301: DATA (IEMPCOD(79,J), J = 1,10) /19,12,5,0,0,0,0,0,0,0/
 00302: DATA (IEMPCOD(80,J), J = 1,10) /19,2,10,17,25,36,4,0,0,0/
 00303: DATA (IEMPCOD(81,J), J = 1,10) /0,0,0,0,0,0,0,0,0,0/
 00304: DATA (IEMPCOD(82,J), J = 1,10) /0,0,0,0,0,0,0,0,0,0/
 00305: DATA (IEMPCOD(83,J), J = 1,10) /0,0,0,0,0,0,0,0,0,0/
 00306: DATA (IEMPCOD(84,J), J = 1,10) /0,0,0,0,0,0,0,0,0,0/
 00307: DATA (IEMPCOD(85,J), J = 1,10) /0,0,0,0,0,0,0,0,0,0/
 00308: DATA (IEMPCOD(86,J), J = 1,10) /0,0,0,0,0,0,0,0,0,0/
 00309: DATA (IEMPCOD(87,J), J = 1,10) /0,0,0,0,0,0,0,0,0,0/
 00310: DATA (IEMPCOD(88,J), J = 1,10) /0,0,0,0,0,0,0,0,0,0/
 00311: DATA (IEMPCOD(89,J), J = 1,10) /0,0,0,0,0,0,0,0,0,0/
 00312: DATA (IEMPCOD(90,J), J = 1,10) /0,0,0,0,0,0,0,0,0,0/
 00313: DATA (IEMPCOD(91,J), J = 1,10) /0,0,0,0,0,0,0,0,0,0/
 00314: DATA (IEMPCOD(92,J), J = 1,10) /0,0,0,0,0,0,0,0,0,0/
 00315: DATA (IEMPCOD(93,J), J = 1,10) /0,0,0,0,0,0,0,0,0,0/
 00316: DATA (IEMPCOD(94,J), J = 1,10) /0,0,0,0,0,0,0,0,0,0/
 00317: DATA (IEMPCOD(95,J), J = 1,10) /0,0,0,0,0,0,0,0,0,0/
 00318: DATA (IEMPCOD(96,J), J = 1,10) /0,0,0,0,0,0,0,0,0,0/
 00319: DATA (IEMPCOD(97,J), J = 1,10) /0,0,0,0,0,0,0,0,0,0/
 00320: DATA (IEMPCOD(98,J), J = 1,10) /0,0,0,0,0,0,0,0,0,0/
 00321: DATA (IEMPCOD(99,J), J = 1,10) /0,0,0,0,0,0,0,0,0,0/
 00322: DATA MODELTS /

C	INDEX	TITLE	(MODEL NO.)	FILE NAME
00324:	C ' 1	HOMESTAKE AU(36B)		* HOMESTAKE1'
00325:	C ' 2	PORPHYRY CU-MO(21A)		* PORPHCM2'
00326:	C ' 3	EPIGENETIC VEINS BARITE(27E)		* EVNBAR3'
00327:	C ' 4	PORPHYRY COPPER(17)		* PORPHCU4'
00328:	C ' 5	STOCKWORK(CLIMAX) MO(16)		* CLIMAXMO5'
00329:	C ' 6	PORPHYRY MO, LOW-F(21B)		* MOLY6'
00330:	C ' 7	FE SKARN(18D)		* FESKARN7'
00331:	C ' 8	COPPER SKARN(18B)		* CUSKARN8'
00332:	C ' 9	PORPHYRY CU, SKARN RELATED(18A)		* PCURELAT9'
00333:	C ' 10	TUNGSTEN SKARN(14A)		* WSKARN10'
00334:	C ' 11	CYPRUS MASSIVE SULFIDE(24A)		* CYPRUSMS11'
00335:	C ' 12	VOLCANOGENIC U(25F)(06/06/91)		* VOLCANU12'
00336:	C ' 13	SEDIMENTARY EXHALATIVE ZN-PB(31A)		* SEDEX13'
00337:	C ' 14	MARINE BEDDED GYPSUM(35AE)		* MBEDG14'
00338:	C ' 15	CARB-HOST. AU-AG(26A)(CURRENT TO 01/15/91)		* CHGOLD15'
00339:	C ' 16	COMSTOCK EPITHERMAL VEINS(25C)		* COMSTOCK16'
00340:	C ' 17	SEDIMENT-HOST. AU(26A.1)(UPDATED 01/16/91)		* SEDHOST17'
00341:	C ' 18	DISTAL DISSEMINATED AG-AU(19C)		* DISTALD18'
00342:	C ' 19	SYNOROGENIC AND SYNVOLCANIC NI-CU(7A)		* SYNORVOL19'
00343:	C ' 20	ALKALINE-HOSTED AU(22B)		* ALKHAU20'
00344:	C ' 21	MIXED BASE & PRECIOUS METAL VEINS		* MXBASE21'
00345:	C ' 22	ZN-PB SKARN(18C)		* LZSKARN22'
00346:	C ' 23	BEDDED BARITE(31B)		* BBARITE23'
00347:	C ' 24	TUNGSTEN VEINS(15A)		* WVEINS24'
00348:	C ' 25	VOLCANOGENIC MN, OLYMPIC-PENINSULA(24C)		* OLYMPICPV25'
00349:	C ' 26	CHUGACH L-SULF AU-Q V(36A.1)(UPD 01/16/91)		* CHUGACHLS26'
00350:	C ' 27	LOW SULFIDE AU-QZ VEINS(36A)(TO 01/15/91)		* LWSULF27'
00351:	C ' 28	SADO EPITHERMAL VEIN(27D)		* SADOEV28'
00352:	C ' 29	HOT SPRINGS HG(27A)		* HSPPRINGHG29'
00353:	C ' 30	BESSHI MASSIVE SULFIDE(24B)		* BESSHIMS30'
00354:	C ' 31	REPLACEMENT MN(19B)		* REPMN31'
00355:	C ' 32	BEACH PLACERS(39C)		* BEACHPL32'
00356:	C ' 33	SEDIMENTARY MN(34B)		* SMANG33'
00357:	C ' 34	PORPHYRY CU-AU(20C)		* PORCUAU34'
00358:	C ' 35	LATERITIC NI(38A)		* LATNI35'
00359:	C ' 36	OOLITIC IRONSTONES(34E)		* OOLITIC36'

00360:	C	37	LATERITE-SAPROLITE AU(38G)	* LATSAP37'
00361:	C	38	EPITHERMAL QZ-AL VEINS(LOW GRADE)(25E)	* EPQALOW38'
00362:	C	39	SOLUTION-COLLAPSE BRECCIA PIPE U(32E)	* SCBRECCIA39'
00363:	C	40	RHYOLITE-HOSTED SN(25H)	* RHOSTSN40'
00364:	C	41	KARST TYPE BAUXITE(38C)	* KARSTB41'
00365:	C	42	S.E. MO PB-ZN(32A) & APPALCHIAN ZN(32B)	* SCARB42'
00366:	C	43	PORPHYRY GOLD(20D.1)	* PORPHAU43'
00367:	C	44	SIERRAN KUROKO MASSIVE SULFIDE(28A)	* SIERRAN44'
00368:	C	45	HOT SPRINGS AU-AG(25A)	* HOTSPRNG45'
00369:	C	46	POLYMETALLIC VEINS(22C)	* POLYVEIN46'
00370:	C	47	POLYMETALLIC REPLACEMENT(19A)	* POLYR47'
00371:	C	48	SN SKARN(14B)	* SKARN48'
00372:	C	49	SN REPLACEMENT(14C)	* RPLACEM49'
00373:	C	50	SN GREISEN(15C)	* GREISEN50'
00374:	C	51	SN VEINS(15B)	* VEINS51'
00375:	C	52	POLYMETALLIC VEIN-KENO HILL(BORA)	* PVEINKH52'
00376:	C	53	(CARBONATITE) REE(10)	* CARBREE53'
00377:	C	54	PLACER AU-PGE(39A)	* PLACER54'
00378:	C	55	TH-R.E.E.(11C)(CURRENT TO 01/15/91)	* TVEIN55'
00379:	C	56	SIMPLE SB(27D)	* SIMPSB56'
00380:	C	57	TH-R.E. VEINS(11C.1)(UPDATED 01-16-91)	* THREV57'
00381:	C	58	CREEDE EPITHERMAL VEINS(25B)	* CREEDE58'
00382:	C	59	PRECIOUS METAL VEINS(?)	* PMVEINS59'
00383:	C	60	IRREGULAR REPLACEMENT(?)	* IREPLACE60'
00384:	C	61	SN SHEETED VEINS	* SHEETEDV61'
00385:	C	62	METAMORPHIC AU-QUARTZ VEINS(?)	* METVEINS62'
00386:	C	63	SEDIMENT-HOSTED CU(30B)	* SHOSTCU63'
00387:	C	64	SEDIMENT-HOSTED CU(KOOTENAI N.F.)(30B ?)	* SHOSTCU64'
00388:	C	65	SERPENTINE-HOSTED ASBESTOS(8D)	* SERPHABS65'
00389:	C	66	N-A	* N-A'
00390:	C	67	CUBAN-TYPE VOLCANOGENIC MN(24C.2)	* CUBANVMN67'
00391:	C	68	UPWELLING-TYPE PHOSPHATE(34C)	* PHOSUPW68'
00392:	C	69	UNCONFORMITY U-AU(37A)	* UNCONUAU69'
00393:	C	70	LATERIC-TYPE BAUXITE(38B)	* LATBAUX70'
00394:	C	71	DUNITIC CU-NI(6B)	* DUNNICU71'
00395:	C	72	FRANCISCAN-TYPE VOLCANOGENIC MN(24C.1)	* FRANCVMN72'
00396:	C	73	CYPRUS-TYPE VOLCANOGENIC MN(24.4)	* CYPVOLMN73'
00397:	C	74	EPITHERMAL MN(25G)	* EPITHMN74'
00398:	C	75	SILICA-CARBONATE HG(27C)	* SILCARBHG75'
00399:	C	76	SANDSTONE-HOSTED PB-ZN(30A)	* SANDPBZN76'
00400:	C	77	SUPERIOR FE & ALGOMA FE(34A)	* SUPERFE77'
00401:	C	78	WARM-CURRENT TYPE PHOSPHATE(34D)	* WARMCPH78'
00402:	C	79	OLYMPIC (?) MN-FE (?)	* OLYPENMN79'
00403:	C	80	KOMATIITIC CU-NI(6A)	* KOMATNC80'
00404:	C	'81','82','83','84','85','86','87','88','89',		
00405:	C	'90','91','92','93','94','95','96','97','98',		
00406:			DATA (TNTABLE(1,ITJ), ITJ = 1,2) /0.00137,-2.99989/	
00407:			DATA (TNTABLE(2,ITJ), ITJ = 1,2) /0.00166,-2.93989/	
00408:			DATA (TNTABLE(3,ITJ), ITJ = 1,2) /0.00201,-2.87990/	
00409:			DATA (TNTABLE(4,ITJ), ITJ = 1,2) /0.00242,-2.81990/	
00410:			DATA (TNTABLE(5,ITJ), ITJ = 1,2) /0.00291,-2.75990/	
00411:			DATA (TNTABLE(6,ITJ), ITJ = 1,2) /0.00349,-2.69991/	
00412:			DATA (TNTABLE(7,ITJ), ITJ = 1,2) /0.00416,-2.63991/	
00413:			DATA (TNTABLE(8,ITJ), ITJ = 1,2) /0.00496,-2.57992/	
00414:			DATA (TNTABLE(9,ITJ), ITJ = 1,2) /0.00589,-2.51992/	
00415:			DATA (TNTABLE(10,ITJ), ITJ = 1,2) /0.00696,-2.45993/	
00416:			DATA (TNTABLE(11,ITJ), ITJ = 1,2) /0.00821,-2.39993/	
00417:			DATA (TNTABLE(12,ITJ), ITJ = 1,2) /0.00966,-2.33993/	
00418:			DATA (TNTABLE(13,ITJ), ITJ = 1,2) /0.01132,-2.27994/	
00419:			DATA (TNTABLE(14,ITJ), ITJ = 1,2) /0.01322,-2.21994/	

00420: DATA (TNTABLE(15,ITJ), ITJ = 1,2) /0.01540,-2.15995/
 00421: DATA (TNTABLE(16,ITJ), ITJ = 1,2) /0.01788,-2.09995/
 00422: DATA (TNTABLE(17,ITJ), ITJ = 1,2) /0.02069,-2.03996/
 00423: DATA (TNTABLE(18,ITJ), ITJ = 1,2) /0.02357,-1.97996/
 00424: DATA (TNTABLE(19,ITJ), ITJ = 1,2) /0.02744,-1.91996/
 00425: DATA (TNTABLE(20,ITJ), ITJ = 1,2) /0.03146,-1.85996/
 00426: DATA (TNTABLE(21,ITJ), ITJ = 1,2) /0.03594,-1.79997/
 00427: DATA (TNTABLE(22,ITJ), ITJ = 1,2) /0.04094,-1.73997/
 00428: DATA (TNTABLE(23,ITJ), ITJ = 1,2) /0.04649,-1.67997/
 00429: DATA (TNTABLE(24,ITJ), ITJ = 1,2) /0.05263,-1.61997/
 00430: DATA (TNTABLE(25,ITJ), ITJ = 1,2) /0.05939,-1.55997/
 00431: DATA (TNTABLE(26,ITJ), ITJ = 1,2) /0.06682,-1.49998/
 00432: DATA (TNTABLE(27,ITJ), ITJ = 1,2) /0.07495,-1.43998/
 00433: DATA (TNTABLE(28,ITJ), ITJ = 1,2) /0.08380,-1.37998/
 00434: DATA (TNTABLE(29,ITJ), ITJ = 1,2) /0.09343,-1.31998/
 00435: DATA (TNTABLE(30,ITJ), ITJ = 1,2) /0.10384,-1.25998/
 00436: DATA (TNTABLE(31,ITJ), ITJ = 1,2) /0.11508,-1.19999/
 00437: DATA (TNTABLE(32,ITJ), ITJ = 1,2) /0.12715,-1.13999/
 00438: DATA (TNTABLE(33,ITJ), ITJ = 1,2) /0.14008,-1.07999/
 00439: DATA (TNTABLE(34,ITJ), ITJ = 1,2) /0.15387,-1.01999/
 00440: DATA (TNTABLE(35,ITJ), ITJ = 1,2) /0.16854,-0.95999/
 00441: DATA (TNTABLE(36,ITJ), ITJ = 1,2) /0.18407,-0.89999/
 00442: DATA (TNTABLE(37,ITJ), ITJ = 1,2) /0.20046,-0.83999/
 00443: DATA (TNTABLE(38,ITJ), ITJ = 1,2) /0.21770,-0.78000/
 00444: DATA (TNTABLE(39,ITJ), ITJ = 1,2) /0.23577,-0.72000/
 00445: DATA (TNTABLE(40,ITJ), ITJ = 1,2) /0.25463,-0.66000/
 00446: DATA (TNTABLE(41,ITJ), ITJ = 1,2) /0.27426,-0.60000/
 00447: DATA (TNTABLE(42,ITJ), ITJ = 1,2) /0.29460,-0.54000/
 00448: DATA (TNTABLE(43,ITJ), ITJ = 1,2) /0.31562,-0.48000/
 00449: DATA (TNTABLE(44,ITJ), ITJ = 1,2) /0.33725,-0.42000/
 00450: DATA (TNTABLE(45,ITJ), ITJ = 1,2) /0.35943,-0.36000/
 00451: DATA (TNTABLE(46,ITJ), ITJ = 1,2) /0.38209,-0.30000/
 00452: DATA (TNTABLE(47,ITJ), ITJ = 1,2) /0.40517,-0.24000/
 00453: DATA (TNTABLE(48,ITJ), ITJ = 1,2) /0.42858,-0.18000/
 00454: DATA (TNTABLE(49,ITJ), ITJ = 1,2) /0.45224,-0.12000/
 00455: DATA (TNTABLE(50,ITJ), ITJ = 1,2) /0.47608,-0.06000/
 00456: DATA (TNTABLE(51,ITJ), ITJ = 1,2) /0.50000, 0.00000/
 00457: DATA (TNTABLE(52,ITJ), ITJ = 1,2) /0.52392, 0.06000/
 00458: DATA (TNTABLE(53,ITJ), ITJ = 1,2) /0.54776, 0.12000/
 00459: DATA (TNTABLE(54,ITJ), ITJ = 1,2) /0.57142, 0.18000/
 00460: DATA (TNTABLE(55,ITJ), ITJ = 1,2) /0.59483, 0.24000/
 00461: DATA (TNTABLE(56,ITJ), ITJ = 1,2) /0.61791, 0.30000/
 00462: DATA (TNTABLE(57,ITJ), ITJ = 1,2) /0.64057, 0.36000/
 00463: DATA (TNTABLE(58,ITJ), ITJ = 1,2) /0.66275, 0.42000/
 00464: DATA (TNTABLE(59,ITJ), ITJ = 1,2) /0.68438, 0.48000/
 00465: DATA (TNTABLE(60,ITJ), ITJ = 1,2) /0.70540, 0.54000/
 00466: DATA (TNTABLE(61,ITJ), ITJ = 1,2) /0.72574, 0.60000/
 00467: DATA (TNTABLE(62,ITJ), ITJ = 1,2) /0.74537, 0.66000/
 00468: DATA (TNTABLE(63,ITJ), ITJ = 1,2) /0.76423, 0.72000/
 00469: DATA (TNTABLE(64,ITJ), ITJ = 1,2) /0.78230, 0.78000/
 00470: DATA (TNTABLE(65,ITJ), ITJ = 1,2) /0.79954, 0.83999/
 00471: DATA (TNTABLE(66,ITJ), ITJ = 1,2) /0.81593, 0.89999/
 00472: DATA (TNTABLE(67,ITJ), ITJ = 1,2) /0.83146, 0.95999/
 00473: DATA (TNTABLE(68,ITJ), ITJ = 1,2) /0.84613, 1.01999/
 00474: DATA (TNTABLE(69,ITJ), ITJ = 1,2) /0.85992, 1.07999/
 00475: DATA (TNTABLE(70,ITJ), ITJ = 1,2) /0.87285, 1.13999/
 00476: DATA (TNTABLE(71,ITJ), ITJ = 1,2) /0.88492, 1.19999/
 00477: DATA (TNTABLE(72,ITJ), ITJ = 1,2) /0.89616, 1.25998/
 00478: DATA (TNTABLE(73,ITJ), ITJ = 1,2) /0.90657, 1.31998/
 00479: DATA (TNTABLE(74,ITJ), ITJ = 1,2) /0.91620, 1.37998/

```

00480: DATA (TNTABLE(75,ITJ), ITJ = 1,2) /0.92505, 1.43998/
00481: DATA (TNTABLE(76,ITJ), ITJ = 1,2) /0.93318, 1.49998/
00482: DATA (TNTABLE(77,ITJ), ITJ = 1,2) /0.94061, 1.55997/
00483: DATA (TNTABLE(78,ITJ), ITJ = 1,2) /0.94737, 1.61997/
00484: DATA (TNTABLE(79,ITJ), ITJ = 1,2) /0.95351, 1.67997/
00485: DATA (TNTABLE(80,ITJ), ITJ = 1,2) /0.95906, 1.73997/
00486: DATA (TNTABLE(81,ITJ), ITJ = 1,2) /0.96406, 1.79997/
00487: DATA (TNTABLE(82,ITJ), ITJ = 1,2) /0.96854, 1.85996/
00488: DATA (TNTABLE(83,ITJ), ITJ = 1,2) /0.97256, 1.91996/
00489: DATA (TNTABLE(84,ITJ), ITJ = 1,2) /0.97613, 1.97996/
00490: DATA (TNTABLE(85,ITJ), ITJ = 1,2) /0.97934, 2.03996/
00491: DATA (TNTABLE(86,ITJ), ITJ = 1,2) /0.98212, 2.09995/
00492: DATA (TNTABLE(87,ITJ), ITJ = 1,2) /0.98460, 2.15995/
00493: DATA (TNTABLE(88,ITJ), ITJ = 1,2) /0.98678, 2.21994/
00494: DATA (TNTABLE(89,ITJ), ITJ = 1,2) /0.98868, 2.27994/
00495: DATA (TNTABLE(90,ITJ), ITJ = 1,2) /0.99034, 2.33993/
00496: DATA (TNTABLE(91,ITJ), ITJ = 1,2) /0.99179, 2.39993/
00497: DATA (TNTABLE(92,ITJ), ITJ = 1,2) /0.99304, 2.45993/
00498: DATA (TNTABLE(93,ITJ), ITJ = 1,2) /0.99411, 2.51992/
00499: DATA (TNTABLE(94,ITJ), ITJ = 1,2) /0.99504, 2.57992/
00500: DATA (TNTABLE(95,ITJ), ITJ = 1,2) /0.99584, 2.63991/
00501: DATA (TNTABLE(96,ITJ), ITJ = 1,2) /0.99651, 2.69991/
00502: DATA (TNTABLE(97,ITJ), ITJ = 1,2) /0.99709, 2.75990/
00503: DATA (TNTABLE(98,ITJ), ITJ = 1,2) /0.99758, 2.81990/
00504: DATA (TNTABLE(99,ITJ), ITJ = 1,2) /0.99799, 2.87990/
00505: DATA (TNTABLE(100,ITJ), ITJ = 1,2) /0.99834, 2.93989/
00506: DATA (TNTABLE(101,ITJ), ITJ = 1,2) /0.99863, 2.99989/
00507: DATA TRTMOD /'1','2','3','4','5','6','7','8','9','10',
00508: C '11','12','13','14','15','16','17','18','19','20',
00509: C '21','22','23','24','25','26','27','28','29','30',
00510: C '31','32','33','34','35','36','37','38','39','40',
00511: C '41','42','43','44','45','46','47','48','49','50',
00512: C '51','52','53','54','55','56','57','58','59','60',
00513: C '61','62','63','64','65','66','67','68','69','70',
00514: C '71','72','73','74','75','76','77','78','79','80',
00515: C '81','82','83','84','85','86','87','88','89','90',
00516: C '91','92','93','94','95','96','97','98','99'/
00517: JF = 0
00518: NI = 1
00519: C NBT DOESN'T NEED TO BE VARYING
00520: NBT = 987
00521: DO 1111 I = 1,234
00522: KONT (I) = 0
00523: NWKONT (I) = 0
00524: STAR (I) = ' '
00525: TRACT(I) = ' '
00526: TRACTIN(I) = ' '
00527: 1111 CONTINUE
00528: TRACTIN2 = ' '
00529: TMODN = ' '
00530: NX=0
00531: TM2=.01
00532: TM4=.0001
00533: TM6=.000001
00534: ITR=4999
00535: IMETCNT = 0
00536: ITFLAG = 0
00537: ISCR12 = 1
00538: IMODCT = 0
00539: NPFLAG = 0

```

```

00540:      MCNT = 0
00541:      DO 7654 J = 1,101
00542:      DO 7654 I = 1,2
00543:      XNORE(I,J) = TTABLE(J,I)
00544:7654  CONTINUE
00545:      ELOPT = ' '
00546:      TOTF = ' '
00547:      WRITE (1,*) (' ')
00548:      WRITE (1,*) 'ENTER "M" TO DISPLAY LISTING OF AVAILABLE MODELS.'
00549:      WRITE (1,*) 'ENTER "Q" TO END PROGRAM.'
00550:      WRITE (1,*) 'OTHERWISE PRESS C/R TO CONTINUE:'
00551:      READ (1,137) ELOPT
00552:      IF (ELOPT.NE.'M') GO TO 11
00553:      IF (ELOPT.EQ.'Q') GO TO 9346
00554:      WRITE (1,*) ' '
00555:      DO 31 IREADT = 1,98
00556:      WRITE (1,1212) MODELTS(IREADT)
00557:1212  FORMAT (A80)
00558:31    CONTINUE
00559:      WRITE (1,*) (' ')
00560:      WRITE (1,*) (' ')
00561:11    CONTINUE
00562:      MODN(1) = 0
00563:      READPW = ' '
00564:      PRINTFILE = ' '
00565:      ELOPT = ' '
00566:      WRITE (1,*) 'NEED A LISTING OF DATA WITH MODEL/METAL IDS?'
00567:1211  WRITE (1,*) 'ENTER "P" TO CREATE A PRINTABLE FILE'
00568:      WRITE (1,*) 'ENTER "Q" TO END PROGRAM.'
00569:      WRITE (1,*) 'OTHERWISE PRESS C/R TO CONTINUE:'
00570:      READ (1,137) ELOPT
00571:      IF (ELOPT.EQ.'Q') GO TO 9346
00572:      IF (ELOPT.NE.'P') GO TO 22
00573:1213  WRITE (1,*) ' '
00574:      WRITE (1,*) 'ENTER MODEL(INDEX) NUMBER OF FILE TO BE PRINTED'
00575:      READ (1,*) MODN(1)
00576:      TEMPDF = DATAFILES(MODN(1))
00577:      IF (MODN(1).LT.1.OR.MODN(1).GT.99.OR.TEMPDI.EQ.'N-A')
00578:      C WRITE (1,1214) MODN(1)
00579:1214  FORMAT (/, ' MODEL ',I2, ' NOT AVAILABLE.',/)
00580:      IF (MODN(1).LT.1.OR.MODN(1).GT.99.OR.TEMPDI.EQ.'N-A') GO TO 121
00581:1215  DO 1112 ICM=1,10
00582:      PMETAL(ICM) = ' '
00583:1112  CONTINUE
00584:      IF (MODN(1).NE.87.AND.MODN(1).NE.88) GO TO 1616
00585:      WRITE (1,*) ' '
00586:      WRITE (1,*) ('THIS FILE IS NOT AVAILABLE WITHOUT A PASSWORD')
00587:      WRITE (1,*) ('ENTER PASSWORD, OTHERWISE PRESS C/R TO CONTINUE:')
00588:      READ (1,12) READPW
00589:12    FORMAT (A3)
00590:      IF (READPW.EQ.'#!&*@') GO TO 1616
00591:      READPW = 'Q'
00592:      IF (READPW.NE.'#!&*@') GO TO 1211
00593:C
00594:C      ENTER FILENAME, OPEN DATA FILE, AND LIST FILE
00595:C
00596:16    FORMAT(A32)
00597:22    CONTINUE
00598:      WRITE (1,*)('PLEASE TYPE INITIAL INFORMATION FOR HEADER PAGE -
00599:      WRITE (1,*) ('REQUESTED BY: ')

```

```

00600:      READ (1,405) REQNAME
00601:405    FORMAT (A65)
00602:      WRITE (1,*) ('ASSESSMENT AREA: ')
00603:      READ (1,405) AREANM
00604:      WRITE (1,*) ('DATE REQUESTED: ')
00605:      READ (1,405) TODAYDT
00606:      WRITE (1,*) ('FOR TRACT TOTALS, ENTER "T"')
00607:      WRITE (1,*) ('FOR GRAND TOTAL, ENTER "G"')
00608:      READ (1,137) TOTF
00609:      OPEN (UNIT=18,STATUS='SCRATCH')
00610:      WRITE (1,*) (' ')
00611:      WRITE (1,*) ('LIKE A ".CPL" FILE CREATED ?')
00612:      READ (1,137) CPLOP
00613:      IF (CPLOP.NE.'Y'.AND.CPLOP.NE.'y') GO TO 317
00614:      CLOSE (18)
00615:      CPLINE = ' '
00616:      CPLFILE = ' '
00617:      WRITE (1,*) ('ENTER NAME FOR CPL FILE:')
00618:      READ (1,402) CPLFILE
00619:      LENCAT = INDEX (CPLFILE,' ')
00620:      CPLINE = CPLFILE
00621:      CPLINE (LENCAT:LENCAT+4) = '.CPL '
00622:      CPLFILE = CPLINE
00623:      OPEN (UNIT=18,FILE=CPLFILE,STATUS='NEW')
00624:      CPLINE = '3DATA R MARK3'
00625:      WRITE (18,16) CPLINE
00626:      WRITE (18,*) 'X'
00627:      WRITE (18,*) 'X'
00628:      WRITE (18,*) REQNAME
00629:      WRITE (18,*) AREANM
00630:      WRITE (18,*) TODAYDT
00631:      WRITE (18,137) TOTF
00632:      WRITE (18,*) 'N'
00633:317    WRITE (*,318)
00634:318    FORMAT (/,' ENTER NUMBER OF MODELS FOR THIS RUN:')
00635:      READ (1,1718) ITOTMOD
00636:      NEWTM = ITOTMOD
00637:      WRITE (18,1718) ITOTMOD
00638:      IMINS = 0
00639:      IMINS = (ITOTMOD*15)
00640:1718   FORMAT(I4)
00641:      WRITE (1,*) (' ')
00642:      WRITE (1,*) ('A PROMPT APPEARS FOR EACH MODEL.')
00643:      WRITE (1,*) (' ')
00644:1719   DO 84018 IMN = 1,ITOTMOD
00645:      IF (ITOTMOD.EQ.1)
00646:      C WRITE (1,*) ('ENTER SINGLE MODEL(INDEX) NUMBER:')
00647:      IF (IMN.EQ.ITOTMOD.AND.ITOTMOD.GT.1)
00648:      C WRITE (1,*) ('ENTER LAST MODEL(INDEX) NUMBER:')
00649:      IF (IMN.LT.ITOTMOD) WRITE (1,*) ('ENTER MODEL(INDEX) NUMBER:')
00650:      IF (TOTF.EQ.'G') READ (1,*) MODN(IMN)
00651:      IF (TOTF.EQ.'G') GO TO 1725
00652:      IF (TOTF.EQ.'T') READ (1,1722) TRACTIN(IMN)
00653:1722   FORMAT (A12)
00654:      TRACTIN2 = ' '
00655:      TRACTIN2 = TRACTIN(IMN)
00656:      LENCAT = INDEX (TRACTIN2,' ')
00657:      TMODN = TRACTIN2 (1:LENCAT-1)
00658:      DO 1724 ITM = 1,99
00659:      IF (TMODN.EQ.(TRTMOD(ITM))) MODN(IMN) = ITM

```

```

00660:1724  CONTINUE
00661:1725  TEMPDF = DATAFILES(MODN(IMN))
00662:      IF (MODN(IMN).LT.1.OR.MODN(IMN).GT.99.OR.TEMPDF.EQ.'N-A')
00663:      C WRITE (1,1214) MODN(IMN)
00664:      IF (MODN(IMN).LT.1.OR.MODN(IMN).GT.99.OR.TEMPDF.EQ.'N-A')
00665:      C ITOTMOD = ITOTMOD-1
00666:      IF (MODN(IMN).LT.1.OR.MODN(IMN).GT.99.OR.TEMPDF.EQ.'N-A')
00667:      C NEWTM = NEWTM - 1
00668:      IF (MODN(IMN).LT.1.OR.MODN(IMN).GT.99.OR.TEMPDF.EQ.'N-A')
00669:      C GO TO 84018
00670:      IF (MODN(IMN).NE.87.AND.MODN(IMN).NE.88) GO TO 1720
00671:      WRITE (1,*) ' '
00672:      WRITE (1,*) ('THIS FILE IS NOT AVAILABLE WITHOUT A PASSWORD')
00673:      WRITE (1,*) ('ENTER PASSWORD, OTHERWISE PRESS C/R TO CONTINUE:')
00674:      READ (1,12) READPW
00675:      IF (READPW.EQ.'#18*a') GO TO 1720
00676:      IF (READPW.NE.'#18*a') GO TO 1719
00677:1720  IF (TOTF.EQ.'G') WRITE (18,*) MODN(IMN)
00678:      IF (TOTF.EQ.'T')
00679:      C WRITE (18,1722) TRACTIN2
00680:84018  CONTINUE
00681:      WRITE (*,418)
00682:418  FORMAT (/,' ENTER UNIQUE FILE NAME FOR MARK3 OUTPUT RESULTS: ')
00683:      READ (1,16) DPFILE
00684:      WRITE (18,16) DPFILE
00685:      DO 1818 I18 = 1,ITOTMOD
00686:      ELOPT = ' '
00687:      WRITE (1,518) MTITLES(MODN(I18))
00688:518  FORMAT (/,' TERMINAL INPUT FOR MODEL ',A65)
00689:      WRITE (1,*) ' '
00690:519  WRITE (1,*) ('ENTER "L" FOR LOGNORMAL')
00691:      WRITE (1,*) ('ENTER "E" FOR EMPIRICAL')
00692:      READ (1,137) ELOPT
00693:      IF (ELOPT.NE.'L'.AND.ELOPT.NE.'E') GO TO 519
00694:      WRITE (18,137) ELOPT
00695:      DO 510 IDP = 1,6
00696:      ND(IDP) = 0
00697:510  CONTINUE
00698:520  WRITE (1,*) (' ')
00699:      WRITE (1,*) ('ENTER EST. # OF DEP. COUNT(3 OR 5): ')
00700:      READ (1,1519) INOD
00701:      IF (INOD.NE.3.AND.INOD.NE.5) GO TO 520
00702:      WRITE (18,1519) INOD
00703:      IF (INOD.EQ.5) GO TO 521
00704:      WRITE (1,*) ' '
00705:      WRITE (1,*) ('ENTER ESTIMATED NUMBER OF DEPOSITS(3) - ')
00706:      WRITE (1,*) ('          90%          50%          10% ')
00707:      WRITE (1,*) ('          ND(2)          ND(3)          ND(4)')
00708:      WRITE (1,*) (' ')
00709:      WRITE (1,*) ('90% =')
00710:      READ (1,1518) ND(2)
00711:      WRITE (18,1518) ND(2)
00712:      WRITE (1,*) ('50% =')
00713:      READ (1,1518) ND(3)
00714:      WRITE (18,1518) ND(3)
00715:      WRITE (1,*) ('10% =')
00716:      READ (1,1518) ND(4)
00717:      WRITE (18,1518) ND(4)
00718:1518  FORMAT (I2)
00719:1519  FORMAT (I1)

```

```

00720:      WRITE (1,*) (' ')
00721:      IF (INOD.EQ.3) GO TO 522
00722:521    WRITE (1,*) (' ')
00723:      WRITE (1,*) ('ENTER ESTIMATED NUMBER OF DEPOSITS(5) - ')
00724:      WRITE (1,*) (' 95%      90%      50%      10%      05%')
00725:      WRITE (1,*) ('ND(2)    ND(3)    ND(4)    ND(5)    ND(6)')
00726:      WRITE (1,*) (' ')
00727:      WRITE (1,*) ('95% =')
00728:      READ (1,1518) ND(2)
00729:      WRITE (18,1518) ND(2)
00730:      WRITE (1,*) ('90% =')
00731:      READ (1,1518) ND(3)
00732:      WRITE (18,1518) ND(3)
00733:      WRITE (1,*) ('50% =')
00734:      READ (1,1518) ND(4)
00735:      WRITE (18,1518) ND(4)
00736:      WRITE (1,*) ('10% =')
00737:      READ (1,1518) ND(5)
00738:      WRITE (18,1518) ND(5)
00739:      WRITE (1,*) ('05% =')
00740:      READ (1,1518) ND(6)
00741:      WRITE (18,1518) ND(6)
00742:      WRITE (1,*) (' ')
00743:522    WRITE (1,*) ('TO FIX THE PROBABILITY OF ZERO DEPOSITS,')
00744:      WRITE (1,*) ('ENTER PROBABILITY BETWEEN 0. & 1.')
00745:      WRITE (1,*) ('OR ENTER -1. FOR DEFAULT.')
00746:      READ (1,*) PROBZ
00747:      WRITE (18,*) PROBZ
00748:1818   CONTINUE
00749:      IF (CPLOP.NE.'Y'.AND.CPLOP.NE.'y') GO TO 1819
00750:      CPLINE = '&END'
00751:      WRITE (18,16) CPLINE
00752:      CLOSE (13)
00753:      WRITE (1,9342) CPLFILE
00754:9342    FORMAT (/,'   CPL FILE CREATED IS - ',A32,/)
00755:      GO TO 9346
00756:1819   REWIND (18)
00757:      IF (CPLOP.EQ.'Y') READ (18,137) TOTF
00758:      READ (18,17) ITOTMOD
00759:17      FORMAT(I4)
00760:      ITOTMOD = NEWTM
00761:      DO 840 IMN = 1,ITOTMOD
00762:      IF (TOTF.EQ.'G') READ (18,*) MODN(IMN)
00763:      IF (TOTF.EQ.'G') GO TO 840
00764:      TRACTIN2 = ' '
00765:      IF (TOTF.EQ.'T') READ (18,1722) TRACTIN(IMN)
00766:      TRACTIN2 = TRACTIN(IMN)
00767:      LENCAT = INDEX (TRACTIN2, ' ')
00768:      TMODN = TRACTIN2 (1:LENCAT-1)
00769:      DO 1723 ITM = 1,99
00770:      IF (TMODN.EQ.(TRTMOD(ITM))) MODN(IMN) = ITM
00771:1723    CONTINUE
00772:      TRACTIN2 = TRACTIN2(LENCAT+1:)
00773:      LENCAT = J
00774:      LENCAT = INDEX (TRACTIN2, ' ')
00775:      STAR(IMN) = TRACTIN2 (1:LENCAT-1)
00776:      TRACTIN2 = TRACTIN2 (LENCAT+1:)
00777:      LENCAT = 0
00778:      LENCAT = INDEX (TRACTIN2, ' ')
00779:      TRACT(IMN) = TRACTIN2 (1:LENCAT-1)

```

```

00780:840    CONTINUE
00781:841    IMN = IMN -1
00782:      READ (18,16) DPFIL
00783:      ITOTSV = ITOTMOD
00784:      IF (ITOTMOD.EQ.1) IMODCT = 1
00785:      IF (ITOTMOD.EQ.1) ITOTMOD = 2
00786:      IMETAL=1
00787:      IPMESS = 0
00788:      DATA (ZERO(I),I=1,35) /35*0.0/
00789:      IF (ITOTMOD.EQ.1) IMETAL = 2
00790:      WRITE (1,*) (' ')
00791:      WRITE (1,66)
00792:66      FORMAT (' TERMINAL INPUT COMPLETE.',/)
00793:      WRITE (1,*) ('ALLOW 15 MINUTES FOR EACH MODEL -')
00794:      WRITE (1,667) IMINS
00795:667      FORMAT (' THIS RUN WILL TAKE APPROX. ',I4,' MINUTES.')
```

```

00796:      WRITE (*,668)
00797:668      FORMAT (' (DEPENDS GREATLY ON VALUES ENTERED)',/)
00798:      WRITE (1,*) ('FORMATTING FILES...')
```

```

00799:      DO 5 I = 1,35
00800:      IPMESS = IPMESS + 1
00801:      IF (IPMESS.LT.14) GO TO 6
00802:      IPMESS = 0
00803:6      DO 5 J=1,ITR
00804:5      SMETAL(J,I)=0.
00805:      OPEN (UNIT=UNITMP,FILE=DPFIL,STATUS='NEW')
00806:      WRITE (UNITMP,*) (' ')
00807:      WRITE (UNITMP,409)
00808:409      FORMAT (//////////,20X,'MARK3 SIMULATION RESULTS')
00809:      WRITE (UNITMP,410) REQNAME
00810:410      FORMAT (////////,14X,'REQUESTED BY:',13X,A65)
00811:      WRITE (UNITMP,411) AREANM
00812:411      FORMAT (////,11X,'ASSESSMENT AREA:',13X,A65)
00813:      WRITE (UNITMP,412) TODAYDT
00814:412      FORMAT (////,12X,'DATE REQUESTED:',13X,A65,'^214')
```

```

00815:C
00816:C      INITIALIZE VARIABLES, THEN READ INFORMATION
00817:C
00818:      OPEN (UNIT=UNITSC1,STATUS='SCRATCH')
00819:      OPEN (UNIT=UNITSC2,STATUS='SCRATCH')
```

```

00820:      DO 18 J = 1,ITR
00821:      WRITE (UNITSC1,45) (ZERO(I),I=1,35)
00822:45      FORMAT (5X,35(E11.5))
00823:18      CONTINUE
00824:      REWIND (11)
00825:      OPEN (UNIT=17,STATUS='SCRATCH')
```

```

00826:      MDI = 0
00827:      IMODN = 1
00828:      DATA (ERR(IERR),IERR=1,12) /12*0.0/
00829:50      DO 1313 I=1,ITR
00830:      DO 1313 J=1,35
00831:      SMETAL (I,J) = 0.
00832:1313      CONTINUE
00833:      ISW=1.
00834:      ISZ=1.
00835:      IS=1.
00836:      IZDFLAG = 0
00837:      DINFILE = DATAFILES(MODN(IMODN))
00838:      IMODN = IMODN + 1
00839:      EN=0.
```

```

00840:      MDI = MDI + 1
00841:      IMODCT = IMODCT + 1
00842:      IF (IMODCT.GT.ITOTMOD) GO TO 770
00843:      DO 1234 I = 1,234
00844:      KONT (I) = 0
00845:      NWKONT (I) = 0
00846:1234  CONTINUE
00847:      AREA = 5000.00
00848:      TYPE = 1
00849:      EN = 1.0000
00850:      WRITE (1,*) (' ')
00851:      II = MODN (MDI)
00852:      IF (ELOPT.NE.'P') GO TO 170
00853:1616  CONTINUE
00854:      DINFILE = ' '
00855:      PRINTFILE = ' '
00856:      DINFILE = DATAFILES(MODN(1))
00857:726   FORMAT (A240)
00858:      LENCAT = INDEX (DINFILE,' ')
00859:      PRINTFILE = DINFILE
00860:      PRINTFILE (LENCAT:LENCAT+5) = '.DATA '
00861:      DINFILE = DATAFILES(MODN(1))
00862:      OPEN (UNIT=7,FILE=DINFILE)
00863:      OPEN (UNIT=6,FILE=PRINTFILE)
00864:      DO 2424 IPMET = 1,NUMOFMTL(MODN(1)) + 1
00865:      PMETAL(IPMET) = CMETAL(IEMPCOD(MODN(1),IPMET))
00866:2424  CONTINUE
00867:      WRITE (6,725) MTITLES(MODN(1))
00868:725   FORMAT (/8X,'METALS IN ',A65,/)
00869:      WRITE (6,727) PMETAL(1),PMETAL(2),PMETAL(3),PMETAL(4),PMETAL(5),
00870:      C      PMETAL(6),PMETAL(7),PMETAL(8),PMETAL(9),PMETAL(10)
00871:727   FORMAT ('      DEPOSITS',5X,'COUNTRY',
00872:      C      6X,A10,A10,A10,5X,A10,6X,A10,4X,A12,
00873:      C      4X,A10,4X,A10,4X,A10,4X,A10,/)
00874:1717  CONTINUE
00875:      READ (7,726,END=167) DATALINE
00876:      DATALINE = ' ' // DATALINE
00877:      WRITE (6,726) DATALINE
00878:166   GO TO 1717
00879:167   CLOSE (7)
00880:      CLOSE (6)
00881:      WRITE (1,169) PRINTFILE
00882:      ELOPT = ' '
00883:      IF (MODN(1).LT.1.OR.MODN(1).GT.99.OR.
00884:      C DATAFILES(MODN(1)).EQ.'N-A') GO TO 1699
00885:169   FORMAT (/,' PRINTABLE DATAFILE CREATED CALLED ',A80,/)
00886:1699  WRITE (1,*) 'WANT TO PRINT ANOTHER FILE? ("Y" OR "N"):'
00887:      READ (1,137) ELOPT
00888:      IF (ELOPT.EQ.'Y') GO TO 1213
00889:      WRITE (1,*) ' '
00890:      GO TO 9346
00891:170   CONTINUE
00892:      NAM = 'MODEL '
00893:70    FORMAT(15X,A6,I2)
00894:      IF(TYPE.EQ.1.OR.TYPE.EQ.2)GO TO 80
00895:      WRITE(1,75)
00896:75    FORMAT(' INVALID MODEL TYPE! AREA NAME IGNORED = ',A)
00897:      GO TO 50
00898:80    CONTINUE
00899:90    FORMAT(1X,'DEPOSIT MODEL TYPE USED TO EVALUATE THIS AREA IS ',I2.

```

```

00900:100  FORMAT(1X,'ESTIMATE OF THE EXPECTED NUMBER OF SIGNIFICANT',/,1X,
00901: 1 'DEPOSITS OF THIS TYPE THAT EXIST IN THIS REGION IS ',F4.1)
00902:C
00903:C      CHECK MODEL NUMBER FOR ACCEPTABILITY
00904:C
00905:120  IF(II.LT.1.OR.II.GT.99)GO TO 140
00906:125  CONTINUE
00907:      IF(II.EQ.1.AND.EN.EQ.0.)EN=AREA/25.
00908:      IF(II.NE.1.AND.EN.EQ.0.)EN=AREA/ITR
00909:      IF(TYPE.EQ.2)EN=EN/10.
00910:      METNUM = NUMOFMTL(II)
00911:      NUMCOM = METNUM + 1
00912:      GA=0.
00913:      GB=0.
00914:      GC=0.
00915:      GD=0.
00916:      DO 131 IMT = 1,10
00917:      METN(IMT) = 0
00918:131  CONTINUE
00919:      DO 135 IMT = 1,10
00920:      IECOD = IEMPCOD(II,IMT)
00921:      METN(IMT) = IEMPCOD(II,IMT)
00922:      IF (IECOD.EQ.0) GO TO 136
00923:135  CONTINUE
00924:136  IMT = IMT-2
00925:      DO 133 IMTT = 1,45
00926:      METTEM(IMTT) = 0
00927:133  CONTINUE
00928:      DO 134 IMTT = 1,10
00929:      IF (METN(IMTT).LT.1.OR.METN(IMTT).GT.46) GO TO 134
00930:      METTEM(METN(IMTT)) = IMTT
00931:134  CONTINUE
00932:      ELOPT = ' '
00933:      READ (18,137) ELOPT
00934:      IF (ELOPT.NE.'L') ELOPT = 'E'
00935:137  FORMAT(A1)
00936:      ELNLABEL = ' '
00937:      READ (18,1519) INOD
00938:      IF (ELOPT.EQ.'E') ELNLABEL = 'EMPIRICAL '
00939:      IF (ELOPT.EQ.'L') ELNLABEL = 'LOGNORMAL '
00940:      IF (ELOPT.EQ.'E') GO TO 666
00941:      IF (ELOPT.EQ.'L') GO TO 777
00942:COMMENT      WRITE (1,*) ' '
00943:C
00944:C      UNACCEPTABLE MODEL NUMBER
00945:C
00946:140  WRITE(1,150)II
00947:150  FORMAT(/,1X,'NO DEPOSIT MODEL OF TYPE ',I2,' EXISTS IN THE FILE'
00948:159  WRITE (1,160)
00949:160  FORMAT (' ENTER A VALUE BETWEEN 1 AND 66: ')
00950:      READ (1, * ) INUM
00951:180  II=INUM
00952:      IF(II.LT.1.OR.II.GT.99)GO TO 140
00953:      GO TO 125
00954:C ^214
00955:C
00956:C      LOGNORMAL DISTRIBUTION ROUTINE
00957:C
00958:777  CONTINUE
00959:      IF (ND(2).EQ.0.AND.ND(3).EQ.0.AND.

```

```

00960:      C ND(4).EQ.0.AND.ND(5).EQ.0.AND.
00961:      C ND(6).EQ.0) WRITE (UNITMP,1346)
00962:      CALL LOGNORMAL (II,ELOPT,ELNLABEL)
00963:      IF (ELOPT.NE.'L') GO TO 666
00964:      PROBZ = 0.
00965:      MCNT = MCNT + 1
00966:      READ (18,15) ND(2)
00967:      READ (18,15) ND(3)
00968:      READ (18,15) ND(4)
00969:      IF (INOD.EQ.3) GO TO 1347
00970:      READ (18,15) ND(5)
00971:      READ (18,15) ND(6)
00972:1347  READ (18,*) PROBZ
00973:      WRITE (1,5177) MCNT, II,ELNLABEL
00974:5177  FORMAT (' ',I3,' - ','PROCESSING MODEL ',I2,' - ', A12)
00975:      IF (INOD.EQ.3)
00976:      C WRITE (UNITMP,1345) ELNLABEL,ND(2),ND(3),ND(4),MTITLES(II)
00977:      IF (INOD.EQ.5) WRITE (UNITMP,1698)
00978:      C ELNLABEL,ND(2),ND(3),ND(4),ND(5),ND(6),MTITLES(II)
00979:1345  FORMAT ('5214',16X,A10,'('',I2,'','','I2,'','','I2,'')',1X,' ',
00980:      C ',11X,A65,/)
00981:1348  DO 1350 I=1,ITR
00982:      DO 1350 J=1,35
00983:      SMETAL (I,J) = 0.
00984:1350  CONTINUE
00985:      DO 1351 I = 1,234
00986:      NWKONT (I) = 0
00987:1351  CONTINUE
00988:      DO 2211 IMN = 1,10
00989:      IECOD = IEMPCOD(II,IMN)
00990:      METN(IMN) = IEMPCOD(II,IMN)
00991:      IF (IECOD.EQ.0) GO TO 2233
00992:2211  CONTINUE
00993:2233  IMN = IMN - 2
00994:      IZDFLAG = 0
00995:      IPMESS = 0
00996:      DO 13854 LL = 1,ITR
00997:COMMENT      CALL DISTRIB(NX)
00998:      IF (INOD.EQ.3) CALL DISTRIB3(NX)
00999:      IF (INOD.EQ.5) CALL DISTRIB5(NX)
01000:COMMENT      NX = 1      THIS IS USED FOR PROBLEM CHECKING ONLY
01001:COMMENT      NX = 1
01002:      KONT (NX + 1) = KONT (NX + 1) + 1
01003:      NWKONT (NX + 1) = NWKONT (NX + 1) + 1
01004:      IF (NX.LT.1) GO TO 13854
01005:      DO 13853 KK = 1,NX
01006:      DUM = 0.
01007:      CALL URNMX (NBT,NI,RR)
01008:      DO 1 I = 1,NSUITE(1)
01009:      DUM = DUM + SPROB (1,I)
01010:      IF (DUM.GT.RR) GO TO 2
01011:      GO TO 1
01012:2      JS = I
01013:      GO TO 3
01014:1      CONTINUE
01015:      JS = NSUITE (1)
01016:3      DO 4 I = 1,NUMCOM
01017:      CALL URNMX (NBT,NI,RR)
01018:      CALL NORVALE (RR,X(I))
01019:4      CONTINUE

```

```

01020:      DO 1355 I = 1,NUMCOM
01021:      Z(I) = 0.
01022:      DO 1366 J = 1,I
01023:      Z(I) = Z(I) + AEMP(1,JS,I,J) * X(J)
01024:1366  CONTINUE
01025:1355  CONTINUE
01026:      DO 1367 I=1,NUMCOM
01027:      Z(I) = EXP(ZM(JS,I) + ZD(JS,I) * Z(I))
01028:1367  CONTINUE
01029:COMMENT      IF (JS.EQ.3) WRITE (1,6767) Z(1),Z(2),Z(3),Z(4),Z(5),Z(6)
01030:6767  FORMAT (6(E12.4))
01031:      SMETAL(LL,19)=SMETAL(LL,19)+Z(1)
01032:COMMENT      CALL QUANTILE (Z(1),JS,METTEM(19))
01033:      DO 999 L=2,NUMCOM
01034:C      ZD(JS,L) = -1.      'IF METAL IS ABSENT'
01035:      IF (ZD(JS,L).LT.-0.5) GO TO 999
01036:      TTTT=Z(1)*Z(L) / 100
01037:      SMETAL(LL,METN(L))=SMETAL(LL,METN(L))+TTTT
01038:COMMENT      CALL QUANTILE (Z(L),JS,L)
01039:      999 CONTINUE
01040:13853  CONTINUE
01041:      IPMESS = 0
01042:13854  CONTINUE
01043:      IZDFLAG = 0
01044:      IMETCNT=IMETCNT+1
01045:      IF (ITOTMOD.GT.1) GO TO 55555
01046:1399  GO TO 50
01047:C ^214
01048:C
01049:C      EMPIRICAL DISTRIBUTION ROUTINE
01050:C
01051:666  CONTINUE
01052:402  FORMAT (A60)
01053:C
01054:C      UNITS 7 & 8 HAVE BEEN OPENED IN EMPIRICAL
01055:C
01056:      IF (ND(2).EQ.0.AND.ND(3).EQ.0.AND.
01057:  C ND(4).EQ.0.AND.ND(5).EQ.0.AND.
01058:  C ND(6).EQ.0) WRITE (UNITMP,1346)
01059:1346  FORMAT (/,,6X,'NON ZERO DEPOSITS INDICATED')
01060:      MODNUM = II
01061:      IMETCNT = IMETCNT + 1
01062:      NGRID=50
01063:      GRID=NGRID
01064:      MCNT = MCNT + 1
01065:      N=1
01066:      DO 221 I = 1,10
01067:      IECOD = IEMPCOD(II,I)
01068:      METN(I) = IEMPCOD(II,I)
01069:      IF (IECOD.EQ.0) GO TO 223
01070:221  CONTINUE
01071:223  INFOC(1) = I-1
01072:      READ (18,15) ND(2)
01073:      READ (18,15) ND(3)
01074:      READ (18,15) ND(4)
01075:      IF (INOD.EQ.3) GO TO 1617
01076:      READ (18,15) ND(5)
01077:      READ (18,15) ND(6)
01078:15  FORMAT (I2)
01079:1617  READ (18,*) PROBZ

```

```

01080:      IF (INOD.EQ.3)
01081:      C WRITE (UNITMP,1345) ELNLABEL,ND(2),ND(3),ND(4),MTITLES(II)
01082:      IF (INOD.EQ.5) WRITE (UNITMP,1698)
01083:      C ELNLABEL,ND(2),ND(3),ND(4),ND(5),ND(6),MTITLES(II)
01084:1698  FORMAT ('^214',11X,A10,'(',I2,',',I2,',',I2,',',I2,',',I2,',')',
01085:      C 1X,' ',/,16X,A65,/)
01086:      IZDFLAG = 1
01087:      WRITE (1,517) MCNT, II,ELNLABEL
01088:517  FORMAT (' ',I3,' - ', 'PROCESSING MODEL ',I2, ' - ', A12)
01089:      DO 233 METQ = 1,10
01090:      DO 220,KKK=1,12
01091:      KKONT(KKK,METQ) = 0
01092:220  CONTINUE
01093:      DO 230 III = 1,3
01094:      DO 231 IJI = 1,100
01095:      DO 232 JJ = 1,12
01096:      CLINE(III,IJI,JJ,METQ) = 0.
01097:232  CONTINUE
01098:231  CONTINUE
01099:230  CONTINUE
01100:233  CONTINUE
01101:      CALL EMPIRICAL (II)
01102:C      SIMUINSE ROUTINE BEGINS
01103:      DO 111 LL=1,ITR
01104:COMMENT      CALL DISTRIB(NX)
01105:1100  IF (INOD.EQ.3) CALL DISTRIB3(NX)
01106:      IF (INOD.EQ.5) CALL DISTRIB5(NX)
01107:COMMENT      IF (MCNT.EQ.3) NX = 9
01108:COMMENT      NX = 1      THIS IS USED FOR PROBLEM CHECKING ONLY
01109:COMMENT      NX = 1
01110:      KONT(NX+1)=KONT(NX+1)+1
01111:      NWKONT(NX+1)=NWKONT(NX+1)+1
01112:      IF(NX.LT.1) GO TO 111
01113:      DO 456 INDX = 1,NX
01114:      CALL URNMX(NBT,NI,EST)
01115:      TEST=0.
01116:      DO 2222 J=1,NSUITE(1)
01117:      TEST=TEST+SPROB(1,J)
01118:      IF(EST.GT.TEST) GO TO 2222
01119:      JS=J
01120:      GO TO 333
01121:2222  CONTINUE
01122:      WRITE(1,444) EST, TEST, (SPROB(1,K),K=1,NSUITE(1))
01123:444  FORMAT('SUITE TROUBLE ',14(F8.4,1X))
01124:      JS = NSUITE(1)
01125:333  DO 555 L=1,IC
01126:      CALL URNMX(NBT,NI,DUM)
01127:      CALL NORVALE(DUM,XZ(L))
01128:555  CONTINUE
01129:      DO 6666 L=1,IC
01130:      VAL=0.
01131:      DO 7777 J=1,L
01132:      VAL=VAL+AEMP(1,JS,L,J)*XZ(J)
01133:7777  CONTINUE
01134:      DO 8888 K=1,IDIST
01135:      AA(K)=FEMP(1,JS,K,L)
01136:8888  CONTINUE
01137:      CALL NORPROBE(DUM,VAL)
01138:      CALL EMPVAL(DUM,AA,Z(L))
01139:6666  CONTINUE

```

```

01140:      SMETAL(LL,19)=SMETAL(LL,19)+Z(1)
01141:COMMENT      CALL QUANTILE (Z(1),JS,METTEM(19))
01142:      DO 9992 L=2,NUMCOM
01143:      TTTT=Z(1)*Z(L) / 100
01144:      SMETAL(LL,METN(L))=SMETAL(LL,METN(L))+TTTT
01145:COMMENT      CALL QUANTILE (Z(L),JS,L)
01146: 9992 CONTINUE
01147:456  CONTINUE
01148: 111 CONTINUE
01149:C
01150:C      SIMUINSE ROUTINE ENDS
01151:C
01152:      IF (ITOTMOD.GT.1) GO TO 55555
01153:      GO TO 50
01154:987  WRITE (1, * ) ('MODEL NOT AVAILABLE.')
01155:      GO TO 9341
01156:C ^214
01157:C
01158:C      ***  SINGLE MODELS  ***
01159:C
01160:55555 DO 2780 I=1,35
01161:      SU(I)=0.
01162:      DO 2780 J=1,ITR
01163:2780  SU(I)=SU(I) + SMETAL(J,I)
01164:      IF (ISCR12.EQ.2) GO TO 2785
01165:      CLOSE (12)
01166:      OPEN (UNIT=UNITSC2,STATUS='SCRATCH')
01167:      GO TO 2787
01168:2785  CLOSE (11)
01169:      OPEN (UNIT=UNITSC1,STATUS='SCRATCH')
01170:2787  DO 2786 J = 1,ITR
01171:      IF (ISCR12.EQ.1) READ (UNITSC1,2788) (TMPARR(I),I=1,35)
01172:      IF (ISCR12.EQ.2) READ (UNITSC2,2788) (TMPARR(I),I=1,35)
01173:2788  FORMAT (5X,35(E11.5))
01174:      DO 2789 I = 1,35
01175:      TMPARR(I) = SMETAL(J,I) + TMPARR(I)
01176:2789  CONTINUE
01177:      IF (ISCR12.EQ.1) WRITE (UNITSC2,2788) (TMPARR(I),I=1,35)
01178:      IF (ISCR12.EQ.2) WRITE (UNITSC1,2788) (TMPARR(I),I=1,35)
01179:2786  CONTINUE
01180:      REWIND (11)
01181:      REWIND (12)
01182:      IF (ISCR12.EQ.1) ISCR12 = 22
01183:      IF (ISCR12.EQ.2) ISCR12 = 1
01184:      IF (ISCR12.EQ.22) ISCR12 = 2
01185:      CALL SRT(ITR,IMETAL)
01186:      TOT = 0.
01187:      DO 2890 J=1,35
01188:      IF(SU(J).EQ.0.)GO TO 2890
01189:      IFIRST = ITR
01190:      DO 2790 I = 1,ITR
01191:      IF (SMETAL(I,J).LE.0.) GO TO 2790
01192:      IFIRST = I
01193:      GO TO 2794
01194:2790  CONTINUE
01195:2794  JORDER(1) = IFIRST
01196:      DO 2792 I = 2,10
01197:      JORDER(I) = ((ITR-IFIRST + 2) * (I-1))/10 + IFIRST-1
01198:2792  CONTINUE
01199:      JORDER(11)=((ITR-IFIRST+2)*95)/100+IFIRST-1

```

```

01200:      JORDER(12)=((ITR-IFIRST+2)*99)/100+IFIRST-1
01201:      JORDER(13) = ITR
01202:      WRITE(UNITMP,9310)CMETAL(J)
01203:      WRITE(UNITMP,9315)
01204:      WRITE (UNITMP,9317)
01205:      DO 2888 I=1,13
01206:      IDR=IORDER(I)
01207:      JDR = JORDER(I)
01208:      WRITE(UNITMP,9320)IDR,SMETAL(IDR,J),JDR,SMETAL(JDR,J)
01209:      TTTT = SU(J)/(ITR-IFIRST + 1)
01210:2888  CONTINUE
01211:COMMENT(Q)      WRITE (UNITMP,*) (' ')
01212:      DO 1010 JS = 1,NSUITE(1)
01213:COMMENT(Q)      WRITE (UNITMP,1770) RNUMD(1,JS,METTEM(J)),
01214:COMMENT(Q)      C RNUMD(2,JS,METTEM(J)),RNUMD(3,JS,METTEM(J))
01215:1010  CONTINUE
01216:C(Q)1770  FORMAT (' 90%= ',F16.5,'      50%= ',F16.5,'      10%= ',F16.
01217:COMMENT1771  FORMAT (' 95%= ',F16.5,'  90%= ',F16.5,'  50%= ',F16.5,
01218:COMMENT      C ' 10%= ',F16.5,'  05%= ',F16.5)
01219:C      WRITE (UNITMP,9325)
01220:      WRITE(UNITMP,9330)CMETAL(J),SU(J)/ITR,TTTT
01221:      WRITE (UNITMP,9331)
01222:COMMENT      I2222 = ND(4) + 1
01223:      IF (INOD.EQ.3) I2222 = ND(4) + 1
01224:      IF (INOD.EQ.5) I2222 = ND(6) + 1
01225:      DO 22222 I = 1,I2222
01226:      IM = I-1
01227:      IF (ND(INOD+1).EQ.0) GO TO 22222
01228:      WRITE (UNITMP,9999) IM,KONT(I),XX(I)
01229:COMMENT      TOT = TOT + FLOAT(KONT(I) * FLOAT(I-1))
01230:22222 CONTINUE
01231:      TOT = 0.
01232:      DO 2324 I = 1,234
01233:      TOT = TOT + FLOAT(KONT(I) * FLOAT(I-1))
01234:2324  CONTINUE
01235:      XMEANDP = TOT / FLOAT(ITR)
01236:      IF (ERR(METTEM(J)).EQ.0.0) WRITE (UNITMP,2229) XMEANDP
01237:      IF (ERR(METTEM(J)).NE.0.0) WRITE (UNITMP,2889)
01238:      C XMEANDP,ERR(METTEM(J))
01239:2889  FORMAT (/,'11X','MEAN NUMBER OF DEPOSITS ',F9.4,'3X,
01240:      C 'ERROR: MEAN TONNES METAL/DEPOSIT ',E12.4)
01241:      TOT = 0.
01242:2890  CONTINUE
01243:      IF (TOTF.EQ.'G'.OR.TOTF.NE.'T') GO TO 50
01244:      IF (STAR(IMODN-1).EQ.'*') GO TO 770
01245:      GO TO 50
01246:C ^214
01247:C
01248:C      ***  MODEL TOTALS      ***
01249:770  IF (TOTF.EQ.'G'.OR.TOTF.NE.'T') GO TO 7878
01250:      IF (IMODCT.GT.ITOTMOD) GO TO 9341
01251:      GO TO 4832
01252:7878  IF (ITOTSV.EQ.1) GO TO 9341
01253:4832  WRITE (1,4834)
01254:4834  FORMAT (/,' PROCESSING TOTALS...',/)
01255:      DO 765 J = 1,ITR
01256:      IF (ISCR12.EQ.1) READ (UNITSC1,4833) (SMETAL(J,I),I=1,35)
01257:      IF (ISCR12.EQ.2) READ (UNITSC2,4833) (SMETAL(J,I),I=1,35)
01258:4833  FORMAT (5X,35(E11.5))
01259:765  CONTINUE

```

```

01260:      DO 780 I=1,35
01261:      SU(I)=0.
01262:      DO 780 J=1,ITR
01263:780    SU(I)=SU(I) + SMETAL(J,I)
01264:      CALL SRT(ITR,IMETAL)
01265:      TOT = 0.
01266:      DO 890 J=1,35
01267:      IF(SU(J).EQ.0.)GO TO 890
01268:      IFIRST = ITR
01269:      DO 790 I = 1,ITR
01270:      IF (SMETAL(I,J).LE.0.) GO TO 790
01271:      IFIRST = I
01272:      GO TO 794
01273:790    CONTINUE
01274:794    JORDER(1) = IFIRST
01275:      DO 792 I = 2,10
01276:      JORDER(I) = ((ITR-IFIRST + 2) * (I-1))/10 + IFIRST-1
01277:792    CONTINUE
01278:      JORDER(11)=((ITR-IFIRST+2)*95)/100+IFIRST-1
01279:      JORDER(12)=((ITR-IFIRST+2)*99)/100+IFIRST-1
01280:      IF (ITFLAG.EQ.1) GO TO 796
01281:      WRITE (UNITMP,798)
01282:798    FORMAT(' ^214 ')
01283:      IF (TOTF.EQ.'T') WRITE (UNITMP,543) TRACT(IMODN-1)
01284:543    FORMAT (22X,'TRACT ',A5,/)
01285:      WRITE (UNITMP,793)
01286:793    FORMAT (12X,'***      TOTALS FOR METALS AND ORE      ***')
01287:      ITFLAG = 1
01288:796    JORDER(13) = ITR
01289:      WRITE(UNITMP,9310)CMETAL(J)
01290:      WRITE(UNITMP,9315)
01291:      WRITE(UNITMP,9317)
01292:      DO 888 I=1,13
01293:      IDR=IORDER(I)
01294:      JDR = JORDER(I)
01295:      WRITE(UNITMP,9320)IDR,SMETAL(IDR,J),JDR,SMETAL(JDR,J)
01296:      TTTT = SU(J)/(ITR-IFIRST + 1)
01297:888    CONTINUE
01298:C      WRITE (UNITMP,*) (' ')
01299:C      DO 10100 JS = 1,NSUITE(1)
01300:C      WRITE (UNITMP,1771) RNUMD(1,JS,METTEM(J)),RNUMD(2,JS,METTEM(J)
01301:C      C RNUMD(3,JS,METTEM(J)),KKONT(JS,METTEM(J))
01302:COMMENT1771  FORMAT (' 90%= ',F16.5,'    50%= ',F16.5,'    10%= ',F16.5
01303:C      C ' * KKONT= ',I6)
01304:C10100 CONTINUE
01305:C      WRITE (UNITMP,9325)
01306:      IF (J.EQ.19) WRITE(UNITMP,9329)CMETAL(J),SU(J)/ITR,TTTT
01307:      IF (J.NE.19) WRITE(UNITMP,9330)CMETAL(J),SU(J)/ITR,TTTT
01308:      WRITE (UNITMP,9331)
01309:9331    FORMAT (11X,'      ')
01310:C
01311:COMMENT      I2222 = ND(4) + 1
01312:      IF (INOD.EQ.3) I2222 = ND(4) + 1
01313:      IF (INOD.EQ.5) I2222 = ND(6) + 1
01314:      DO 2323 I = 1, I2222
01315:      IM = I-1
01316:      IF (ND(INOD+1).EQ.0) GO TO 2323
01317:C      WRITE (UNITMP,9999) IM,KONT(I),XX(I)
01318:C      TOT = TOT + FLOAT(NWKONT(I) * FLOAT (I-1))
01319:9999    FORMAT (11X,'N=',I3,3X,'FREQ(N)=',I8,3X,'PROB(N)=',F7.4)

```

```

01320:2323  CONTINUE
01321:      XMEANDP = TOT / FLOAT(ITR)
01322:C      WRITE (UNITMP,2229) XMEANDP
01323:2229  FORMAT (/ ,18X, 'MEAN NUMBER OF DEPOSITS      ',F9.4)
01324:      TOT = 0.
01325:890   CONTINUE
01326:      IF (TOTF.EQ.'G') GO TO 9341
01327:C
01328:C      ***** NEW TRACT ROUTINE ADDED
01329:      CLOSE (11)
01330:      CLOSE (12)
01331:      OPEN (UNIT=UNITSC1,STATUS='SCRATCH')
01332:      OPEN (UNIT=UNITSC2,STATUS='SCRATCH')
01333:      ISCR12 = 1
01334:      DO 188 J = 1,ITR
01335:      WRITE (UNITSC1,45) (ZERO(I),I=1,35)
01336:188      CONTINUE
01337:      REWIND (11)
01338:      JF = 1
01339:      ITFLAG = 0
01340:      DO 9309 ICPM = 1,10
01341:      PMETAL(ICPM) = ' '
01342:9309      CONTINUE
01343:      WRITE (UNITMP,*) (' ')
01344:      WRITE (UNITMP,*) (' ' *****
01345:      WRITE (UNITMP,*) ('^214')
01346:      GO TO 50
01347:C      ***** TRACT ROUTINE ENDS
01348:C
01349:9310  FORMAT(/ ,11X, '    SORTED SIMULATION RESULTS FOR ',A24)
01350:9315  FORMAT(/22X, 'ORDER OF OCCURRENCE'/)
01351:9317  FORMAT(17X, 'UNCONDITIONAL',12X, 'CONDITIONAL')
01352:9320  FORMAT(11X, I4,2X,1PE12.5,6X, I4,2X,1PE12.5)
01353:9325  FORMAT (/ ,47X, 'UNCONDITIONAL'3X, 'CONDITIONAL')
01354:9329  FORMAT(1X,18X, 'EXPECTED MEAN ',A5,
01355:      C / ,17X,2(1PE12.5,12X))
01356:C9330  FORMAT(1X, 'EXPECTED MEAN ',A16, ' METAL TONNAGE = ',2(1PE12.5,3X)
01357:9330  FORMAT(1X,20X, 'EXPECTED MEAN ',A16, / ,17X,2(1PE12.5,12X))
01358:C
01359:9341  CLOSE (11)
01360:      CLOSE (12)
01361:      OPEN (UNIT=9,FILE='TTTTF1',STATUS='SCRATCH')
01362:      CLOSE (9)
01363:      CLOSE (17)
01364:      WRITE (1,9343) DPFILE
01365:9343  FORMAT (/ , '    FILE CREATED IS - ',A32)
01366:      WRITE (1,9345)
01367:      WRITE (1,9345)
01368:9345  FORMAT ('^207')
01369:9346  END
01370:C
01371:^214
01372:      SUBROUTINE DISTRIB3(NX)
01373:      COMMON /BLK2/ XX(234), ND(6), D(6),INOD
01374:      COMMON /DRDIST/ PROBZ,IZDFLAG,DISTP3(3),DISTP5(5)
01375:C
01376:C      DIMENSION OF XX IS MAX NUMBER DEPOSITS
01377:C      DIMENSION XX MUST BE GE ND(4)
01378:C
01379:      IF (IZDFLAG.EQ.1) GO TO 20

```

```

01380:      IF (ELOPT.EQ.'E') GO TO 20
01381:      IZDFLAG = 1
01382:20    ND(1) = 0
01383:      DO 1 I = 1,INOD + 1
01384:      ND(I) = ND(I) + 1
01385:      D(I) = ND(I)
01386:1      CONTINUE
01387:      DO 2 L = 1, ND(INOD + 1)
01388:      XX(L) = 0.
01389:2      CONTINUE
01390:      IDUM = ND(2) - ND(1)
01391:      IF (IDUM.EQ.0) XX(ND(1)) = 1.-DISTP3(1)
01392:      IF (IDUM.EQ.0) GO TO 12
01393:      IF (IDUM.GT.0) DUM = (1.-DISTP3(1))/(2*(D(2)-D(1))+1)
01394:      DO 3 L = ND(1), ND(2)
01395:      IF (L.EQ.ND(2)) XX(L) = XX(L)+DUM
01396:      IF (L.LT.ND(2)) XX(L) = XX(L) + 2.*DUM
01397:3      CONTINUE
01398:12    DO 4 I = 2,INOD
01399:      IDUM = ND(I+1)-ND(I)
01400:      IF (IDUM.EQ.0) DUM = DISTP3(I-1)-DISTP3(I)
01401:      IF (IDUM.EQ.1) DUM = (DISTP3(I-1)-DISTP3(I))/2.
01402:      IF (IDUM.GT.1) DUM = (DISTP3(I-1)-DISTP3(I))/(2.*(IDUM-1.)+2.)
01403:      DO 5 L = ND(I), ND(I + 1)
01404:      IF (L.EQ.ND(I).OR.L.EQ.ND(I+1)) XX(L) = XX(L) + DUM
01405:      IF (L.GT.ND(I).AND.L.LT.ND(I+1)) XX(L) = XX(L)+2.*DUM
01406:5      CONTINUE
01407:4      CONTINUE
01408:      XX(ND(4)) = XX(ND(4)) + DISTP3(3)
01409:      IF (PROBZ.LT.-0.01.OR.PROBZ.GT.1.) GO TO 9
01410:      DUM = 1.-XX(1)
01411:      XX(1) = PROBZ
01412:      IF (DUM.GT.0) GO TO 11
01413:      WRITE (1,*) ('DIST. OF DEPOSIT NUMBER INCONSISTENTLY SPECIFIED.')
01414:      STOP
01415:11    IF (DUM.GT.0) A = ((1.-PROBZ)/DUM)
01416:      DO 10 I = 2, ND(INOD + 1)
01417:      XX(I) = XX(I) * A
01418:10    CONTINUE
01419:9      R = RAND()
01420:      TEST = 0.
01421:      DO 6 L = 1,ND(INOD + 1)
01422:      TEST = TEST + XX(L)
01423:      IF(TEST.GT.R) GO TO 7
01424:6      CONTINUE
01425:7      NX = L-1
01426:      DO 8 I = 1,INOD + 1
01427:      ND(I) = ND(I) - 1
01428:8      CONTINUE
01429:      RETURN
01430:      END
01431:C
01432:      SUBROUTINE DISTRIB5(NX)
01433:      COMMON /BLK2/ XX(234), ND(6), D(6),INOD
01434:      COMMON /DRDIST/ PROBZ,IZDFLAG,DISTP3(3),DISTP5(5)
01435:C
01436:C      DIMENSION OF XX IS MAX NUMBER DEPOSITS
01437:C      DIMENSION XX MUST BE GE ND(6)
01438:C
01439:      IF (IZDFLAG.EQ.1) GO TO 20

```

```

01440:      IF (ELOPT.EQ.'E') GO TO 20
01441:      IZDFLAG = 1
01442:20     ND(1) = 0
01443:      DO 1 I = 1,INOD+1
01444:      ND(I) = ND(I) + 1
01445:      D(I) = ND(I)
01446:1      CONTINUE
01447:      DO 2 L = 1, ND(INOD+1)
01448:      XX(L) = 0.
01449:2      CONTINUE
01450:      IDUM = ND(2) - ND(1)
01451:      IF (IDUM.EQ.0) XX(ND(1)) = 1.-DISTP5(1)
01452:      IF (IDUM.EQ.0) GO TO 12
01453:      IF (IDUM.GT.0) DUM = (1.-DISTP5(1))/(2*(D(2)-D(1))+1)
01454:      DO 3 L = ND(1), ND(2)
01455:      IF (L.EQ.ND(2)) XX(L) = XX(L)+DUM
01456:      IF (L.LT.ND(2)) XX(L) = XX(L) + 2.*DUM
01457:3      CONTINUE
01458:12     DO 4 I = 2,INOD
01459:      IDUM = ND(I+1)-ND(I)
01460:      IF (IDUM.EQ.0) DUM = DISTP5(I-1)-DISTP5(I)
01461:      IF (IDUM.EQ.1) DUM = (DISTP5(I-1)-DISTP5(I))/2.
01462:      IF (IDUM.GT.1) DUM = (DISTP5(I-1)-DISTP5(I))/(2.*(IDUM-1.)+2.)
01463:      DO 5 L = ND(I), ND(I + 1)
01464:      IF (L.EQ.ND(I).OR.L.EQ.ND(I+1)) XX(L) = XX(L) + DUM
01465:      IF (L.GT.ND(I).AND.L.LT.ND(I+1)) XX(L) = XX(L)+2.*DUM
01466:5      CONTINUE
01467:4      CONTINUE
01468:      XX(ND(6)) = XX(ND(6)) + DISTP5(5)
01469:      IF (PROBZ.LT.-0.01.OR.PROBZ.GT.1.) GO TO 9
01470:      DUM = 1.-XX(1)
01471:      XX(1) = PROBZ
01472:      IF (DUM.GT.0) GO TO 11
01473:      WRITE (1,*) ('DIST. OF DEPOSIT NUMBER INCONSISTENTLY SPECIFIED.')
01474:      STOP
01475:11     IF (DUM.GT.0) A = ((1.-PROBZ)/DUM)
01476:      DO 10 I = 2, ND(INOD+1)
01477:      XX(I) = XX(I) * A
01478:10     CONTINUE
01479:9      R = RAND()
01480:      TEST = 0.
01481:      DO 6 L = 1,ND(INOD+1)
01482:      TEST = TEST + XX(L)
01483:      IF(TEST.GT.R) GO TO 7
01484:6      CONTINUE
01485:7      NX = L-1
01486:      DO 8 I = 1,INOD+1
01487:      ND(I) = ND(I) - 1
01488:8      CONTINUE
01489:      RETURN
01490:      END
01491:C
01492:      SUBROUTINE EMPIRICAL (II)
01493:      COMMON /BLKE1/ ICO(234,10),INFOC(234),NSUITE(234),SPROB(234,12)
01494:      COMMON /BLKE2/ FEMP(234,12,11,10),DEMP(500,12),PVAL(11)
01495:      COMMON /BLKE3/ AEMP(234,12,10,10),ISUITE(234,12)
01496:      COMMON /BLK4/ JSUITE(234,12),P(234,12,10,10)
01497:      COMMON /BLK5/ R(10,10),ADUM(10,10),ICOMM(234),ERR(12)
01498:      COMMON /BLK6/ ICP1,IC,IDIST,NOR
01499:      COMMON /BLK7/ AA(11),BB(11),ZEMP(101),XNORE(2,1100),XEMP(101)

```

```

01500:      COMMON /METFLAG/ METNUM,NUMCOM,UNITMP
01501:      COMMON /BLK8/ METYES(10)
01502:      COMMON /OWN/L1(521),P2(32),LP,LQ,NQ1,MM,MEMP
01503:      COMMON /DFILE/ DINFILE,METN(10),IECOD,IEMPCOD(99,10)
01504:      COMMON /TABLE/ TTABLE(1100,2)
01505:      INTEGER * 2 IEMPET,IECOD,IELEN
01506:      DIMENSION XMEAN(10,12,10),PS(10,12,10,10)
01507:      DIMENSION XC(10),XZ(10),TOT(10,12,10)
01508:      CHARACTER*60 FILE(10),DINFILE
01509:      CHARACTER * 80 EMPFILE
01510:      DATA PVAL/0.,.1,.2,.3,.4,.5,.6,.7,.8,.9,1./
01511:C      EXPLANATION OF DIMENSIONS: MEMP=# MODELS,MAXS=MAXIMUM # OF
01512:C      SUITES, IC=# OF COMMODITIES INCLUDING TONNAGE,MAXD=MAX
01513:C      NUMBER OF DEPOSITS IN ONE DEPOSIT TYPE FILE
01514:C      IDIST=# BREAK POINTS IN CUMULATIVE DISTRIBUTION APPROX.
01515:C      ICO(MEMP,IC),NSUITE(MAXS),SPROB(MEMP,MAXS),FEMP(MEMP,MAXS,IDIST,
01516:C      DEMP(MAXD,IC+1),AEMP(MEMP,MAXS,IC,IC),ISUITE(MEMP,MAXS)
01517:C      JSUITE(MEMP,MAXS),NSUITE(MEMP),R(IC,IC),ADUM(IC,IC),ICOMM(IC)
01518:C      METYES(IC) IS A INDICATOR 0 IF METAL ABSENT, 1 IF PRESENT
01519:C      FILE(MEMP),INFOC(MEMP),P(MEMP,MAXS,IC,IC),AA(IDIST),BB(IDIST)
01520:C      ICO() GIVES THE CODES OF THE METALS PRESENT IN EACH MODEL
01521:C      INFOC() GIVES THE NUMBER OF METAL PRESENT IN EACH MODEL
01522:C      NSUITE() GIVES # OF SUITES IN A MODEL
01523:C      SPROB() IS THE PROBABILITY OF A SUITE GIVEN A MODEL
01524:C      FEMP() GIVES FOR A MODEL, SUITE, AND METAL IDIST POINTS ON THE
01525:C      CUMULATIVE DISTRIBUTION FUNCTION
01526:C      DEMP() HAS ORES AND GRADES FOR EACH DEPOSIT WITHIN A MODEL
01527:C      PVAL() GIVES THE PROBABILITIES FOR THE POINTS IN FEMP()
01528:C      AEMP() GIVES THE COEFFICIENTS FOR LINEAR COMBINATIONS OF
01529:C      INQENDENT NORMALS THAT WILL HAVE THE CORRELATIONS GIVEN
01530:C      R() HAS PAIRWISE CORRELATIONS BETWEEN METALS WITHIN A SUITE
01531:C      ISUITE() AND JSUITE ARE FIRST AND LAST ROW NUMBERS OF SUITE
01532:C      ADUM() WAS NOT USED. ICOMM() WAS NOT USED. NOR= # POINTS
01533:C      IN NORMAL.TBL. AA() AND BB() ARE DUMMIES THAT CARRY DISTRI-
01534:C      BUTIONS. ZEMP() IS THE POINTS USED IN THE PIECEWISE LINEAR
01535:C      APPROXIMATION OF A NORMAL. XEMP() HAS CORRESPONDING PROBA-
01536:C      BILITIES. XNORE() HAS VALUES AND PROBABILITIES FROM NORMAL.TBL.
01537:C
01538:COMMENT      I=1
01539:COMMENT      OPEN(UNIT=8,FILE='NORMAL.TBL')
01540:COMMENT      31 READ (8,29,END=30) (XNORE(J,I),J=1,2)
01541:COMMENT      29 FORMAT(2(1X,F8.5))
01542:COMMENT      I=I+1
01543:COMMENT      GO TO 31
01544:COMMENT      30 CLOSE(8)
01545:COMMENT      NOR=I-1
01546:COMMENT      I PREVIOUSLY WAS 101 FOR NORMAL.TBL
01547:COMMENT      I-1 PREVIOUSLY MADE NOR 100, NOW PRESET TO 100
01548:      NOR = 100
01549:      NBT=501
01550:      NI=1
01551:      RANGE=1.5
01552:      MEMP=10
01553:      ITR=4999
01554:      IC=10
01555:      ICP1=IC+1
01556:      MAXD=500
01557:      MAXS=12
01558:      IDIST=11
01559:      DO 4 I=1,MEMP

```

```

01560:      NSUITE(I)=0
01561:      DO 5 J=1,MAXS
01562:      SPROB(I,J)=0.
01563:      ISUITE(I,J)=0
01564:      JSUITE(I,J)=0
01565:      DO 6 K=1,IC
01566:      ICO(I,K)=0
01567:      ICOMM(IC)=0
01568:      DO 7 L=1,IDIST
01569:      FEMP(I,J,L,K)=0.
01570:      DO 8 I8=1,IC
01571:      R(K,I8)=0.
01572:      ADUM(K,I8)=0.
01573:      AEMP(I,J,K,I8)=0.
01574:      8 CONTINUE
01575:      7 CONTINUE
01576:      6 CONTINUE
01577:      5 CONTINUE
01578:      4 CONTINUE
01579:      DO 9 L=1,MAXD
01580:      DO 10 I=1,ICP1
01581:      DEMP(L,I)=0.
01582:      10 CONTINUE
01583:      9 CONTINUE
01584:      LENCAT = INDEX (DINFILE,' ')
01585:      EMPFILE = DINFILE
01586:      EMPFILE (LENCAT:LENCAT+3) = '.EMP'
01587:      OPEN (UNIT=20,FILE=EMPFILE)
01588:      READ (20,67,END=65,ERR=65) NDUM,NDEE
01589:67      FORMAT (2I3)
01590:      NSUITE(1) = NDUM
01591:      NUMCOM = NDEE
01592:      DO 66 I = 1,NSUITE(1)
01593:      DO 68 J = 1,IDIST
01594:      READ (20,69) (FEMP(1,I,J,K),K=1,NUMCOM)
01595:69      FORMAT (10E12.5)
01596:68      CONTINUE
01597:      DO 70 J = 1,NUMCOM
01598:      READ (20,71) (AEMP(1,I,J,K),K=1,NUMCOM)
01599:71      FORMAT (10F8.5)
01600:70      CONTINUE
01601:      READ (20,72) SPROB(1,I)
01602:72      FORMAT (F8.5)
01603:66      CONTINUE
01604:      CLOSE (20)
01605:      RETURN
01606:C
01607:C      STORE VALUES FROM A NORMAL TABLE
01608:C
01609:65      I = 1
01610:      MOD = 1
01611:      OPEN (UNIT = 19,STATUS='SCRATCH')
01612:      J = 1
01613:      DO 555 J=1,10
01614:      IEMET = IEMPCOD(II,J)
01615:      WRITE (19,222) IEMET
01616:222      FORMAT (I2)
01617:555      CONTINUE
01618:      REWIND (19)
01619:      I = 1

```

```

01620:333 READ (19,222,END=444) ICO(MEMP,I)
01621: I = I + 1
01622: GO TO 333
01623:444 INFOC(MEMP) = I - 1
01624:C
01625:C UNIT=8 IS THE FILE THAT FILLS ICO() AND INFOC()
01626:C THERE IS A FILE THAT TELLS HOW MANY AND WHAT KIND METALS IN
01627:C EACH OF THE MODELS. ASSUME THAT INFOC IS IN AN ARRAY
01628:C INFOC(MEMP) HAS NUMBER OF METALS, ICO(MEMP,C) HAS METAL CODES
01629:C FOR EACH OF THE MODELS
01630:C
01631:C THERE WILL HAVE TO BE SOME CODE MATCHING THE INDICIES IN
01632:C FILE(I) AND INFOC(I) SO THAT THEY ARE THE SAME MODELS.
01633:C
01634:C INFOC(1)=3 THIS IS NOW SET AT METAL SETTINGS.
01635:C
01636: DO 11 I11=1,MOD
01637: OPEN(UNIT=7,FILE=DINFILE)
01638: DO 12 L=1,MAXD
01639: DO 13 I=1,ICP1
01640: DEMP(L,I)=0.
01641: 13 CONTINUE
01642: 12 CONTINUE
01643:C READ IN DATA FROM THE MODELS ONE AT A TIME
01644: I=1
01645: 14 READ (7,15,END=16) (DEMP(I,J),J=1,INFOC(I11))
01646: 15 FORMAT(25X,F12.0,F9.0,F9.0,6X,F11.0,4X,F12.0,4X,F9.0,
01647: C 7X,F12.0)
01648: IS=0
01649:C COMPUTE THE SUITE INDEX
01650: DO 17 I17=1,INFOC(I11)
01651: IF(DEMP(I,I17).GT.0.) IS=IS+2**(.17-1)
01652:C IF(I17.GT.1) DEMP(I,I17)=DEMP(I,I17)/100.
01653: 17 CONTINUE
01654: DEMP(I,ICP1)=IS
01655: I=I+1
01656: GO TO 14
01657: 16 CLOSE(7)
01658:C N11= THE NUMBER OF DEPOSITS IN MODEL I11
01659: N11=I-1
01660:C GROUP THE DATA SO DEPOSITS WITH COMMON SUITES OF METALS
01661:C ARE TOGETHER
01662: CALL SORTD(N11)
01663:C NOW WE COUNT THE SUITES AND SEE WHERE THEY START AND END
01664: NSUITE(I11)=1
01665: ISUITE(I11,1)=1
01666: JSUITE(I11,1)=1
01667: ISTOP=N11-1
01668: DO 18 I=1,ISTOP
01669: TEST=DEMP(I,ICP1)-DEMP(I+1,ICP1)
01670: IF(TEST.GT.0.5) GO TO 19
01671: JSUITE(I11,NSUITE(I11))=I+1
01672: GO TO 18
01673: 19 NSUITE(I11)=NSUITE(I11)+1
01674: ISUITE(I11,NSUITE(I11))=I+1
01675: JSUITE(I11,NSUITE(I11))=I+1
01676: 18 CONTINUE
01677: DO 20 I=1,NSUITE(I11)
01678: DUM=N11
01679: DEE=JSUITE(I11,I)-ISUITE(I11,I)+1.

```

```

01680:      SPROB(I11,I)=DEE/DUM
01681: 20 CONTINUE
01682:      DO 21 I=1,NSUITE(I11)
01683:      DO 22 J=1,IC
01684:      DO 23 K=1,IC
01685:      DUM=0
01686:      DO 24 L=ISUITE(I11,I),JSUITE(I11,I)
01687:      DUM=DUM+DEMP(L,K)*DEMP(L,J)
01688: 24 CONTINUE
01689:      DUM=DUM/(JSUITE(I11,I)-ISUITE(I11,I)+1.)
01690:      P(I11,I,J,K)=DUM
01691: 23 CONTINUE
01692: 22 CONTINUE
01693: 21 CONTINUE
01694:C      FIND IDIST POINTS ON A PIECEWISE LINEAR DISTRIBUTION THAT
01695:C      APPROXIMATES AN EMPIRICAL AND HAS SAME MEAN
01696:      CALL EMPDIST(I11)
01697:C      CALCULATE THE MEANS WITHIN A SUITE
01698:      DO 58 J=1,NSUITE(I11)
01699:      DO 57 I=1,IC
01700:      DUM=0.
01701:      DO 59 K=ISUITE(I11,J),JSUITE(I11,J)
01702:      DUM=DUM+DEMP(K,I)/(JSUITE(I11,J)-ISUITE(I11,J)+1)
01703: 59 CONTINUE
01704:      XMEAN(I11,J,I)=DUM
01705: 57 CONTINUE
01706: 58 CONTINUE
01707:      DO 42 I42=1,NSUITE(I11)
01708:      DO 39 I39=1,IDIST
01709: 41 FORMAT(4(1X,E10.4))
01710: 39 CONTINUE
01711: 42 CONTINUE
01712:      DO 25 I=1,NSUITE(I11)
01713:C      LIST WHAT METALS ARE PRESENT IN THIS SUITE
01714:      DO 38 K=1,IC
01715:      METYES(K)=0
01716:      TEST=FEMP(I11,I,11,K)*1000000.
01717:      IF(TEST.GT.0.5) METYES(K)=1
01718: 38 CONTINUE
01719:C      FIND ALL THE PAIRWISE CORRELATIONS
01720:      DO 26 K=1,IC
01721:      DO 27 J=1,IC
01722:      IF(K.EQ.J) R(J,K)=1.
01723:      IF(K.EQ.J) GO TO 27
01724:      IF(METYES(K).EQ.1.AND.METYES(J).EQ.1) GO TO 43
01725:      R(J,K)=0.
01726:      GO TO 27
01727: 43 DO 28 L=1,IDIST
01728:      AA(L)=FEMP(I11,I,L,K)
01729:      BB(L)=FEMP(I11,I,L,J)
01730: 28 CONTINUE
01731:      CALL EMPCOR(P(I11,I,K,J),R(K,J))
01732: 27 CONTINUE
01733: 26 CONTINUE
01734:C      FILL THE MATRIX OF COEFFICIENTS, AEMP(I11,I,...), FOR LINEAR
01735:C      COMBINATIONS OF NORMALS THAT GIVE THE RIGHT CORRELATIONS
01736:      DO 32 K=1,IC
01737:      IF(METYES(K).EQ.0) AEMP(I11,I,K,K)=1.
01738:      IF(METYES(K).EQ.1) AEMP(I11,I,K,1)=R(K,1)
01739:      IF(METYES(K).EQ.0) GO TO 32

```

```

01740:      DUM=0.
01741:      DO 33 J=2,K
01742:      DUM=DUM+AEMP(I11,I,K,J-1)**2
01743:      IF(DUM.LE.1.) GO TO 34
01744:      DUM = DUM-AEMP(I11,I,K,J-1) ** 2
01745:      DEE = SQRT (1.-DUM)
01746:      SGN = 1.
01747:      IF (AEMP(I11,I,K,J-1).LT.0.) SGN = -1.
01748:      AEMP (I11,I,K,J-1) = SGN * DEE
01749:      WRITE(1,35) I11,DUM
01750: 35  FORMAT('INCONSISTENT CORRELATIONS',1X,I3,3X,E11.4)
01751:      AEMP(I11,I,K,K) = 0.
01752:      GO TO 32
01753: 34  IF(J.EQ.K) AEMP(I11,I,K,J)=SQRT(1.-DUM)
01754:      IF(J.EQ.K) GO TO 33
01755:      IF(METYES(J).EQ.0) GO TO 33
01756:      JM1=J-1
01757:      DEE=0.
01758:      DO 36 L=1,JM1
01759:      DEE=DEE+AEMP(I11,I,K,L)*AEMP(I11,I,J,L)
01760: 36  CONTINUE
01761:      DEE=R(K,J)-DEE
01762:      IF(AEMP(I11,I,J,J).GT.0.) GO TO 37
01763:      AEMP(I11,I,K,J)=0.
01764:      GO TO 33
01765: 37  AEMP(I11,I,K,J)=DEE/AEMP(I11,I,J,J)
01766: 33  CONTINUE
01767: 32  CONTINUE
01768:      DO 44 K=1,IC
01769:      DO 45 J=1,IC
01770:      DUM=0.
01771:      DO 46 L=1,IC
01772:      DUM=DUM+AEMP(I11,I,J,L)*AEMP(I11,I,K,L)
01773: 46  CONTINUE
01774:      IF(METYES(J).EQ.0.OR.METYES(K).EQ.0) GO TO 44
01775: 45  CONTINUE
01776: 44  CONTINUE
01777: 25  CONTINUE
01778: 11  CONTINUE
01779:      WRITE (1,*) (' ')
01780:      REWIND (20)
01781:C      INFOC(MEMP) NOW EQUALS NUMCOM
01782:      INFOC(MEMP) = NUMCOM
01783:      WRITE (20,67) NSUITE(1),INFOC(MEMP)
01784:      DO 73 I = 1,NSUITE(1)
01785:      DO 74 J = 1,IDIST
01786:      WRITE (20,69) (FEMP(1,I,J,K), K = 1,NUMCOM)
01787:74  CONTINUE
01788:      DO 75 J = 1,NUMCOM
01789:      WRITE (20,71) (AEMP(1,I,J,K), K = 1,NUMCOM)
01790:75  CONTINUE
01791:      WRITE (20,72) SPROB(1,I)
01792:73  CONTINUE
01793:      ENDFILE (20)
01794:      CLOSE (20)
01795:      RETURN
01796:      END
01797:C
01798:      SUBROUTINE EMPCOR(PROD,RHO)
01799:C      THIS WILL FIND THE CORRELATION BETWEEN TWO DISTRIBUTIONS

```

```

01800:C      THAT WILL GIVE A SPECIFIED MEAN PRODUCT
01801:      COMMON /BLK7/ AA(11),BB(11),ZEMP(101),XNORE(2,1100),XEMP(101)
01802:      COMMON /BLK6/ ICP1,IC,IDIST,NOR
01803:      COMMON /BLKE2/ FEMP(234,12,11,10),DEMP(500,12),PVAL(11)
01804:      NGRID=100
01805:      LESS=NGRID-1
01806:      GRID=NGRID
01807:      XX=0.5*(1./GRID)
01808:      DO 1 I=1,NGRID
01809:      XEMP(I)=XX
01810:      CALL NORVALE(XX,ZEMP(I))
01811:      CRES=1./GRID
01812:      IF(I.EQ.LESS) CRES=0.5*(1./GRID)
01813:      XX=XX+CRES
01814: 1 CONTINUE
01815:      TH=3.14159/4.
01816:      TL=-3.14159/4.
01817:      LOOP=0
01818: 2 T=(TH+TL)/2.
01819:      A1=COS(T)
01820:      A2=SIN(T)
01821:      RHO=SIN(2.*T)
01822:      TEST=0.
01823:      LOOP=LOOP+1
01824:      IF(LOOP.GT.15) GO TO 3
01825:      DO 4 I=1,NGRID
01826:      DO 5 J=1,NGRID
01827:      Z1=A1*ZEMP(I)+A2*ZEMP(J)
01828:      Z2=A2*ZEMP(I)+A1*ZEMP(J)
01829:      CALL NORPROBE(XX,Z1)
01830:      CALL NORPROBE(YY,Z2)
01831:      CALL EMPVAL(XX,AA,V1)
01832:      CALL EMPVAL(YY,BB,V2)
01833:      TEST=TEST+V1*V2
01834: 5 CONTINUE
01835: 4 CONTINUE
01836:      TEST=TEST/(GRID*GRID)
01837:      IF(TEST.GE.PROD) TH=T
01838:      IF(TEST.LE.PROD) TL=T
01839:      RHOH=SIN(2.*TL)
01840:      RHOI=SIN(2.*TH)
01841:      GO TO 2
01842: 3 RETURN
01843:      END
01844:C
01845:      SUBROUTINE EMPVAL(PROB,CC,VAL)
01846:C      THIS WILL FIND THAT VALUE THAT CORRESPONDS TO A PROBABILITY
01847:C      IN A PIECEWISE LINEAR DISTRIBUTION. VAL IS RETURNED
01848:C      DIMENSION CC(IDIST)
01849:      DIMENSION CC(13)
01850:      COMMON /BLKE2/ FEMP(234,12,11,10),DEMP(500,12),PVAL(11)
01851:      COMMON /BLK6/ ICP1,IC,IDIST,NOR
01852:      IF (PROB.LE.0.0) VAL = CC(1)
01853:      IF (PROB.GE.1.0) VAL = CC(11)
01854:      IF (PROB.LE.0.0.OR.PROB.GE.1.0) GO TO 2
01855:      DO 1 I=2,IDIST
01856:      IF(PROB.GT.PVAL(I)) GO TO 1
01857:      AEMP=PVAL(I)-PROB
01858:      AEMP=AEMP/(PVAL(I)-PVAL(I-1))
01859:      VAL=AEMP*CC(I-1)+(1.-AEMP)*CC(I)

```

```

01860:      GO TO 2
01861:      1 CONTINUE
01862:      2 RETURN
01863:      END
01864:C
01865:      SUBROUTINE NORVALE(PROB,VAL)
01866:C      THIS WILL FIND THE VALUE OF A NORMAL RANDOM VARIABLE THAT
01867:C      CORRESPONDS TO A GIVEN PROBABILITY. VAL IS RETURNED
01868:      COMMON /BLK7/ AA(11),BB(11),ZEMP(101),XNORE(2,1100),XEMP(101)
01869:      COMMON /BLK6/ ICP1,IC,IDIST,NOR
01870:      VAL=XNORE(2,1)-(XNORE(2,2)-XNORE(2,1))
01871:      IF(PROB.LT.XNORE(1,1)) GO TO 2
01872:      DO 1 I=2,NOR
01873:      IF(PROB.GE.XNORE(1,I)) GO TO 1
01874:      AEMP=XNORE(1,I)-PROB
01875:      AEMP=AEMP/(XNORE(1,I)-XNORE(1,I-1))
01876:      VAL=AEMP*XNORE(2,I-1)+(1.-AEMP)*XNORE(2,I)
01877:      GO TO 2
01878:      1 CONTINUE
01879:      VAL=XNORE(2,NOR)+(XNORE(2,NOR)-XNORE(2,NOR-1))
01880:      2 RETURN
01881:      END
01882:C
01883:      SUBROUTINE NORPROBE(PROB,VAL)
01884:C      THIS WILL FIND THE PROBABILITY THAT CORRESPONDS TO A GIVEN
01885:C      VALUE OF A NORMAL RANDOM VARIABLE. PROB IS RETURNED
01886:      COMMON /BLK7/ AA(11),BB(11),ZEMP(101),XNORE(2,1100),XEMP(101)
01887:      COMMON /BLK6/ ICP1,IC,IDIST,NOR
01888:      IF(VAL.LT.XNORE(2,NOR)) GO TO 1
01889:      PROB=.999
01890:      GO TO 2
01891:      1 IF(VAL.GT.XNORE(2,1)) GO TO 3
01892:      PROB=0.001
01893:      GO TO 2
01894:      3 DO 4 I=2,NOR
01895:      IF(VAL.GE.XNORE(2,I)) GO TO 4
01896:      AEMP=XNORE(2,I)-VAL
01897:      AEMP=AEMP/(XNORE(2,I)-XNORE(2,I-1))
01898:      PROB=AEMP*XNORE(1,I-1)+(1.-AEMP)*XNORE(1,I)
01899:      GO TO 2
01900:      4 CONTINUE
01901:      2 RETURN
01902:      END
01903:C
01904:      SUBROUTINE SORTD(N11)
01905:C      THIS SORTS THE ROWS OF THE DATA MATRIX DEMP() ON THE LAST
01906:C      COLUMN, THE SUITE INDEX
01907:      COMMON /BLK2/ FEMP(234,12,11,10),DEMP(500,12),PVAL(11)
01908:      COMMON /BLK6/ ICP1,IC,IDIST,NOR
01909:      ISTART=N11
01910:      3 ISWITCH=0
01911:      DO 1 I=2,ISTART
01912:      IF(DEMP(I-1,ICP1).GE.DEMP(I,ICP1)) GO TO 1
01913:      DO 2 J=1,ICP1
01914:      DUM=DEMP(I,J)
01915:      DEMP(I,J)=DEMP(I-1,J)
01916:      DEMP(I-1,J)=DUM
01917:      2 CONTINUE
01918:      ISWITCH=ISWITCH+1
01919:      1 CONTINUE

```

```

01920:      ISTART=ISTART-1
01921:      IF(ISWITCH.NE.0.AND.ISTART.GE.2) GO TO 3
01922:      RETURN
01923:      END
01924:C
01925:      SUBROUTINE EMPDIST(MODEL)
01926:C      THIS ROUTINE CALCULATES DISTRIBUTIONS OF GRADES AND
01927:C      TONNAGES THAT APPROXIMATE THE EMPIRICAL AND HAVE THE
01928:C      SAME MEANS.  THE DISTRIBUTIONS ARE PIECEWISE LINEAR
01929:C      AND SPECIFIED AT IDIST POINTS.
01930:      COMMON /BLKE2/ FEMP(234,12,11,10),DEMP(500,12),PVAL(11)
01931:      COMMON /BLKE1/ ICO(234,10),INFOC(234),NSUITE(234),SPROB(234,12)
01932:      COMMON /BLKE3/ AEMP(234,12,10,10),ISUITE(234,12)
01933:      COMMON /BLK4/ JSUITE(234,12),P(234,12,10,10)
01934:      COMMON /BLK6/ ICP1,IC,IDIST,NOR
01935:      RANGE=1.5
01936:      DO 1 I=1,NSUITE(MODEL)
01937:      DO 2 J=1,IC
01938:      TRUE=0.
01939:      TOP=0.
01940:      BOT=10.**10.
01941:      SIZE=JSUITE(MODEL,I)-ISUITE(MODEL,I)+1.
01942:      DO 3 K=ISUITE(MODEL,I),JSUITE(MODEL,I)
01943:      TRUE=TRUE+DEMP(K,J)
01944:      IF(DEMP(K,J).GT.TOP) TOP=DEMP(K,J)
01945:      IF(DEMP(K,J).LT.BOT) BOT=DEMP(K,J)
01946:      3 CONTINUE
01947:      TOP=TOP*RANGE
01948:      BOT=BOT*(1.-(RANGE-1.))
01949:      TRUE=TRUE/SIZE
01950:      IF(TRUE.GT.0.0) GO TO 4
01951:      DO 5 K=1,IDIST
01952:      FEMP(MODEL,I,K,J)=0.
01953:      5 CONTINUE
01954:      GO TO 2
01955:      4 KSTOP=IDIST-1
01956:      DO 6 K=1,KSTOP
01957:      LOOP=0
01958:      XH=TOP
01959:      XL=BOT
01960:      9 XT=(XH+XL)/2.
01961:      VAL=0.
01962:      LOOP=LOOP+1
01963:      IF(LOOP.GT.35) GO TO 10
01964:      DO 7 I7=ISUITE(MODEL,I),JSUITE(MODEL,I)
01965:      TEST=RANGE*DEMP(I7,J)
01966:      BEST=(1.-(RANGE-1.))*DEMP(I7,J)
01967:      IF(XT.LE.BEST) GO TO 7
01968:      IF(XT.GT.TEST) GO TO 8
01969:      VAL=VAL+(XT-BEST)/(TEST-BEST)*(1./SIZE)
01970:      GO TO 7
01971:      8 VAL=VAL+1./SIZE
01972:      7 CONTINUE
01973:      IF(VAL.GE.PVAL(K)) XH=XT
01974:      IF(VAL.LE.PVAL(K)) XL=XT
01975:      GO TO 9
01976:      10 FEMP(MODEL,I,K,J)=XT
01977:      6 CONTINUE
01978:      KSTOP=IDIST-1
01979:      AVG=0.

```

```

01980:      XINT=IDIST-1
01981:      DO 11 K=1,KSTOP
01982:      FAC=1./XINT
01983:      IF(K.EQ.1) FAC=FAC/2.
01984:      AVG=AVG+FAC*FEMP(MODEL,I,K,J)
01985: 11 CONTINUE
01986:      FEMP(MODEL,I,IDIST,J)=(XINT*2.)*(TRUE-AVG)
01987:      IF(FEMP(MODEL,I,IDIST,J).GT.FEMP(MODEL,I,IDIST-1,J)) GO TO 2
01988:      DIFF = FEMP (MODEL,I,IDIST-1,J) - FEMP(MODEL,I,IDIST,J)
01989:      FEMP (MODEL,I,IDIST,J) = FEMP (MODEL,I,IDIST-1,J)
01990:      FEMP (MODEL,I,IDIST-1,J) = FEMP (MODEL,I,IDIST-1,J)-DIFF/2.
01991:      IF (FEMP(MODEL,I,IDIST-1,J).GT.FEMP(MODEL,I,IDIST-2,J)) GO TO 2
01992: 12 WRITE(1,13) MODEL,I,J,(FEMP(MODEL,I,K,J),K=IDIST-2,IDIST-1)
01993: 13 FORMAT('NO MEAN WORK',1X,I4,1X,I4,1X,I4,2(1X,E10.4))
01994:      2 CONTINUE
01995:      1 CONTINUE
01996:      RETURN
01997:      END
01998:C
01999:      SUBROUTINE LOGNORMAL (II,ELOPT,ELNLABEL)
02000:      COMMON /BLKE1/ ICO(234,10),INFOC(234),NSUITE(234),SPROB(234,12)
02001:      COMMON /BLKE2/ FEMP(234,12,11,10),DEMP(500,12),PVAL(11)
02002:      COMMON /BLKE3/ AEMP(234,12,10,10),JSUITE(234,12)
02003:      COMMON /BLK4/ JSUITE(234,12),P(234,12,10,10)
02004:      COMMON /BLK5/ R(10,10),ADUM(10,10),ICOMM(234),ERR(12)
02005:      COMMON /BLK6/ ICP1,IC,IDIST,NOR
02006:      COMMON /BLK7/ AA(11),BB(11),ZEMP(101),XNORE(2,1100),XEMP(101)
02007:      COMMON /TABLE/ TTABLE(1100,2)
02008:      COMMON /METFLAG/ METNUM,NUMCOM,UNITMP
02009:      COMMON /BLK8/ METYES(10)
02010:      COMMON /BLK9/ ZD(12,10), ZM(12,10)
02011:      COMMON /OWN/L1(521),P2(32),LP,LQ,NQ1,MM,MEMP
02012:      COMMON /DFILE/ DINFILE,METN(10),IECOD,IEMPCOD(99,10)
02013:      INTEGER UNITMP
02014:      INTEGER * 2 IEMPET,IECOD,IELEN
02015:      DIMENSION XMEAN(10,12,10),PS(10,12,10,10)
02016:      DIMENSION XC(10),XZ(10),TOT(10,12,10)
02017:      CHARACTER*60 FILE(10),DINFILE
02018:      CHARACTER * 30 LOGFILE
02019:      CHARACTER * 10 ELNLABEL
02020:      CHARACTER * 1 ELOPT
02021:      DATA PVAL/0.,.1,.2,.3,.4,.5,.6,.7,.8,.9,1./
02022:C      EXPLANATION OF DIMENSIONS: MEMP=# MODELS,MAXS=MAXIMUM # OF
02023:C      SUITES, IC=# OF COMMODITIES INCLUDING TONNAGE,MAXD=MAX
02024:C      NUMBER OF DEPOSITS IN ONE DEPOSIT TYPE FILE
02025:C      IDIST=# BREAK POINTS IN CUMULATIVE DISTRIBUTION APPROX.
02026:C      ICO(MEMP,IC),NSUITE(MAXS),SPROB(MEMP,MAXS),FEMP(MEMP,MAXS,IDIST,
02027:C      DEMP(MAXD,IC+1),AEMP(MEMP,MAXS,IC,IC),JSUITE(MEMP,MAXS)
02028:C      JSUITE(MEMP,MAXS),NSUITE(MEMP),R(IC,IC),ADUM(IC,IC),ICOMM(IC)
02029:C      METYES(IC) IS A INDICATOR 0 IF METAL ABSENT, 1 IF PRESENT
02030:C      FILE(MEMP),INFOC(MEMP),P(MEMP,MAXS,IC,IC),AA(IDIST),BB(IDIST)
02031:C      ICO() GIVES THE CODES OF THE METALS PRESENT IN EACH MODEL
02032:C      INFOC() GIVES THE NUMBER OF METAL PRESENT IN EACH MODEL
02033:C      NSUITE() GIVES # OF SUITES IN A MODEL
02034:C      SPROB() IS THE PROBABILITY OF A SUITE GIVEN A MODEL
02035:C      FEMP() GIVES FOR A MODEL, SUITE, AND METAL IDIST POINTS ON THE
02036:C      CUMULATIVE DISTRIBUTION FUNCTION
02037:C      DEMP() HAS ORES AND GRADES FOR EACH DEPOSIT WITHIN A MODEL
02038:C      PVAL() GIVES THE PROBABILITIES FOR THE POINTS IN FEMP()
02039:C      AEMP() GIVES THE COEFFICIENTS FOR LINEAR COMBINATIONS OF

```

```

02040:C      INQENDENT NORMALS THAT WILL HAVE THE CORRELATIONS GIVEN
02041:C      R() HAS PAIRWISE CORRELATIONS BETWEEN METALS WITHIN A SUITE
02042:C      ISUITE() AND JSUITE ARE FIRST AND LAST ROW NUMBERS OF SUITE
02043:C      ADUM() WAS NOT USED.  ICOMM() WAS NOT USED.  NOR= # POINTS
02044:C      IN NORMAL.TBL.  AA() AND BB() ARE DUMMIES THAT CARRY DISTRI-
02045:C      BUTIONS.  ZEMP() IS THE POINTS USED IN THE PIECEWISE LINEAR
02046:C      APPROXIMATION OF A NORMAL.  XEMP() HAS CORRESPONDING PROBA-
02047:C      BILITIES.  XNORE() HAS VALUES AND PROBABILITIES FROM NORMAL.TBL
02048:C
02049:COMMENT      I=1
02050:COMMENT      OPEN(UNIT=8,FILE='NORMAL.TBL')
02051:COMMENT      31 READ (8,29,END=30) (XNORE(J,I),J=1,2)
02052:COMMENT      29 FORMAT(2(1X,F8.5))
02053:COMMENT      I=I+1
02054:COMMENT      GO TO 31
02055:COMMENT      30 CLOSE(8)
02056:COMMENT      NOR=I-1
02057:COMMENT      I PREVIOUSLY WAS 101 FOR NORMAL.TBL
02058:COMMENT      I-1 PREVIOUSLY MADE NOR 100, NOW PRESET TO 100
02059:      NOR = 100
02060:      NBT=501
02061:      NI=1
02062:      RANGE=1.5
02063:      MEMP=10
02064:      ITR=4999
02065:      IC=10
02066:      ICP1=IC+1
02067:      MAXD=500
02068:      MAXS=12
02069:      DO 4 I=1,MEMP
02070:      NSUITE(I)=0
02071:      DO 5 J=1,MAXS
02072:      SPROB(I,J)=0.
02073:      ISUITE(I,J)=0
02074:      JSUITE(I,J)=0
02075:      DO 6 K=1,IC
02076:      ICO(I,K)=0
02077:      ICOMM(IC)=0
02078:      DO 8 I8=1,IC
02079:      R(K,I8)=0.
02080:      ADUM(K,I8)=0.
02081:      AEMP(I,J,K,I8)=0.
02082:      8 CONTINUE
02083:      6 CONTINUE
02084:      5 CONTINUE
02085:      4 CONTINUE
02086:      DO 9 L=1,MAXD
02087:      DO 10 I=1,ICP1
02088:      DEMP(L,I)=0.
02089:      10 CONTINUE
02090:      9 CONTINUE
02091:      ELOPT = 'L'
02092:      ELNLABEL = 'LOGNORMAL '
02093:      LENCAT = INDEX (DINFILE,' ')
02094:      LOGFILE = DINFILE
02095:      LOGFILE (LENCAT:LENCAT+3) = '.LOG'
02096:      OPEN (UNIT=20,FILE=LOGFILE)
02097:      READ (20,67,END=65,ERR=666) NDUM,NDEE
02098:67      FORMAT (2I3)
02099:      NSUITE(1) = NDUM

```

```

02100:      NUMCOM = NDEE
02101:      DO 66 I = 1, NSUITE(1)
02102:      READ (20,69,ERR=666) (ZM(I,K),ZD(I,K),K=1,NUMCOM)
02103:69      FORMAT (20E12.5)
02104:68      CONTINUE
02105:      DO 70 J = 1, NUMCOM
02106:      READ (20,71,ERR=666) (AEMP(1,I,J,K),K=1,NUMCOM)
02107:71      FORMAT (10F8.5)
02108:70      CONTINUE
02109:      READ (20,72,ERR=666) SPROB(1,I)
02110:72      FORMAT (F8.5)
02111:66      CONTINUE
02112:      CLOSE (20)
02113:      RETURN
02114:666      CLOSE (20)
02115:      WRITE (1,*) ('L/N ERR, GOING TO EMP ROUTINE...')
02116:      WRITE (UNITMP,667)
02117:667      FORMAT (/6X,'ABNORMAL L/N RUN, EMPIRICAL ROUTINE USED.')
02118:      ELOPT = 'E'
02119:      ELNLABEL = 'EMPIRICAL '
02120:      RETURN
02121:C
02122:C      STORE VALUES FROM A NORMAL TABLE
02123:C
02124:65      I = 1
02125:      MOD = 1
02126:      OPEN (UNIT = 19,STATUS='SCRATCH')
02127:      J = 1
02128:      DO 555 J=1,10
02129:      IEMET = IEMPCOD(II,J)
02130:      IF (IEMET.EQ.0) GO TO 555
02131:      WRITE (19,222) IEMET
02132:222      FORMAT (I2)
02133:555      CONTINUE
02134:      REWIND (19)
02135:      I = 1
02136:333      READ (19,222,END=444) ICO(MEMP,I)
02137:      I = I + 1
02138:      GO TO 333
02139:444      INFOC(1) = I - 1
02140:C
02141:C      UNIT=8 IS THE FILE THAT FILLS ICO() AND INFOC()
02142:C      THERE IS A FILE THAT TELLS HOW MANY AND WHAT KIND METALS IN
02143:C      EACH OF THE MODELS.  ASSUME THAT INFOC IS IN AN ARRAY
02144:C      INFOC(MEMP) HAS NUMBER OF METALS, ICO(MEMP,C) HAS METAL CODES
02145:C      FOR EACH OF THE MODELS
02146:C
02147:C      THERE WILL HAVE TO BE SOME CODE MATCHING THE INDICIES IN
02148:C      FILE(I) AND INFOC(I) SO THAT THEY ARE THE SAME MODELS.
02149:C
02150:C      INFOC(1)=3      THIS IS NOW SET AT METAL SETTINGS.
02151:C
02152:      DO 11 I11=1,MOD
02153:      OPEN(UNIT=7,FILE=DINFILE)
02154:      DO 12 L=1,MAXD
02155:      DO 13 I=1,ICP1
02156:      DEMP(L,I)=0.
02157:13      CONTINUE
02158:12      CONTINUE
02159:C      READ IN DATA FROM THE MODELS ONE AT A TIME

```

```

02160:      I=1
02161:      14 READ (7,15,END=16) (DEMP(I,J),J=1,INFOC(I11))
02162:      15 FORMAT(25X,F12.0,F9.0,F9.0,6X,F11.0,4X,F12.0,4X,F9.0,
02163:      C 7X,F12.0)
02164:      IS=0
02165:C      COMPUTE THE SUITE INDEX
02166:      DO 17 I17=1,INFOC(I11)
02167:      IF(DEMP(I,I17).GT.0.) IS=IS+2** (I17-1)
02168:C      IF(I17.GT.1) DEMP(I,I17)=DEMP(I,I17)/100.
02169:      17 CONTINUE
02170:      DEMP(I,ICP1)=IS
02171:      I=I+1
02172:      GO TO 14
02173:      16 CLOSE(7)
02174:C      N11= THE NUMBER OF DEPOSITS IN MODEL I11
02175:      N11=I-1
02176:C      GROUP THE DATA SO DEPOSITS WITH COMMON SUITES OF METALS ARE
02177:C      TOGETHER
02178:      CALL SORTD(N11)
02179:C      NOW WE COUNT THE SUITES AND SEE WHERE THEY START AND END
02180:      NSUITE(I11)=1
02181:      ISUITE(I11,1)=1
02182:      JSUITE(I11,1)=1
02183:      ISTOP=N11-1
02184:      DO 18 I=1,ISTOP
02185:      TEST=DEMP(I,ICP1)-DEMP(I+1,ICP1)
02186:      IF(TEST.GT.0.5) GO TO 19
02187:      JSUITE(I11,NSUITE(I11))=I+1
02188:      GO TO 13
02189:      19 NSUITE(I11)=NSUITE(I11)+1
02190:      ISUITE(I11,NSUITE(I11))=I+1
02191:      JSUITE(I11,NSUITE(I11))=I+1
02192:      18 CONTINUE
02193:      DO 20 I=1,NSUITE(I11)
02194:      DUM=N11
02195:      DEE=JSUITE(I11,I)-ISUITE(I11,I)+1.
02196:      SPROB(I11,I)=DEE/DUM
02197:      20 CONTINUE
02198:      DO 21 I=1,NSUITE(I11)
02199:      DO 22 J=1,IC
02200:      DO 23 K=1,IC
02201:      DUM=0
02202:      DO 24 L=ISUITE(I11,I),JSUITE(I11,I)
02203:      DUM=DUM+DEMP(L,K)*DEMP(L,J)
02204:      24 CONTINUE
02205:      DUM=DUM/(JSUITE(I11,I)-ISUITE(I11,I)+1.)
02206:      P(I11,I,J,K)=DUM
02207:      23 CONTINUE
02208:      22 CONTINUE
02209:      21 CONTINUE
02210:C      CALCULATE LOGNORMAL PARAMETERS ZM(SUIT,COMMODITY), ZD(*,*)
02211:      DO 85 I=1,NSUITE(I11)
02212:      DO 76 K=1,NUMCOM
02213:      METYES(K)=0
02214:      ZM(I,K)=0.
02215:      ZD(I,K) = -1.0
02216:      DUM=0.
02217:      DEE=0.
02218:      IF (DEMP(ISUITE(I11,I),K).GT.0.0000001) METYES(K) = 1
02219:      DIM=JSUITE(I11,I)-ISUITE(I11,I)+1

```

```

02220:      IF(DIM.LT.1.5) GO TO 78
02221:      DO 77 J=ISUITE(I11,I),JSUITE(I11,I)
02222:      IF(DEMP(J,K).GT.0.000001) METYES(K)=1.
02223:      DUM=DUM+DEMP(J,K)
02224:      DEE=DEE+DEMP(J,K)*DEMP(J,K)
02225: 77 CONTINUE
02226:      IF(METYES(K).EQ.0) GO TO 76
02227:      DUM=DUM/DIM
02228:      DEE=DEE-DUM*DUM*DIM
02229:      DEE=DEE/(DIM-1.)
02230:      GO TO 79
02231: 78 DUM=DEMP(JSUITE(I11,I),K)
02232:      DEE=DUM*DUM/12.
02233:      IF (METYES(K).EQ.0) GO TO 76
02234: 79 ZD(I,K)=SQRT(ALOG(DEE/(DUM*DUM)+1.))
02235:      ZM(I,K)=ALOG(DUM*EXP(-ZD(I,K)*ZD(I,K)/2.))
02236: 76 CONTINUE
02237:      DO 81 J=1,NUMCOM
02238:      IF (METYES(J).EQ.0) GO TO 81
02239:      DUM1=EXP(ZM(I,J)+ZD(I,J)*ZD(I,J)/2.)
02240:      DO 82 K=1,NUMCOM
02241:      R(J,K)=0.
02242:      DUM2=EXP(ZM(I,K)+ZD(I,K)*ZD(I,K)/2.)
02243:      IF(J.EQ.K) R(J,K)=1.
02244:      IF (J.EQ.K) GO TO 82
02245:      IF(METYES(K).EQ.0.OR.METYES(J).EQ.0) GO TO 82
02246:      DUM = P(I11,I,J,K)/(DUM1*DUM2)
02247:      R(J,K)=ALOG(DUM)/(ZD(I,J)*ZD(I,K))
02248: 82 CONTINUE
02249: 31 CONTINUE
02250:      DO 83 J=2,NUMCOM
02251:      IF(R(1,J).LT.1..AND.R(1,J).GT.-1.) GO TO 83
02252:      WRITE(1,84) DINFILE
02253: 84 FORMAT('LOGNORMAL NO GOOD ON ',A60)
02254:      WRITE (13,844)
02255: 844 FORMAT (/,'MODEL MISSPECIFICATION ERROR  TON/DEPOSIT',/)
02256:      DUM1 = EXP(ZM(I,1)+(ZD(I,1)**2)/2.)
02257:      IF (R(1,J).GT.1.) DUM = (-EXP(ZD(I,J)*ZD(I,1)*R(1,J))
02258:      C +EXP(ZD(I,J)*ZD(I,J)))
02259:      IF (R(1,J).LT.-1.) DUM = (-EXP(ZD(I,J)*ZD(I,1)*R(1,J))
02260:      C +EXP(-ZD(I,J)*ZD(I,1)))
02261:      DUMJ = EXP(ZM(I,J)+(ZD(I,J)**2)/2.)
02262:      ERR(J) = ERR(J)+SPROB(1,I)*DUM1*DUMJ*DUM/100.
02263: COMMENT      IF (R(1,J).GT.1.) R(1,J) = 1.
02264: COMMENT      IF (R(1,J).LT.-1.) R (1,J) = -1.
02265: 83 CONTINUE
02266: C      FILL THE MATRIX OF COEFFICIENTS, AEMP(I11,I,...), FOR LINEAR
02267: C      COMBINATIONS OF NORMALS THAT GIVE THE RIGHT CORRELATIONS
02268:      DO 32 K=1,NUMCOM
02269:      IF(METYES(K).EQ.0) AEMP(I11,I,K,K)=1.
02270:      IF(METYES(K).EQ.1) AEMP(I11,I,K,1)=R(K,1)
02271:      IF(METYES(K).EQ.0) GO TO 32
02272:      DUM=0.
02273:      IF(K.EQ.1) GO TO 32
02274:      DO 33 J=2,K
02275:      DUM=DUM+AEMP(I11,I,K,J-1)**2
02276:      IF(DUM.LE.1.) GO TO 34
02277:      DUM = DUM-AEMP(I11,I,K,J-1) ** 2
02278:      DEE = SQRT (1.-DUM)
02279:      SGN = 1.

```

```

02280:      IF (AEMP(I11,I,K,J-1).LT.0.) SGN = -1.
02281:      AEMP (I11,I,K,J-1) = SGN * DEE
02282:      WRITE(1,35) I11,DUM
02283: 35  FORMAT('INCONSISTENT CORRELATIONS',1X,I3,3X,E11.4)
02284:      AEMP(I11,I,K,K) = 0.
02285:      GO TO 32
02286: 34  IF(J.EQ.K) AEMP(I11,I,K,J)=SQRT(1.-DUM)
02287:      IF(J.EQ.K) GO TO 33
02288:      IF(METYES(J).EQ.0) GO TO 33
02289:      JM1=J-1
02290:      DEE=0.
02291:      DO 36 L=1,JM1
02292:      DEE=DEE+AEMP(I11,I,K,L)*AEMP(I11,I,J,L)
02293: 36  CONTINUE
02294:      DEE=R(K,J)-DEE
02295:      IF(AEMP(I11,I,J,J).GT.0.) GO TO 37
02296:      AEMP(I11,I,K,J)=0.
02297:      GO TO 33
02298: 37  AEMP(I11,I,K,J)=DEE/AEMP(I11,I,J,J)
02299: 33  CONTINUE
02300: 32  CONTINUE
02301:      DO 44 K=1,IC
02302:      DO 45 J=1,IC
02303:      DUM=0.
02304:      DO 46 L=1,IC
02305:      DUM=DUM+AEMP(I11,I,J,L)*AEMP(I11,I,K,L)
02306: 46  CONTINUE
02307:      IF(METYES(J).EQ.0.OR.METYES(K).EQ.0) GO TO 44
02308: 45  CONTINUE
02309: 44  CONTINUE
02310: 85  CONTINUE
02311: 11  CONTINUE
02312:      WRITE (1,*) (' ')
02313:      REWIND (20)
02314:C    INFOC(MEMP) NOW EQUALS NUMCOM
02315:      INFOC(MEMP) = NUMCOM
02316:      WRITE (20,67) NSUITE(1),INFOC(MEMP)
02317:      DO 73 I = 1,NSUITE(1)
02318:      WRITE (20,69) (ZM(I,K),ZD(I,K), K = 1,NUMCOM)
02319:      DO 75 J = 1,NUMCOM
02320:      WRITE (20,71) (AEMP(1,I,J,K), K = 1,NUMCOM)
02321:75  CONTINUE
02322:      WRITE (20,72) SPROB(1,I)
02323:73  CONTINUE
02324:      ENDFILE (20)
02325:      CLOSE (20)
02326:      RETURN
02327:      END
02328:C
02329:      SUBROUTINE QUANTILE (EST,NSUT,METQ)
02330:      COMMON /OWN/L1(521),P2(32),LP,LQ,NQ1,MM,MEMP
02331:      COMMON /CLN/ CLINE(3,100,12,10)
02332:      COMMON /NUMDEP/ RNUMD(3,12,10),NQ,KKONT(12,10)
02333:      REAL PCENT(3)
02334:      REAL THOLD,TL,TR,PONE,PTWO
02335:      INTEGER*4 ILBASK(3,12,10),IRBASK(3,12,10),ISKIPF,ISRCH,ITSRCH,I
02336:      PCENT(1) = 0.9
02337:      PCENT(2) = 0.5
02338:      PCENT(3) = 0.1
02339:      KKONT(NSUT,METQ) = KKONT(NSUT,METQ) + 1

```

```

02340:C
02341:      K = KKONT (NSUT,METQ) -1
02342:C
02343:      IF (KKONT(NSUT,METQ).GT.100) GO TO 115
02344:      DO 1515 NQ = 1,3
02345:      CLINE(NQ,KKONT(NSUT,METQ),NSUT,METQ) = EST
02346:C
02347:      IF (K.EQ.0) GO TO 1515
02348:      DO 93 I = 1,K
02349:      J = K-I+2
02350:      DUM = CLINE (NQ,J,NSUT,METQ)
02351:      DUMM = CLINE (NQ,J-1,NSUT,METQ)
02352:      IF (DUM.GE.DUMM) GO TO 96
02353:      CLINE (NQ,J,NSUT,METQ) = DUMM
02354:      CLINE (NQ,J-1,NSUT,METQ) = DUM
02355:93      CONTINUE
02356:96      NNQ = ((1.-PCENT(NQ)) * (K+1) + 1.0)
02357:      RNUMD(NQ,NSUT,METQ) = CLINE (NQ,NNQ,NSUT,METQ)
02358:1515     CONTINUE
02359:      IF (KKONT(NSUT,METQ).EQ.100) GO TO 112
02360:      IF (KKONT(NSUT,METQ).LE.100) RETURN
02361:112      CALL DRSORT (NSUT,METQ)
02362:113      FORMAT (100F9.5)
02363:      DO 1155 IBASK = 1,3
02364:      ILBASK (IBASK,NSUT,METQ) = 0
02365:      IRBASK (IBASK,NSUT,METQ) = 0
02366:1155     CONTINUE
02367:      RETURN
02368:115      DO 123 NQ = 1,3
02369:      INSERT = KKONT(NSUT,METQ) - 100
02370:      XN = INSERT + 100
02371:      PTWO = PCENT(NQ) - (100./(2.*XN))
02372:      PONE = (1-PCENT(NQ)) - (100./(2.*XN))
02373:      IF (EST.LE.CLINE(NQ,1,NSUT,METQ)) ILBASK(NQ,NSUT,METQ)=
02374:      C ILBASK(NQ,NSUT,METQ) + 1
02375:      IF (EST.GE.CLINE(NQ,100,NSUT,METQ)) IRBASK(NQ,NSUT,METQ)=
02376:      C IRBASK(NQ,NSUT,METQ) + 1
02377:      IF (EST.LE.CLINE(NQ,1,NSUT,METQ).OR.EST.GE.
02378:      C CLINE(NQ,100,NSUT,METQ)) GO TO 998
02379:117      FORMAT ('ILBASK(NQ,NSUT,METQ)=
02380:      C ',I4,'      IRBASK(NQ,NSUT,METQ)= ',I4)
02381:      TL=(((ILBASK(NQ,NSUT,METQ)+1)/XN)-PONE)**2)
02382:      C + (((IRBASK(NQ,NSUT,METQ)/XN) - PTWO)
02383:      C ** 2)
02384:      TR=(((ILBASK(NQ,NSUT,METQ)/XN)-PONE)**2)
02385:      C + (((IRBASK(NQ,NSUT,METQ) + 1)/XN) - PTWO)
02386:      C ** 2)
02387:      IF (TL.LE.TR) GO TO 120
02388:      IF (TR.LE.TL) GO TO 130
02389:120      CONTINUE
02390:      ILBASK(NQ,NSUT,METQ) = ILBASK(NQ,NSUT,METQ) + 1
02391:      ISKIPF = 0
02392:      ITSrch = 0
02393:      DO 50 ISrch = 1,99
02394:      IF (CLINE(NQ,ISrch+1,NSUT,METQ).LE.EST)
02395:      C CLINE(NQ,ISrch,NSUT,METQ)=CLINE(NQ,ISrch+1,NSUT,METQ)
02396:      IF (CLINE(NQ,ISrch+1,NSUT,METQ).LE.EST) GO TO 50
02397:      CLINE(NQ,ISrch,NSUT,METQ) = EST
02398:      GO TO 998
02399:50      CONTINUE

```

```

02400:      GO TO 998
02401:130   IRBASK(NQ,NSUT,METQ) = IRBASK(NQ,NSUT,METQ) + 1
02402:      ISKIPF = 0
02403:      ITSCH = 0
02404:      DO 150 ISCH = 1,99
02405:      JSCH = 100-ISCH + 1
02406:      IF (CLINE(NQ,JSCH-1,NSUT,METQ).GE.EST)
02407:      C CLINE(NQ,JSCH,NSUT,METQ) = CLINE(NQ,JSCH-1,NSUT,METQ)
02408:      IF (CLINE(NQ,JSCH-1,NSUT,METQ).GE.EST) GO TO 150
02409:      CLINE(NQ,JSCH,NSUT,METQ) = EST
02410:      GO TO 998
02411:150   CONTINUE
02412:998   CONTINUE
02413:1188  FORMAT (//,'TL= ',F9.5,'      TR= ',F9.5,/)
02414:      K = (((1.-PCENT(NQ))*(KKONT(NSUT,METQ)+1)) + 0.5)
02415:      ISCH = K-ILBASK(NQ,NSUT,METQ)
02416:      IF (K.LE.ILBASK(NQ,NSUT,METQ)) RNUMD(NQ,NSUT,METQ) =
02417:      C CLINE(NQ,1,NSUT,METQ)
02418:      IF (ISCH.GE.100)
02419:      C RNUMD(NQ,NSUT,METQ) = CLINE(NQ,100,NSUT,METQ)
02420:      IF (ISCH.GE.1.AND.ISCH.LE.100)
02421:      C RNUMD(NQ,NSUT,METQ) = CLINE(NQ,K-ILBASK(NQ,NSUT,METQ),NSUT,METQ)
02422:123   CONTINUE
02423:      RETURN
02424:      END
02425:C
02426:      SUBROUTINE URNMX(NBT,NI,R)
02427:C
02428:C URNMT GENERATES UNIFORM RANDOM NUMBERS IN THE INTERVAL
02429:C 0 TO 1 USING THE METHOD OF TAUSWORTHE. VARIABLE NBT (A
02430:C POSITIVE INTEGER) IS THE STARTING BIT OF THE SEQUENCE.
02431:C VARIABLE NI IS 1 TO INITIALIZE URNMT AND IS 2 THEREAFTER.
02432:C VARIABLE R IS THE RANDOM NUMBER. THE PRIMITIVE
02433:C POLYNOMIAL IS XEMP**LP + XEMP**LQ + 1.
02434:C
02435:      COMMON /OWN/L1(521),P2(32),LP,LQ,NQ1,MM,MEMP
02436:      GO TO (5,50), NI
02437:C
02438:C INITIALIZATION ROUTINE.
02439:C
02440:5      LP=521
02441:      LQ = 32
02442:      NQ1 = LQ+1
02443:      MM = LP/LQ
02444:C
02445:C A PRIME NUMBER 48611 IS USED TO GENERATE THE FIRST LP
02446:C BITS.
02447:C
02448:      DO 10 I=1,LP
02449:10     L1(I) = MOD(48611/I,2)
02450:      DO 15 I = 1,LQ
02451:15     P2(I) = 0.5**I
02452:      NI = 2
02453:      MEMP = 0
02454:      KL =(NBT - 1)/LP
02455:      IF (KL.EQ.0) GO TO 30
02456:      NSW1 = 1
02457:      DO 27 J = 1, KL
02458:      GO TO 52
02459:27     CONTINUE

```

```

02460:30      KNBT = NBT - KL*LP - 1
02461:        IF (KNBT.EQ.0) GO TO 50
02462:C
02463:C  SHIFT BITS.
02464:C
02465:        DO 40 J = 1,KNBT
02466:        NTMP = L1(1)+ L1(LP-LQ+1)
02467:        IF (NTMP.EQ.2) NTMP = 0
02468:        DO 37 JJ = 2,LP
02469:37      L1(JJ-1) = L1(JJ)
02470:        L1(LP) = NTMP
02471:40      CONTINUE
02472:50      R = 0.0
02473:        MEMP = MEMP+1
02474:        IF (MEMP.LE.MM) GO TO 60
02475:        MEMP = 1
02476:        NSW1 = 2
02477:C
02478:C  GENERATE THE NEXT LP BITS.
02479:C
02480:52      DO 53 I = 1,LQ
02481:        NS = LP - LQ + I
02482:        L1(I) = L1(I) + L1(NS)
02483:        IF(L1(I).EQ.2) L1(I) = 0
02484:53      CONTINUE
02485:        DO 54 I = NQ1,LP
02486:        L1(I) = L1(I) + L1(I-LQ)
02487:        IF (L1(I).EQ.2) L1(I)= 0
02488:54      CONTINUE
02489:        GO TO (27,60)  NSW1
02490:C
02491:C      COMPUTE RANDOM NUMBER R.
02492:C
02493:60      DO 70 I =1,LQ
02494:70      R = R + P2(I)*FLOAT(L1((MEMP-1)*LQ+I))
02495:        RETURN
02496:        END
02497:C
02498:        SUBROUTINE DRSORT (NSUT,METQ)
02499:        COMMON /CLN/ CLINE(3,100,12,10)
02500:        COMMON /NUMDEP/ RNUMD(3,12,10),NQ,KKONT(12,10)
02501:        DO 4 NQ = 1,3
02502:3        DO 1 I = 1,99
02503:        IF (CLINE(NQ,I,NSUT,METQ).LE.CLINE(NQ,I+1,NSUT,METQ)) GO TO 1
02504:        DUM = CLINE(NQ,I,NSUT,METQ)
02505:        CLINE(NQ,I,NSUT,METQ) = CLINE(NQ,I+1,NSUT,METQ)
02506:        CLINE(NQ,I+1,NSUT,METQ) = DUM
02507:        ISWITCH = 1
02508:1        CONTINUE
02509:        IF (ISWITCH.EQ.0) GO TO 2
02510:        ISWITCH = 0
02511:        GO TO 3
02512:2        CONTINUE
02513:4        CONTINUE
02514:        RETURN
02515:        END
02516:C
02517:        SUBROUTINE SRT(N,IMETAL)
02518:        COMMON/ MET/ SMETAL
02519:        REAL SMETAL(4999,35)

```

```

02520:      DO 60 K=1,35
02521:      MO=N
02522:20    IF(MO.LE.1) GO TO 60
02523:      J=4
02524:      IF(MO.GT.15) J=8
02525:      MO=2 * (MO/J) + 1
02526:      KO=N-MO
02527:      JO=1
02528:30    I=JO
02529:40    J=I + MO
02530:      IF (SMETAL(I,K).LE.SMETAL(J,K)) GO TO 50
02531:      TTTT=SMETAL(I,K)
02532:      SMETAL(I,K)=SMETAL(J,K)
02533:      SMETAL(J,K)=TTTT
02534:      I=I-MO
02535:      IF(I-1) 50,40,40
02536:50    JO=JO + 1
02537:      IF(JO-KO) 30,30,20
02538:60    CONTINUE
02539:      RETURN
02540:      END
02541:C
02542:C      PORTABLE RANDOM NUM GENERATOR USING THE RECURSION
02543:C      IX = IX * A MOD P
02544:C      ACM TRANS ON MATH SOFTWARE, V5, N2, JUNE 1979
02545:C
02546:      FUNCTION RAND()
02547:      SAVE IX
02548:      INTEGER * 4 A,P,IX,B15,B16,XHI,XALO,LEFTLO,FHI,K
02549:      DATA A/16807/,B15/32768/,B16/65536/,P/2147483647/
02550:      DATA IX/1/
02551:      IF (IX.NE.1) GO TO 10
02552:C      RANDOM NUMBER SEED IS SET.
02553:      IX = 987
02554:C      GET 15 HI ORDER BITS OF IX
02555:10    XHI = IX/B16
02556:C      GET 16 LO BITS OF IX AND FORM LO PRODUCT
02557:      XALO = (IX-XHI * B16) * A
02558:C      GET 15 HI ORDER BITS OF LO PRODUCT
02559:      LEFTLO = XALO/B16
02560:C      FORM THE 31 HIGHEST BITS OF FULL PRODUCT
02561:      FHI = XHI * A + LEFTLO
02562:C      GET OVERFLO PAST 31ST BIT OF FULL PRODUCT
02563:      K = FHI/B15
02564:C      ASSEMBLE ALL THE PARTS AND PRESUBTRACT P
02565:      IX = (((XALO-LEFTLO * B16)-P) + (FHI-K * B15) * B16) + K
02566:C      ADD P BACK IN IF NECESSARY
02567:      IF (IX.LT.0) IX = IX + P
02568:C      MULTIPLY BY 1/(2 ** 31-1)
02569:      RAND = FLOAT(IX) * 4.656612875E-10
02570:      RETURN
02571:      END

```

DIALOGUE BETWEEN USER AND THE MARK3 PROGRAM

R MARK3

ENTER "M" TO DISPLAY LISTING OF AVAILABLE MODELS.
ENTER "Q" TO END PROGRAM.
OTHERWISE PRESS C/R TO CONTINUE: (C/R PRESSED)

NEED A LISTING OF DATA WITH MODEL/METAL IDS?
ENTER "P" TO CREATE A PRINTABLE FILE
ENTER "Q" TO END PROGRAM.
OTHERWISE PRESS C/R TO CONTINUE: (C/R PRESSED)

PLEASE TYPE INITIAL INFORMATION FOR HEADER PAGE -
REQUESTED BY:
DAVE ROOT
ASSESSMENT AREA:
SAMPLE SESSION
DATE REQUESTED:
DECEMBER 27, 1991
FOR TRACT TOTALS, ENTER "T"
FOR GRAND TOTAL, ENTER "G"

G

LIKE A ".CPL" FILE CREATED ?
Y
ENTER NAME FOR CPL FILE:
SAMP13

ENTER NUMBER OF MODELS FOR THIS RUN:
1

A PROMPT APPEARS FOR EACH MODEL.

ENTER SINGLE MODEL(INDEX) NUMBER:
13

ENTER UNIQUE FILE NAME FOR MARK3 OUTPUT RESULTS:
S13.OUT

TERMINAL INPUT FOR MODEL 13 - SEDIMENTARY EXHALATIVE ZN-PB(31A)

ENTER "L" FOR LOGNORMAL
ENTER "E" FOR EMPIRICAL
E

ENTER EST. # OF DEP. COUNT(3 OR 5):
3

ENTER ESTIMATED NUMBER OF DEPOSITS(3) -
90% 50% 10%
ND(2) ND(3) ND(4)

90% =
2
50% =
3
10% =
8

TO FIX THE PROBABILITY OF ZERO DEPOSITS,
ENTER PROBABILITY BETWEEN 0. & 1.
OR ENTER -1. FOR DEFAULT.
-1.

CPL FILE CREATED IS - SAMP13.CPL

**** STOP

EMPIRICAL (2, 3, 8)
13 - SEDIMENTARY EXHALATIVE ZN-PB(31A)

SORTED SIMULATION RESULTS FOR ZINC

ORDER OF OCCURRENCE

	UNCONDITIONAL		CONDITIONAL
1	0.00000E+01	191	8.74692E+03
500	5.82414E+05	671	1.03387E+06
1000	2.03174E+06	1152	2.46171E+06
1500	3.84632E+06	1633	4.43486E+06
2000	6.33786E+06	2114	6.93621E+06
2500	9.15906E+06	2595	9.72870E+06
3000	1.23203E+07	3076	1.28333E+07
3500	1.68147E+07	3557	1.73565E+07
4000	2.27293E+07	4038	2.31174E+07
4500	3.36401E+07	4519	3.41827E+07
4750	4.39487E+07	4759	4.43068E+07
4950	6.29718E+07	4951	6.30082E+07
4999	1.08564E+08	4999	1.08564E+08
	EXPECTED MEAN ZINC		
	1.36231E+07		1.41613E+07

N= 0	FREQ(N)=	190	PROB(N)= 0.0400
N= 1	FREQ(N)=	205	PROB(N)= 0.0400
N= 2	FREQ(N)=	1081	PROB(N)= 0.2200
N= 3	FREQ(N)=	1232	PROB(N)= 0.2400
N= 4	FREQ(N)=	419	PROB(N)= 0.0800
N= 5	FREQ(N)=	363	PROB(N)= 0.0800
N= 6	FREQ(N)=	399	PROB(N)= 0.0800
N= 7	FREQ(N)=	411	PROB(N)= 0.0800
N= 8	FREQ(N)=	699	PROB(N)= 0.1400

MEAN NUMBER OF DEPOSITS 4.0842

SORTED SIMULATION RESULTS FOR SILVER

ORDER OF OCCURRENCE

	UNCONDITIONAL		CONDITIONAL
1	0.00000E-01	260	8.13223E-01
500	1.54448E+02	733	3.88874E+02
1000	8.72785E+02	1207	1.36024E+03
1500	2.07552E+03	1681	2.57228E+03
2000	3.59430E+03	2155	4.20044E+03
2500	5.56829E+03	2629	6.26620E+03
3000	8.49581E+03	3103	9.22244E+03
3500	1.23178E+04	3577	1.30773E+04
4000	1.76006E+04	4051	1.83126E+04
4500	2.92635E+04	4525	3.00567E+04
4750	4.00278E+04	4762	4.09144E+04
4950	6.28864E+04	4952	6.35419E+04
4999	1.30196E+05	4999	1.30196E+05

EXPECTED MEAN SILVER
1.08485E+04 1.14412E+04

N= 0	FREQ(N)=	190	PROB(N)= 0.0400
N= 1	FREQ(N)=	205	PROB(N)= 0.0400
N= 2	FREQ(N)=	1081	PROB(N)= 0.2200
N= 3	FREQ(N)=	1232	PROB(N)= 0.2400
N= 4	FREQ(N)=	419	PROB(N)= 0.0800
N= 5	FREQ(N)=	363	PROB(N)= 0.0800
N= 6	FREQ(N)=	399	PROB(N)= 0.0800
N= 7	FREQ(N)=	411	PROB(N)= 0.0800
N= 8	FREQ(N)=	699	PROB(N)= 0.1400

MEAN NUMBER OF DEPOSITS 4.0842

SORTED SIMULATION RESULTS FOR LEAD

ORDER OF OCCURRENCE

	UNCONDITIONAL		CONDITIONAL
1	0.00000E-01	191	2.65203E+03
500	2.71570E+05	671	4.70677E+05
1000	1.01269E+06	1152	1.28183E+06
1500	2.07037E+06	1633	2.37983E+06
2000	3.48658E+06	2114	3.79464E+06
2500	5.06755E+06	2595	5.46176E+06
3000	7.24959E+06	3076	7.62792E+06
3500	1.00246E+07	3557	1.03218E+07
4000	1.37949E+07	4038	1.41498E+07
4500	1.97871E+07	4519	2.00852E+07
4750	2.67210E+07	4759	2.68503E+07
4950	3.84290E+07	4951	3.85822E+07
4999	6.58951E+07	4999	6.58951E+07
	EXPECTED MEAN LEAD		
	8.05667E+06		8.37499E+06

N= 0	FREQ(N)=	190	PROB(N)= 0.0400
N= 1	FREQ(N)=	205	PROB(N)= 0.0400
N= 2	FREQ(N)=	1081	PROB(N)= 0.2200
N= 3	FREQ(N)=	1232	PROB(N)= 0.2400
N= 4	FREQ(N)=	419	PROB(N)= 0.0800
N= 5	FREQ(N)=	363	PROB(N)= 0.0800
N= 6	FREQ(N)=	399	PROB(N)= 0.0800
N= 7	FREQ(N)=	411	PROB(N)= 0.0800
N= 8	FREQ(N)=	699	PROB(N)= 0.1400

MEAN NUMBER OF DEPOSITS 4.0842

SORTED SIMULATION RESULTS FOR
DEPOSIT TONNAGE

ORDER OF OCCURRENCE

	UNCONDITIONAL		CONDITIONAL
1	0.00000E-01	191	4.23218E+05
500	1.11790E+07	671	2.00027E+07
1000	3.54588E+07	1152	4.57709E+07
1500	6.74995E+07	1633	7.68843E+07
2000	1.02523E+08	2114	1.09873E+08
2500	1.42539E+08	2595	1.49791E+08
3000	1.91555E+08	3076	1.99279E+08
3500	2.46781E+08	3557	2.52070E+08
4000	3.15280E+08	4038	3.21577E+08
4500	4.22374E+08	4519	4.27353E+08
4750	5.16485E+08	4759	5.19424E+08
4950	6.93763E+08	4951	6.95226E+08
4999	1.05460E+09	4999	1.05460E+09

EXPECTED MEAN DEPOSIT TONNAGE

1.85783E+08	1.93123E+08
-------------	-------------

N= 0	FREQ(N)=	190	PROB(N)= 0.0400
N= 1	FREQ(N)=	205	PROB(N)= 0.0400
N= 2	FREQ(N)=	1081	PROB(N)= 0.2200
N= 3	FREQ(N)=	1232	PROB(N)= 0.2400
N= 4	FREQ(N)=	419	PROB(N)= 0.0800
N= 5	FREQ(N)=	363	PROB(N)= 0.0800
N= 6	FREQ(N)=	399	PROB(N)= 0.0800
N= 7	FREQ(N)=	411	PROB(N)= 0.0800
N= 8	FREQ(N)=	699	PROB(N)= 0.1400

MEAN NUMBER OF DEPOSITS	4.0842
-------------------------	--------

Appendix 1

METALS IN 13 - SEDIMENTARY EXHALATIVE ZN-PB(31A)

DEPOSITS	COUNTRY	TONNES	LEAD	ZINC	SILVER
BALM		21000000.	0.5	9.0	0.00
BARO		9000000.	3.53	1.51	0.0034
BIGS		101000000.	1.01	2.45	0.0011
BLAC		36000000.	2.67	0.56	0.0023
BROK		72000000.	4.28	2.32	0.0058
BROK		180000000.	11.3	9.8	0.0175
CIRQ		30000000.	2.2	7.8	0.0048
COTT		730000.	6.0	5.0	0.0069
DUGA		1300000.	1.6	11.6	0.0037
DUNC		9000000.	2.7	2.9	0.00
DY		15000000.	5.6	7.1	0.0084
ELUR		27000000.	5.6	8.3	0.0139
FARO		58000000.	3.41	5.72	0.0041
FX		1600000.	0.58	2.6	0.00
GRUM		35000000.	4.23	6.69	0.0064
HB		6500000.	0.77	4.1	0.00048
HILT		37000000.	7.7	9.5	0.0179
HOME		1500000.	2.5	4.0	0.0240
HOWA		270000000.	1.46	3.64	0.00
JERS		7700000.	1.65	3.49	0.00031
KING		2600000.	5.1	5.6	0.0038
LADY		8900000.	6.7	18.1	0.0100
MACM		910000.	5.0	10.0	0.0062
MATT		590000.	6.1	4.6	0.0093
MCAR		240000000.	4.1	9.2	0.0041
MEGG		60000000.	1.3	10.0	0.0003
MEL		4800000.	2.05	5.6	0.00025
MINE		2400000.	2.18	4.73	0.0023
MOUN		89000000.	7.1	6.1	0.0160
NAVA		90000000.	2.3	10.0	0.00
RAJP		30000000.	1.70	6.14	0.0054
RAMM		30000000.	9.0	19.0	0.0085
RAMP		61000000.	1.57	13.48	0.0054
REDD		77000000.	5.0	17.1	0.0082
REEV		5800000.	0.98	3.42	0.00034
ROSH		11000000.	2.0	8.0	0.00
SILV		18000000.	2.8	7.4	0.0021
SQUI		1800000.	0.38	4.4	0.00032
SULL		160000000.	6.10	5.9	0.0068
SWIM		4300000.	3.8	4.7	0.0047
TOM		17000000.	4.55	6.4	0.0045
TYNA		12000000.	4.9	4.5	0.0058

LOGNORMAL FILE

```

2 4
0.17027E+02 0.10142E+01 0.12253E+01 0.57304E+00 0.18027E+01 0.56603E+00-0.53670E+01 0.76566E+00
1.00000 0.00000 0.00000 0.00000 0.00000
0.37232 0.92810 0.00000 0.00000
0.14492 0.41058 0.90023 0.00000
0.20277 0.66370-0.01184 0.71989
0.35714
0.17406E+02 0.11098E+01 0.32239E+00 0.53168E+00 0.16617E+01 0.51762E+00 0.00000E+00-0.10000E+01
1.00000 0.00000 0.00000 0.00000 0.00000
0.04905 0.99830 0.00000 0.00000
-0.17738 0.05213 0.98276 0.00000
0.00000 0.00000 0.00000 1.00000
0.14286

```

Appendix 3

EMPIRICAL FILE FOR REQUESTED MODEL

```

2 4
0.29517E+06 0.19001E+00 0.28001E+00 0.12502E-03
0.11648E+07 0.10100E+01 0.23642E+01 0.35277E-03
0.26145E+07 0.15115E+01 0.33502E+01 0.14192E-02
0.56226E+07 0.21078E+01 0.42155E+01 0.28029E-02
0.95909E+07 0.27221E+01 0.50437E+01 0.37766E-02
0.17438E+08 0.33998E+01 0.58881E+01 0.48006E-02
0.29065E+08 0.41881E+01 0.68412E+01 0.58239E-02
0.43780E+08 0.51057E+01 0.81656E+01 0.72345E-02
0.70947E+08 0.61799E+01 0.10092E+02 0.93513E-02
0.11119E+09 0.77485E+01 0.13527E+02 0.13893E-01
0.24677E+09 0.12119E+02 0.23139E+02 0.26131E-01
1.00000 0.00000 0.00000 0.00000
0.31627 0.94867 0.00000 0.00000
0.12255 0.33573 0.93395 0.00000
0.17639 0.58344 0.03291 0.79209
0.85714
0.80019E+06 0.25000E+00 0.13000E+01 0.00000E+00
0.17600E+07 0.42963E+00 0.20913E+01 0.00000E+00
0.59401E+07 0.59074E+00 0.26888E+01 0.00000E+00
0.89098E+07 0.74846E+00 0.32863E+01 0.00000E+00
0.11617E+08 0.11730E+01 0.38838E+01 0.00000E+00
0.14438E+08 0.15075E+01 0.48939E+01 0.00000E+00
0.23100E+08 0.18090E+01 0.62479E+01 0.00000E+00
0.63000E+08 0.21105E+01 0.80331E+01 0.00000E+00
0.11700E+09 0.25284E+01 0.98182E+01 0.00000E+00
0.24300E+09 0.29882E+01 0.11603E+02 0.00000E+00
0.36367E+09 0.37792E+01 0.14073E+02 0.00000E+00
1.00000 0.00000 0.00000 0.00000
0.04414 0.99903 0.00000 0.00000
-0.14355 0.04477 0.98863 0.00000
0.00000 0.00000 0.00000 1.00000
0.14286

```